Safety-Critical Control for Dynamical Systems under Uncertainties

by

Chuanzheng Wang

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, 2024

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, Dept. of Mechanical Engineering
University of Wisconsin-Madison

Supervisor: Jun Liu
Associate Professor, Dept. of Applied Mathematics
University of Waterloo

Co-supervisor: Stephen Smith
Professor, Dept. of Electrical and computer Engineering

Internal Examiner: Brian Ingalls
Professor, Dept. of Applied Mathematics

Internal Examiner: Xinzhi Liu
Professor, Dept. of Applied Mathematics

Internal-External Examiner: Amir Khajepour
Professor, Department of Mechanical and Mechatronics Engineering

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the 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.

Statement of Contributions

This thesis contains collaborative work between Chuanzheng Wang and Yiming Meng under the supervision of my Professors Dr Jun Liu and Dr Stephen Smith. Chuanzheng Wang is the main author conducting the research in the thesis and Yiming Meng contributes to the theoretical analysis for stochastic control barrier function in Chapter 4 and Chapter 6, and robustness analysis for the learning process in Chapter 3 and Chapter 5.

Abstract

Control barrier functions (CBFs) and higher-order control barrier functions (HOCBFs) have shown great success in addressing control problems with safety guarantees. These methods usually find the next safe control input by solving an online quadratic programming problem. However, model uncertainty is a big challenge in synthesizing controllers. This may lead to the generation of unsafe control actions, resulting in severe consequences. In this thesis, we discuss safety-critical control problems for systems with different levels of uncertainties. We first study systems modeled by stochastic differential equations (SDEs) driven by Brownian motion. We propose a notion of stochastic control barrier functions (SCBFs) and show that SCBFs can significantly reduce the control efforts, especially in the presence of noise, and can provide a reasonable worst-case safety probability. Based on this less conservative probabilistic estimation for the proposed notion of SCBFs, we further extend the results to handle higher relative degree safety constraints using higher-order SCBFs. We demonstrate that the proposed SCBFs achieve good trade-offs of performance and control efforts, both through theoretical analysis and numerical simulations.

Next, we discuss deterministic systems with imperfect information. We focus on higher relative degree safety constraints and HOCBFs to develop a learning framework to deal with such uncertainty. The proposed method learns the derivatives of a HOCBF and we show that for each order, the derivative of the HOCBF can be separated into the nominal derivative of the HOCBF and some remainders. This implies that we can use a neural network to learn the remainders so that we can approximate the real residual dynamics of the HOCBF. Next, we study stochastic systems with unknown diffusion terms. We propose a data-driven method to handle the case where we cannot calculate the generator of the stochastic barrier functions. We provide guarantees that the data-driven method can approximate the Itô derivative of the stochastic control barrier function (SCBF) under partially unknown dynamics using the universal approximation theorem.

Finally, we study completely unknown stochastic systems. We extend our assumption into the case where we do not know either the drift or the diffusion term of SDEs. We employ Bayesian inference as a data-driven approach to approximate the system. To be more specific, we utilize Bayesian linear regression along with the central limit theorem to estimate the drift term. Additionally, we employ Bayesian inference to approximate the diffusion term. We also validate our theoretical results using numerical examples in each chapter.

Acknowledgements

First I would like to express my deepest gratitude to my professors Dr Jun Liu and Dr Stephen Smith, for their support, guidance, and mentorship throughout the journey of this thesis. They give me great help and support in both my study and daily life without which, this research would not have been possible. I also want to express my gratitude to my friend and colleague, Yiming Meng and Yinan Li, who have help me greatly throughout my work with discussions and coorporations. Finally, I will convey my thanks to every one in my family for their support during this process.

Table of Contents

List of Figures

List of Tables

List of Symbols

$\mathbb{R}^n$ The n-dimensional Euclidean space.

$\mathbb{R}$ The set of real numbers.

$\mathbb{R}_{\geq 0}$ The set of nonnegative real numbers.

A^c The complement of set A.

$\|\cdot\|_p$ The p-norm.

$C_b(\cdot)$ The space of all bounded continuous functions.

$a \wedge b$ The minimum value between a and b, i.e., $\min(a, b)$.

$\mathcal{X}$ The state space.

$\mathcal{U}$ The control space.

u The control input.

τ The stopping time.

$\mathcal{K}$ The class kappa functions.

r The relative degree.

$\partial\mathcal{C}$ The boundary of set $\mathcal{C}$.

$\mathcal{C}^\circ$ The interior of set $\mathcal{C}$.

X^u The controlled process under u.

$\mathscr{B}(\cdot)$ The Borel σ-algebra.

$\mathbb{P}$ The probability measure.

$\mathbb{P}_x$ The probability measure of a stochastic process X with the initial condition $X_0 = x$.

$\mathbb{E}^x$ The expectation w.r.t. the probability measure $\mathbb{P}_x$.

$\frac{\partial V}{\partial x}$ The partial derivative of V with respect to x.

Chapter 1

Introduction

1.1 Background

Over the past centuries, the control of dynamical systems has gained significant interest among researchers due to its numerous applications across various fields. A primary goal of controlling such systems is to derive feasible controllers that are capable of meeting control objectives. However, in modern control tasks, the more important goal is to optimize system performance while ensuring safety in the control of a dynamical system.

Stability is a crucial characteristic in system control. It implies the system's capacity to converge to a fixed point under suitable controllers [81]. The concept of Lyapunov stability is usually used to verify the stability of dynamical systems, and control Lyapunov functions (CLFs) are utilized to stabilize the systems [70].

Safety is another critical attribute in the control of dynamical systems. It refers to the system's ability to maintain itself in a safe region throughout the control process. The concept of safety control is initially introduced in [83] in the form of correctness and is then formalized in [3], in which the authors stated that a safety property stipulates that some "bad thing" does not happen during execution. For a non-linear system, model predictive control (MPC) [19,89,95] is an effective tool by solving an optimization problem while taking all safety constraints into account in each discrete time step. However, this introduces substantial computational load for real-time implementations.

Set Invariance and Control Barrier Function

In the context of dynamical systems, a subset of the state space is said to be positively invariant if a trajectory enters the set and never exits. This concept can be expanded to keep the trajectory in the set by choosing proper control actions. As a result, the idea of set invariance can be utilized to describe a system's safety property, in the sense that any trajectory that starts within a safe set will never reach the complement of the set, where unsafe things happen.

Inspired by control Lyapunov functions, control barrier functions (CBFs) are proposed to manage the safety control of dynamical systems. Barrier functions, introduced first in constrained optimization problems [16], are continuous functions whose value of a point increase towards infinity as the point approaches the boundary of the feasible region of the optimizing problem. By using barrier functions, we can guarantee the solution to meet the constraints of such optimization problems.

Quadratic programming (QP) is efficient given the fact that its objective functions are quadratic and easy to solve. Therefore, CLFs and CBFs fit well within the QP problems' constraints. Meanwhile, a control affine system is linear with respect to the control inputs. Given the above two benefits, CLFs and CBFs fits well in the form of constraints of QP problems. As such, we can integrate CLFs and CBFs in quadratic programming problems, to render stable and safe control inputs for dynamical systems [5–7]. Also, the cost function of QP problems can vary based on different purposes, such as minimizing control effort [50], or minimizing the gap between actual control and a desired control [94]. It is shown in [109] that finding safe control inputs by solving QP problems can be extended to an arbitrary number of constraints and any nominal control law. As a result, CBFs are widely used in safety control such as lane keeping [8] and obstacle avoidance [26].

1.2 Literature Review

Control Lyapunov Function

Control affine systems have gathered great attention among researchers in various fields over past decades [41]. The main target in the study of these systems is to design effective controllers that can achieve desired control objectives satisfying stability [39], safety [126], and robustness [110] under different conditions.

Stability is a key characteristic of control affine systems. It generally refers to the system's state converging to a fixed point over time under appropriate controllers [10, 97].

Lyapunov stability [69] is usually used in such context to show that the solution of the system is able to stay close to the equilibrium point. Control Lyapunov functions are designed based on Lyapunov theory to guarantee and the resulting controllers will stabilize they control affine systems [51]. For multi-agent systems, control Lyapunov functions are also used to control a class of systems simultaneously for task completion [103]. In recent years, Lyapunov-like functions are also widely applied to complex systems together with learning methods as in [42, 74]. Reinforcement learning (RL) algorithms have been proved to be useful tools in control problems, especially for complex control tasks [1, 124]. The value function in the RL algorithm is constructed as Lyapunov-like functions so that all the states are guaranteed to be within region of attraction [12]. Lyapunov functions are also combined with RL methods to construct safe RL agents that switch among a set of base controllers. This method is first studied in [106] where safety is guaranteed for each controller. In [30], the Lyapunov function is designed to be an upper bound of the constraint cost functions of constraint Markov decision process, such that the corresponding algorithm guarantees feasibility and optimally under certain conditions. This idea is extended to policy-gradient methods for policy function in [31] where near-constraints are satisfied for every policy update by projecting the action onto a set of feasible solution induced by the state-dependent linearized Lyapunov constraints.

Model Predictive Control

Model Predictive Control (MPC) has emerged as an important technology in the field of control engineering, particularly for safety-critical systems over the past decades. The initial form of MPC originated from its broad applications in industrial projects, notably in a project in oil industry, leading to the development such as Dynamic Matrix Control and Model Heuristic Predictive Control [127]. Parallel to these industrial projects, Generalized Predictive Control emerged within the adaptive control area, aiming at the robustness of the controller [34, 35]. The next stage in the development of MPC mainly emphasized to establish stability and optimality in MPC formulations [85]. Robust MPC, studied in [20], addresses model uncertainty and disturbance. Stability results of non-linear systems for constrained optimal control were analyzed in [68] and the work in [88] proposed a method of finding feasible solution instead of global optimal solutions. In their approach, once a feasible solution is found, subsequent calculation preserves the feasibility and only improves the solution by reducing the cost. In [24], the authors come up with an approach called quasi-infinite-horizon MPC, where a quadratic terminal penalty corresponding to the finite horizon cost of the linearized non-linear system is imposed. Given MPC's ability to solve controllers under constraints, it has been increasingly used to guarantee safety by incorporating safety criteria into these constraints. Existing work using MPC considering safety are among [113, 115, 126]. The safety requirements are formulated as constraints in an optimizing problems such as obstacle avoidance [49, 155]. The majority of work in this area is focusing on collision avoidance and consider distance constraints with Euclidean norms [114, 133, 157]. Additionally, MPC has been combined with the Lyapunov function for stability, as shown in [60].

Safety-Critical Control and Control Barrier Function

Research on safety-critical systems primarily focuses on controlling these systems while satisfying state constraints. Besides the traditional method of using MPC with safety constraints during the optimization phase, as discussed above, safety filters are also employed to ensure the systems' safety. Safety filters are used to verify if the resulting controller remains within a predefined safe set. If the controller does not pass the verification, it will be replaced by a known safe controller [131]. Safety filters are also combined with RL to update safety policies as in [134]. However, the use of safety filters often requires a pre-calculated safe set. Reachability analysis are usually used to calculate such a set as in [57], which is computational demanding in practice.

One important method addressing the safety-critical control for dynamical systems is called control barrier functions (CBF). The CBFs have been studied widely since [6] with a great range of applications such as quadrotor [43, 46, 125, 146, 160], autonomous vehicles [79, 123, 126, 143] and robotic control [58, 64, 96, 102, 104, 140]. Safety analysis can also be adapted to focus on stability. These approaches are typically based on Lyapunov stability analysis. If the safe set is designed as a subset of the region of attraction, then CLFs can be employed to ensure a safe and stable controller [112]. CLFs are also combined with CBFs to guarantee safety for reachability problems as in [37, 128, 136].

Initially, the study of CBFs is mainly focusing on relative degree of one, which means that the control input appears in the first order derivative of the CBF so that we could solve optimization problem directly. However, for more complicated systems, this is not the case. As a result, people start to study CBFs with higher-order scenarios to address this degree issue. Exponential control barrier functions (ECBFs) [101] and higher-order control barrier functions (HOCBFs) [147] are proposed to deal with high relative degree. And research focusing on safety-critical control with high relative degree are well studied in [36, 78, 149, 154].

Control Barrier Function with Reinforcement Learning

Most of the above research works aim to control a deterministic system with perfect information. For systems with uncertain dynamics, learning-based methods are applied to design controllers with CBFs. Reinforcement learning has been combined with CBFs, as seen in [18,28–30,45,158], to design controllers under uncertain dynamics. In [28], a model-free RL algorithm is proposed that combines with model-based CBFs. The dynamics of the model, which are initially unknown, are estimated as the RL algorithm explores the system's states. Another approach is presented in [87] where an off-policy method is used to learn a policy without any knowledge of the systems. During the learning phase, the RL agent uses a safe but conservative policy, achieved by integrating CBFs into the value function. As a result, the learned policy will stay inside the safe set due to the constraints from the CBF. A coefficient is used to serve as a trade-off between optimality of the value function and safety resulting from the CBF.

Control Barrier Function with Supervised Learning

CBFs are also integrated into supervised learning (SL) since an SL algorithm can learn a model through data collection, sampling from controlled trajectories. Unlike RL methods that learn a policy directly, SL algorithms usually identify the system first, and then control the system using the learned system. In [129], the authors present an approach to enhance safety of the system by estimating the model uncertainty using CBFs. A bounded error is considered for the model parameters. Instead of learning the dynamics of the model, they incorporate the uncertainty into the CBFs. It is assumed that if a valid CBF exists for the nominal model, then there is also a valid CBF for the real model such that the dynamics of the CBF can be learned using the observed data. As a result, they use neural networks to identify the dynamics of the CBF to guarantee safety. The work in [137] extends this idea to a more general form for higher relative degree. They show that for each order derivative of the CBF, the real derivative of the CBF can always be represented by the nominal derivative and an extra term. Similar to [129], they also identify the derivative of the CBF to address the uncertainty of the system. A different approach for learning the safe region of unknown system is studied in [142]. This work considers an additional affine disturbance term of the unknown system, that is modelled using a GP to guarantee a high probability confidence interval. Given an initial safe set and with online learning, the safe region is expanded until no more improvement. An adaptive sampling method called information-maximizing-based exploration method is proposed in [11]. A more general approach of safety-critical control using GP model is studied in [75] where the model is identified using a GP model to optimize the system with high probability of safety. A chance constraint is specified using a predefined CBF. Bayesian inference is also utilized to identify the system together with safety certificates as in [17, 40] for unknown systems. In [17], a barrier Bayesian linear regression approach method is proposed for the safety of uncertain system by assuming that the error is bounded between the nominal and true system. In [40], the authors estimate the distribution of the dynamics using Bayesian learning. The objective of the approach is to avoid offline system identification so that the system can autonomously adapt its own model during operation. They propose a matrix variate Gaussian process (MVGP) regression approach with covariance factorization to learn the drift and input gain terms of a non-linear control affine system. The MVGP distribution is then used to optimize the system behavior and ensure safety with high probability.

Control Barrier Function for Stochastic System

Safety-critical control addressing stochastic noise is also well studied in recent years. The reciprocal stochastic control barrier function and zeroing stochastic control barrier function are considered in [33] to provide safety guarantee under stochastic disturbance. The safety guarantee in the sense of probability is widely studied in [90,99,100,118,153]. The reciprocal stochastic control barrier function is applied to higher-order scenarios in [119], and [118] investigates worst-case safety verification using SZCBFs, regardless of the noise magnitude. In [138], the authors propose the stochastic control barrier function (SCBF) with milder conditions at the cost of sacrificing the almost sure safety.

1.3 Motivation

Safety-Critical Control for Unknown Deterministic System

In practice, models used to design controllers are usually imperfect because of system uncertainty, which could lead to unsafe and even hazardous situations. Therefore, it is essential to develop controllers that are capable of addressing these uncertainties. Learning-based methods have shown great potential in controlling systems subject to uncertainty [74]. Many methodologies leverage data-driven approaches in this field. This process involves collecting training data to better understand the real dynamics of the systems in order to design better controllers. In [152], a method involving the HOCBF under external disturbance is proposed using imitation learning to obtain a feedback controller. Techniques such as the Gaussian process (GP) have also been utilized to approximate the model, as in [32]. A non-parametric Gaussian control barrier function is introduced in [72] that is working on safety samples by assuming a GP prior. Reinforcement learning (RL) has been used to learn the model uncertainty for input-output linearization control [144] and a RL-based framework for policy improvement is proposed in [29] as well. However, both methods do not rely on a nominal controller. Using nominal controllers is more flexible, as they can be replaced by other reliable models and controllers in practice. In [129], the authors take the advantage of the nominal model and controllers and study safety-critical control of unknown deterministic systems. They estimate the derivative of the CBF using observed data. However, they only address relative degree of one. As a result, in Chapter 3, we extend our case into higher-order control barrier functions and propose a framework with higher relative degree.

Safety-Critical Control for Stochastic System

For systems influenced by Gaussian-type noise, stochastic differential equations (SDEs) are typically used to depict the randomness of such systems. Previous research on stochastic stability in the context of diffusion-type SDEs has a wide variety of applications in verifying probabilistic quantification of safe set invariance [80]. Investigations are focused on the worst-case safety verification, utilizing SZCBFs regardless of the intensity of noise. The authors in [118] have proposed a method for synthesizing polynomial state feedback controllers that achieve a specified probability of safety based on the existing verification results. In connection to stochastic hybrid systems with more complex specifications, a compositional framework is proposed by [107]. This framework focuses on the construction of control barrier functions for networks of continuous-time stochastic hybrid systems, enforcing complex logic specifications expressed by finite-state automata. However, applying conservative quantification to higher-order control systems is problematic due to low-quality estimation. The authors in [119] applied the strong set-invariance certificate (generated by SRCBFs) from [33] to higher-order stochastic control systems. The conditions are rather strong and able to effectively cancel the effects of diffusion to force a probability-one path safety. However, the above results admit unbounded control inputs, and the hard constraints may cause failures of satisfying safety specifications. As a result, we study safety-critical control of SDEs in Chapter 4 and propose our SCBF to improve worst-case safety probability with less control effort.

Safety-Critical Control for Stochastic System with Unknown Diffusion

Even though we propose the SCBF for safely controlling stochastic systems, there may be cases where we do not fully understand these systems. In some practical scenarios, we do not have precise information about the Brownian motion affecting the system. This lack of precise modeling of the diffusion presents a challenge. As a result, the first scenario we consider in the context of stochastic system uncertainties involves lacking information about the diffusion term. In such cases, we cannot directly calculate the generator of the SCBFs, which is required in the corresponding QP problems. In [98], the authors propose a method of estimating the value of the generator of a given function at a specific point within the domain. Building on this concept, we extend the idea to the entire state space and propose our data-driven stochastic control barrier functions (DDSCBF) framework for systems with unknown diffusion, as discussed in Chapter 5.

Safety-Critical Control for Fully Unknown Stochastic System

A natural extension of such uncertainties is for stochastic systems without any knowledge of either drift or diffusion. One way of handling these unknown dynamics is to identify the model first and then apply control barrier certificates to guarantee safety. GP models have been widely used to approximate unknown systems. Consequently, a Matrix-Variate Gaussian process model is used to learn non-linear control affine discrete systems and the learned model is incorporated into a multi-agent CBF framework as in [27]. A two-stage solution is proposed in [66] for unknown systems, where Gaussian learning is used to learn the model first and then use CBF for safety guarantee. In [40] and [17], Bayesian learning is used to approximate the unknown model with barrier certificates for safety-critical control. However, all of these methods only focus on deterministic systems. Data-driven methods with scenario convex programming (SCP) are also studied in guaranteeing safety of unknown dynamical systems. In [25] and [116], random trajectories are sampled and used with barrier certificates as a finite number of constraints in optimization problems for unknown deterministic and stochastic systems, respectively. However, such methods only apply to discrete systems. Formal methods with Gaussian process learning are also studied in [135] for discrete systems as well. On the other hand, data-driven methods with Bayesian learning has been widely used to identify unknown stochastic differential equations as in [9] and [120]. And recently, the authors in [22] proposed sparse Bayesian learning architecture to handle insufficient data resources or prior information for unknown SDEs. Motivated by such results, we propose a two-stage method using Bayesian inference in Chapter 6 for safety-critical control of fully unknown systems.

1.4 Contribution

This thesis addresses safety-critical control with various levels of uncertainties, primarily focusing on deterministic systems with imperfect information and stochastic systems influenced by Gaussian noise. The key contributions and outcomes for each chapter are outlined below.

1.4.1 Safety-Critical Control with Imperfect Deterministic Systems

In Chapter 3, we study safety-critical control for systems with imperfect information. We focus on higher relative degree and HOCBFs to develop a learning framework to deal with system uncertainty. Our method learns the derivatives of a HOCBF. We show that for each order, the derivatives of the HOCBF can be separated into the nominal derivative of the HOCBF and some remainders. This implies that we can use a neural network to learn the remainders so that we can approximate the real dynamics of the HOCBF. We also show that our learning framework maintains the control affine structure of the system and under some reasonable assumptions, the learned derivatives guarantee robustness for the safety of the system. In the simulation, we validate our method using Dubins vehicle and quadrotor [137].

As a result, the main contributions of Chapter 3 are summarized as follows .

- We propose a learning framework for derivatives of HOCBFs for safety-critical control for systems imperfect deterministic systems.
- We provide sufficient conditions on controllers via HOCBF for set invariance.
- We show the real derivatives of the HOCBF can be separated into two terms, i.e., the nominal derivative of the HOCBF and a remainder that is independent of the control input.
- We propose a control affine learning framework to approximate the real derivatives of the HOCBF.
- We validate our method in simulation using Dubins vehicle and quadrotor models.

1.4.2 Safety-Critical Control with Gaussian Noise

In Chapter 4, we study systems modeled by stochastic differential equations (SDEs) driven by Brownian motions. We introduce the concept of stochastic control barrier functions (SCBFs) and demonstrate that SCBFs can significantly reduce control efforts, especially in the presence of noise, compared to stochastic reciprocal control barrier functions (SRCBFs). The proposed SCBFs also provide a less conservative estimation of safety probability, compared to stochastic zeroing control barrier functions (SZCBFs). Motivated by the need to reduce potential severe control constraints generated by SRCBF certificates [33] in the neighborhood of the safety boundary, and improving the worst-case safety probability provided by SZCBF certificates, we propose SCBFs as a middle ground to characterize safety properties for systems driven by Brownian motions. We demonstrate that the proposed SCBFs achieve good trade-offs of performance and control efforts, both through theoretical analysis and numerical simulations. We show that SCBFs generate milder conditions compared to SRCBFs at the cost of sacrificing the almost-sure safety. The verification results, which provide a non-vanishing lower bound of safety probability for any finite time period, are still less conservative than widely used SZCBFs.

Unlike control systems with relative degree one, where optimal control schemes can be applied to synthesize finite-time almost-surely safety controllers [80, Chapter 5] or even to characterize the probabilistic winning set of finite-time safety with a priori probability requirement [47], off-the-shelf optimal control schemes and numerical tools cannot be straightforwardly applied to stochastic control systems with high relative degree. As a result, we further extend the results to handle high relative degree safety constraints using higher-order SCBFs [138].

The main contributions of Chapter 4 are summarized as follows.

- We propose the notion of stochastic control barrier functions (SCBFs) for safety-critical control. We show both theoretically and empirically that the proposed SCBFs achieve good trade-offs between mitigating the severe control constraints (potentially unbounded control inputs) and quantifying the worse-case safety probability.
- Based on the less conservative worse-case safety probability, we extend our result to higher-order SCBFs in handling high relative degree safety constraints and formally prove the worst-case safety probability.
- We validate our work in simulation and show that the proposed SCBFs with high relative degree have less control effort compared with SRCBFs and better safe probability compared with SZCBFs.

1.4.3 Data-Driven Learning of Safety-Critical Control with Stochastic Control Barrier Function

In real-world applications, dynamical systems sometimes encounter unknown disturbances in the form of Gaussian noise, of which the exact expressions are often unknown. Consequently, Chapter 5 studies the safety-critical control of stochastic systems with unknown diffusion parts, proposing a data-driven method to address these scenarios. Specifically, we propose a data-driven stochastic control barrier function (DDSCBF) framework and use supervised learning to learn the unknown stochastic dynamics via the DDSCBF scheme. Under some reasonable assumptions, we provide guarantees that the DDSCBF scheme can approximate the Itô derivative of the stochastic control barrier function (SCBF) under partially unknown dynamics using the universal approximation theorem. Furthermore, we demonstrate that the DDSCBF scheme can achieve the same safety guarantees as the SCBF approach presented in Chapter 4, despite that less knowledge of the stochastic system is accessible. We use two non-linear stochastic systems to validate our theory in simulation [139].

The principal contributions and outcomes of Chapter 5 can be summarized as follows.

- We study safety-critical control of stochastic systems with unknown diffusion term.
- We propose a data-driven method to approximate the generator of unknown SDEs.
- We provide guarantees that the DDSCBF scheme can approximate the Itô derivative of the stochastic control barrier function (SCBF) under partially unknown dynamics using the universal approximation theorem.
- We verify our results using two non-linear systems in simulation.

1.4.4 Safety-Critical Control with Unknown Drift and Diffusion

In Chapter 6, we extend our research to safety-critical control for unknown stochastic systems. This chapter extends the study from previous chapters on stochastic systems by addressing the additional challenge of unknown drift and diffusion terms. This exploration is important for real-world applications where system parameters are not always known a priori, and uncertainty can significantly impact system behavior. The proposed method takes the advantage of Bayesian learning to estimate the system's dynamics accurately. By leveraging data-driven methods, we effectively estimate both drift and diffusion terms based on the system's observations. This is a two-stage method where the system will be identified first and then the stochastic control barrier functions are applied to control the system safely. We validate our method in the simulation and compare results mainly in safety performance and time efficiency. For safety performance, we show that the estimated system could achieve a similar safety ratio compared to the true systems when using the stochastic control barrier functions. And as for the issue of time efficiency, the results highlight that identifying the system first using observed data is more efficient, especially in scenarios with higher relative degrees.

The main contributions of Chapter 6 are summarized as follows.

- We study safety-critical control for stochastic systems with completely unknown stochastic dynamics.
- We use Bayesian learning framework to identify the systems and then use SCBFs to control the identified system for safety guarantee.
- We validate our results and compare with our previous work in [138] and [139] for safety ratio and running time. We show that identifying the system with observed data first is more time-efficient when dealing with higher relative degree cases.

1.5 Overview of the Thesis

Below is a brief overview of our thesis addressing the problem of safety-critical control for uncertain systems. In Chapter 1, we talk about the foundation of our thesis, mainly about the background, motivation, literature review and our contribution of each chapter. We discuss the state of the art in safety-critical control area, which includes the exploration in both deterministic and stochastic systems and various type of (stochastic) barrier functions, which leads to our motivation of the research within our thesis.

In Chapter 2, we introduce the fundamental concepts related to each chapter of the thesis such as dynamical systems, set invariance, control Lyapunov functions, control barrier functions and higher-order control barrier functions. Stochastic differential equations together with different types of stochastic barrier functions are also discussed in this chapter. Additionally, we present the methodologies closely related to our proposed methods, such as supervised learning and (Bayesian) linear regression models.

Chapter 3 addresses safety-critical control for imperfect deterministic systems. We prove that the higher-order derivatives of HOCBFs can be isolated the derivative of such

HOCBF with respect to the nominal model and a remainder. We also show that our isolation architecture maintains the control affine property of the original system so that we could identify the remainder using learning techniques. Once we have learned the derivative of the higher-order derivative of the HOCBFs, we could control the system safely.

Chapter 4 focuses on safety-critical control in stochastic systems with Gaussian noise. We discuss different types of stochastic barrier certificates such as stochastic reciprocal and zeroing barrier functions. We propose our higher-order stochastic control barrier functions and show that our barrier certificates under stochastic scenarios require milder control actions while maintaining a reasonable worst-case safety probability.

The following Chapter 5 explores data-driven methods for safety-critical control using stochastic control barrier functions where the diffusion is unknown. We talk about probabilistic quantification of safety with respect to the unknown system using a learned generator. We also discuss the feasibility of our assumptions during the learning phase and provide robustness analysis of the resulting system.

And in Chapter 6, we also cover the challenge of safety-critical control for unknown stochastic systems. This chapter discusses safety-critical control strategies using stochastic control barrier functions, specifically for systems with unknown drift and diffusion. We emphasize our study mainly in discussing data collection and system identification techniques for accurately characterizing such unknown terms. Following this, the chapter introduces a control framework for the learned dynamics, integrating stochastic control barrier functions to guarantee safety.

Finally, in Chapter 7, we provide a thorough conclusion of our work, followed by insights into future research in safety-critical control for systems with uncertainty.

Chapter 2

Preliminaries

2.1 Dynamical System and Set Invariance

The field of dynamical systems and control plays a crucial role in understanding and manipulating the behavior of various systems. It involves the study of systems characterized by a state that changes over time. The primary goal is to develop and implement controllers that enable the system to execute specific tasks. Consider a nonlinear system as

$$\dot{X}(t) = F(X(t)), \quad X(t_0) = X_0, \tag{2.1}$$

where $X(t) \in \mathbb{R}^n$ is the state that evolves over time t, X_0 is the initial state at time t_0 and the function $F : \mathbb{R}^n \to \mathbb{R}^n$ is the vector field that governs the system's dynamics.

The concept of Lipschitz continuity is crucial in the context of differential equations. It ensures that small changes in the input lead to bounded changes in the output. This property is of great importance for the existence and uniqueness of solutions to differential equations.

Definition 2.1.1. *The function $F : \mathbb{R}^n \to \mathbb{R}^n$ is said to be Lipschitz continuous if there exists a positive number K such that*

$$|F(X_1) - F(X_2)| \leq K|X_1 - X_2|,$$

for all X_1 and X_2 in $\mathbb{R}^n$.

According to the existence and uniqueness theorem, if F is Lipschitz continuous, there exists a unique solution for system (2.1) for $t > 0$. This theorem guarantees that the system's behavior is well-defined and predictable under the given initial conditions.

The concept of positively invariant sets is important in the analysis and design of dynamical systems. A set $\mathcal{C}$ is called positively invariant for a system if the system starts in this set, it remains in this set for all future times. This property is critical in ensuring safety and stability in control systems, where certain states must be avoided.

Definition 2.1.2. *A set $\mathcal{C}$ is said to be positively invariant for system (2.1) if for all X_0, $X_0 \in \mathcal{C}$ implies $X(t) \in \mathcal{C}$ for all $t > 0$.*

In many practical scenarios, dynamical systems are driven by external controls and such systems are described by the following control affine systems

$$\dot{x} = f(x) + g(x)u, \quad x(t_0) = x_0, \tag{2.2}$$

with $x \in \mathcal{X} \subseteq \mathbb{R}^n$ and control input u taking values from $\mathcal{U} \subseteq R^m$. Controlled invariance extends the concept of set invariance to control systems. A set $\mathcal{C}$ is controlled invariant if, for every initial state in a certain set $\mathcal{O} \subseteq \mathcal{C}$, there exists a control input u that keeps the system's state within $\mathcal{C}$ for all future times. This is fundamental in control theory, as it allows the designer to ensure that the system remains within safe condition by applying appropriate control inputs.

2.1.1 Control Lyapunov Function and Stability

To motivate the idea of using barrier functions for safety control, we begin by considering the problem of stability control. Stability analysis for dynamical systems is crucial in control theory. It helps to determine whether a system will converge to a stable point over time. The Lyapunov functions are commonly used for stability analysis of dynamical systems. They provide a method to prove the stability of an equilibrium point in a system without explicitly solving the differential equations. A scalar function $V(x)$ is called Lyapunov function if it satisfying the following property as

- $V(x)$ is continuous and differentiable,
- $V(x) > 0$ for all $x \neq 0$ and $V(0) = 0$,
- $\dot{V}(x) = \frac{\partial V}{\partial x} \cdot f(x) \leq 0$ for all $x \neq 0$.

Control Lyapunov Functions (CLFs) are used to design control laws for guiding the system towards desired equilibrium points. It extends the concept of a Lyapunov function to systems with control inputs. To be more specific, given a control affine dynamical system as (2.2) and suppose that our control target is to control such system to reach a fixed point $x^* = 0$, we can use a CLF to drive the system to the origin. A function $V : \mathbb{R}^n \to \mathbb{R}$ is called a Control Lyapunov Function if it is positive definite and satisfies the condition that

$$\inf_{u \in \mathcal{U}} [L_f V(x) + L_g V(x) u] \leq -\gamma(V(x)), \tag{2.3}$$

where γ is a class $\mathcal{K}$ function, i.e., $\gamma(0) = 0$ and it is strictly monotonic. Taking the advantage of the above definition, we could design control laws according to

$$\exists u = k(x) \quad \text{s.t.} \quad \dot{V}(x, k(x)) \leq -\gamma(V(x)), \tag{2.4}$$

where

$$\dot{V}(x, k(x)) = L_f V(x) + L_g V(x) k(x),$$

to make sure that the system is stablizable at $x^* = 0$. Here, $L_f V(x) = f(x) \cdot \frac{\partial V}{\partial x}(x)$, $L_g V(x) = g(x) \cdot \frac{\partial V}{\partial x}(x)$.

We can consider the set of all possible stable controllers for every point in the state space as

$$K_{\text{clf}} := \{u \in \mathcal{U} : L_f V(x) + L_g V(x) u \leq -\gamma(V(x))\}. \tag{2.5}$$

Consequently, by finding control inputs that satisfying (2.4), we can control the system to reach $x^* = 0$.

2.1.2 Control Barrier Function and Safety

Barrier functions ensure safety constraints within dynamical systems. These functions are used to enforce hard constraints on system states, ensuring that certain undesirable or unsafe states are never reached. Barrier functions work by continuously monitoring the state of the system and dynamically adjusting the control inputs to ensure that the system does not break the safety barrier. The fundamental idea behind barrier functions is to create a 'barrier' within the state space that the system cannot cross. A barrier function $B : \mathbb{R}^n \to \mathbb{R}$ is typically designed such that it approaches infinity as the state approaches the boundary of the safe set.

Definition 2.1.3. *Given a safe set $\mathcal{C} \subseteq \mathbb{R}$, a function $B(x)$ is called a barrier function if*

- $B(x) > 0$ *for* $x \in \mathcal{C}$,

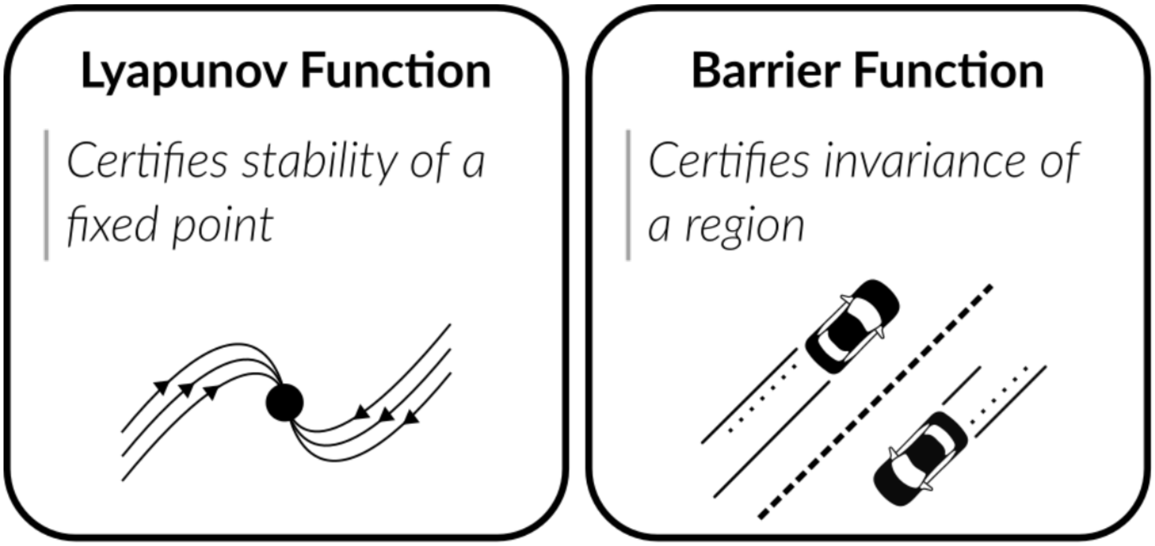


Figure 2.1. Control Lyapunov functions and control barrier functions. [38]

- $B(x) \to \infty$ *as* $x \to \partial\mathcal{C}$.

Inspired by control Lyapunov functions, safety can be described as finding controllers for system (2.2) to enforce the invariance of safe sets. More specifically, consider a set $\mathcal{C}$ defined as the superlevel set of a continuously differentiable function $h : D \subset \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$. Define

$$\begin{aligned} \mathcal{C} &= \{x \in D : h(x) \geq 0\}, \\ \partial\mathcal{C} &= \{x \in D : h(x) = 0\}, \\ \mathcal{C}^\circ &= \{x \in D : h(x) > 0\}. \end{aligned} \tag{2.6}$$

We aim to find feedback controllers $u = k(x)$ such that the resulting dynamical system

$$\dot{x} := f(x) + g(x)k(x)$$

is invariant in set $\mathcal{C}$. This means that for any initial state $x_0 \in \mathcal{C}$, $x(t) \in \mathcal{C}$ for $t \in [0, \infty)$.

Similar as control Lyapunov function, we define control barrier functions as

Definition 2.1.4. *Let* $h : D \subset \mathbb{R}^n \to \mathbb{R}$ *be a continuously differentiable function and* $\mathcal{C} \subset \mathcal{D} \subset \mathbb{R}^n$ *be a superlevel set of* h *as defined in* (2.6). *Then* h *is a control barrier* *function (CBF) if there exists an extended $\mathcal{K}_\infty$ function α such that for the control system* (2.2),

$$\sup_{u\in\mathbb{R}} [L_f h(x) + L_g h(x)u] \geq -\alpha(h(x))$$

for all $x \in D$, where $L_f h(x) = f(x) \cdot \frac{\partial h}{\partial x}(x)$ and $L_g h(x) = g(x) \cdot \frac{\partial h}{\partial x}(x)$.

Similar to (2.3), all controllers in

$$K_{\text{cbf}} := \{u \in \mathcal{U} : L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0\} \tag{2.7}$$

render $h \geq 0$, i.e., the safety of set $\mathcal{C}$.

Reciprocal Control Barrier Function and Zeroing Control Barrier Function

The definition of reciprocal control barrier function (RCBF) and zeroing control barrier functions (ZCBF) are first proposed in [6]. Given the function of the form

$$B(x) = -\log \frac{h(x)}{1 + h(x)},$$

with h defined as in Definition 2.6. The function satisfies the properties

$$\inf_{x\in\mathcal{C}^\circ} B(x) \geq 0,$$
$$\lim_{x\to\partial\mathcal{C}} B(x) = \infty.$$

The results in [132] and [107] show that the condition $\dot{B} \leq 0$ makes $\mathcal{C}^\circ$ forward invariant. It is shown in [6] that relaxing the condition to

$$\dot{B} \leq \frac{\alpha}{B}$$

also guarantees forward invariance of $\mathcal{C}^\circ$. As a result, the reciprocal barrier function (RBF) is defined as follows.

Definition 2.1.5. *Given a system as in* (2.1) *a continuously differentiable function $B : \mathcal{C}^\circ \to \mathbb{R}$ is a reciprocal barrier function (RBF) for the set $\mathcal{C}$ defined as in* (2.6) *for a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, if there exist class $\mathcal{K}$ functions α_1, α_2,α_3 such that , for all $x \in \mathcal{C}^\circ$,*

$$\frac{1}{\alpha_1(h(x))} \leq B(x) \leq \frac{1}{\alpha_2(h(x))},$$
$$L_F B(x) \leq \alpha_3(h(x)).$$

Based on the definition of RBF, the reciprocal control barrier function for a dynamical system as in (2.2) is defined as

Definition 2.1.6. *Given a control system as in* (2.2)*, a continuously differentiable function* $B : \mathcal{C}^\circ \to \mathbb{R}$ *is a reciprocal control barrier function (RCBF) for the set* $\mathcal{C}$ *defined as in* (2.6) *for a continuously differentiable function* $h : \mathbb{R}^n \to \mathbb{R}$*, if there exist class* $\mathcal{K}$ *functions* α_1*,* α_2*,*α_3 *such that , for all* $x \in \mathcal{C}^\circ$*,*

$$\frac{1}{\alpha_1(h(x))} \leq B(x) \leq \frac{1}{\alpha_2(h(x))},$$
$$\inf_{u \in \mathcal{U}} [L_f B(x) + L_g B(x) u - \alpha_3(h(x)) \leq 0].$$

However, the property that RBF/RCBF tends to infinity when approaching the boundary of $\mathcal{C}$ makes it undesirable for real-time applications. As a result, another barrier function and control barrier function are also considered in [6].

Definition 2.1.7. *A continuous function* $\alpha : (-b, a) \to (-\infty, \infty)$ *is said to belong to extended class* $\mathcal{K}$ *for some* $a, b > 0$ *if it is strictly increasing and* $\alpha(0) = 0$*.*

Definition 2.1.8. *Given a system as* (2.1)*, a continuously differentiable function* $h : \mathbb{R}^n \to \mathbb{R}$ *is a zeroing barrier function (RCBF) for the set* $\mathcal{C}$ *defined as in 2.6, if there exist an extended class* $\mathcal{K}$ *function* α *such that , for all* $x \in \mathcal{C}^\circ$*,*

$$L_F h(x) \geq -\alpha(h(x)). \tag{2.8}$$

This definition leads to the definition for zeroing control barrier function as below.

Definition 2.1.9. *Given a control system as in* (2.2) *a continuously differentiable function* $h : \mathbb{R}^n \to \mathbb{R}$ *is a zeroing control barrier function (ZCBF) for the set* $\mathcal{C}$ *defined as in* (2.6) *if there exist an extended class* $\mathcal{K}$ *function* α *such that , for all* $x \in \mathcal{C}^\circ$

$$\inf_{u \in \mathcal{U}} [L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0].$$

2.1.3 Exponential Control Barrier Function and Higher-Order Control Barrier Function

The above definitions of control barrier functions require that the first-order derivative of h is dependent of the control input u, which is not always the case in some real-time applications. As a result, we have to consider such control barrier functions of relative degree $r > 1$. The definition of relative degree is

Definition 2.1.10. *Given an r^{th}-order continuously differentiable function h and a system as defined in (2.2), we say h has a relative degree of r with respect to system (2.2) if $L_g L_f^{r-1} h(x) \neq 0$ and $L_g L_f h(x) = L_g L_f^2 h(x) = \cdots = L_g L_f^{r-2} h(x) = 0$, where $L_f^r h(x) = L_f L_f^{r-1} h(x)$.*

The r^{th}-order derivative of $h(x)$ is

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

which is dependent on the control input u. The system is input-output linearizable if $L_g L_f^{r-1} h(x)$ is invertible. For a given control $\mu \in \mathbb{R}$, u can be chosen such that $L_f^r h(x) + L_g L_f^{r-1} h(x) u = \mu$. The control input u renders the input-output dynamics of the system linear. Defining a system with state

$$\eta(x) := \begin{bmatrix} h(x) \\ \dot{h}(x) \\ \vdots \\ h^{r-1}(x) \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{r-1} h(x) \end{bmatrix},$$

we can then construct a state-transformed linear system

$$\begin{aligned} \dot{\eta}(x) &= F\eta(x) + G\mu, \\ h(x) &= C\eta(x), \end{aligned} \tag{2.9}$$

where

$$F = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$
$$C = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

The exponential control barrier function (ECBF) is defined below as in [101].

Definition 2.1.11. *Given a r^{th}-order continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ and a superlevel set $\mathcal{C}$ of h as defined in (2.6), then h is an exponential control barrier function (ECBF) if there exists a row vector $K = [k_0, k_1, \dots, k_{r-1}]$ such that*

$$\sup_{u \in \mathcal{U}} [L_f^r h(x) + L_g L_f^{r-1} h(x) u] \geq -K\eta(x).$$

for any $x \in \mathcal{C}$, where K is chosen such that the transformed system (2.9) is stable.

ECBFs can be seen as a special case of higher-order control barrier functions (HOCBFs) defined in [147]. In order to define HOCBFs, we have to first define a series of continuously differentiable functions $b_j : \mathbb{R}^n \to \mathbb{R}$ for $j = 1, 2, \ldots, r$ as

$$\begin{aligned} b_0(x) &= h(x), \\ b_j(x) &= \dot{b}_{j-1}(x) + c_j \alpha_j(b_{j-1}(x)), \end{aligned} \tag{2.10}$$

and the corresponding superlevel sets $\mathcal{C}_j$ as

$$\mathcal{C}_j = \{x \in \mathbb{R}^n : b_{j-1}(x) \geq 0\}, \tag{2.11}$$

where $c_j > 0$ are constants and $\alpha_j(\cdot)$ are differentiable extended class $\mathcal{K}$ functions.

Definition 2.1.12. *A continuously differentiable function h is an r^{th}-order control barrier function (HOCBF) for system (2.2), if there exists extended differentiable class $\mathcal{K}$ functions $\alpha_j(\cdot)$ for $j = 1, 2, ..., r$, such that for $b_j(x)$ defined in (2.10) with any arbitrary $c_j > 0$ and the corresponding superlevel sets $\mathcal{C}_j$ defined as in (2.11), the following*

$$\sup_{u \in \mathcal{U}} [L_f^r h(x) + L_g L_f^{r-1} h(x) u + \mathcal{O}(h)] \geq -c_r \alpha_r(b_{r-1}(x)) \tag{2.12}$$

holds for all $x \in \bigcap_{j=1}^r \mathcal{C}_j$, where $\mathcal{O}(h)$ denotes the Lie derivatives of h along f with degree up to $r-1$.

2.2 Stochastic System and Barrier Certificate

Stochastic systems are one kind of systems that is of great importance in practice ranging from engineering fields to financial scenarios. Different from deterministic systems, they are always driven by randomness and uncertainty with random variables which makes them difficult to analyze and control. Studying stochastic system is important, especially when dealing with safety-critical circumstances. Formally, consider a continuous-time stochastic process $X(t)$ over a filtered probability space in $\mathcal{X} \subseteq \mathbb{R}^n$ and a control input from a compact set $\mathcal{U} \subseteq \mathbb{R}^p$. The evolution of $X(t)$ is driven by a stochastic differential equation (SDE) as

$$dX_t = f(X_t)dt + g(X_t)u(t)dt + \sigma(X_t)dW_t, \tag{2.13}$$

with f and g the drift term and σ the diffusion term. The term $\sigma(X_t)dW_t$ introduces the stochastic noise with W_t a Brownian motion. A stochastic process X(t) is said to be a strong solution of an SDE if it satisfies

$$X_t = \xi + \int_0^t (f(X_s) + g(X_s)u(s))ds + \int_0^t \sigma(X_s)dW_s.$$

This definition emphasizes that the solution is constructed explicitly in terms of a given Brownian motion W. Additionally, the infinitesimal generator $\mathcal{A}$ of the stochastic process $X(t)$ w.r.t a continuous function $h(x)$ is defined as

$$\mathcal{A}h(x) = \lim_{t \to 0} \frac{\mathbb{E}^x[h(X_t)] - h(x)}{t},$$

where $\mathbb{E}^x(\cdot)$ represents the conditional expectation w.r.t an initial state x. Let X solves (2.13). According to Dynkin's formula [44], if $h \in C^2$, then

$$\mathcal{A}h(x) = \frac{\partial h}{\partial x}(f(x) + g(x)u(t)) + \frac{1}{2}\sum_{i,j}\left(\sigma\sigma^T\right)_{i,j}(x)\frac{\partial^2 h}{\partial x_i \partial x_j}.$$

Similar as CBFs in deterministic scenarios, stochastic barrier functions are used to ensure safety of SDEs in the presence of stochastic disturbance.

Definition 2.2.1. *Given a safe set $\mathcal{C}$, a function $B : \mathcal{C}^\circ \to \mathbb{R}$ is called a stochastic reciprocal control barrier function (SRCBF) [33] for system (2.13) if $B \in \mathcal{D}(\mathcal{A})$ and satisfies the following properties:*

(i) there exist class-$\mathcal{K}$ functions α_1, α_2 such that for all $x \in \mathcal{X}$ we have

$$\frac{1}{\alpha_1(h(x))} \le B(x) \le \frac{1}{\alpha_2(h(x))}; \tag{2.14}$$

(ii) there exists a class-$\mathcal{K}$ function α_3 such that

$$\inf_{u \in \mathcal{U}}[\mathcal{A}B(x) - \alpha_3(h(x))] \le 0, \tag{2.15}$$

where $\mathcal{D}(\mathcal{A})$ is the domain of the infinitesimal generator.

Such barrier certificate renders a control set based on (2.15)

$$k(x) := \{u \in \mathcal{U} : \mathcal{A}B(x) - \alpha_3(h(x)) \le 0\} \tag{2.16}$$

that guarantees safety for stochastic systems.

Stochastic zeroing barrier function, which is also introduced in [33] is another stochastic barrier certificate that guarantees a milder safety guarantee but requires more feasible control inputs. [1]

[1]See Chapter 4 for more detail.

Definition 2.2.2 (SZCBF)**.** *Given a safe set $\mathcal{C}$, a function $B : \mathcal{C}^\circ \to \mathbb{R}$ is called a stochastic zeroing control barrier function (SZCBF) for system* (2.13) *if $B \in \mathcal{D}(\mathcal{A})$ and*

(i) $B(x) \geq 0$ *for all* $x \in \mathcal{C}$*;*

(ii) $B(x) < 0$ *for all* $x \notin \mathcal{C}$*;*

(iii) there exists an extended $\mathcal{K}_\infty$ function α such that

$$\sup_{u \in \mathcal{U}} [\mathcal{A}B(x) + \alpha(B(x))] \geq 0. \tag{2.17}$$

Similar as SRCBF, we also refer the control strategy generated by (2.17) as

$$\kappa(x) := \{u \in \mathcal{U} : \mathcal{A}B(x) + \alpha(B(x)) \geq 0\} \tag{2.18}$$

for safety actions under stochastic noise.

2.3 Supervised Learning for System Identification

Supervised learning is a classical method in machine learning which aims to map input data to output data (labels) based on a set of training data. The fundamental goal of supervised learning it to construct a model by learning the patterns between the input and output data, and accurately make predictions of unseen data. Regression is one of the classical supervised learning method when the outputs of the model are some real values. Due to this property, regression models are widely used in system identification when we do not have full knowledge of the systems.

2.3.1 Linear Regression

Linear regression is a statistic model that is used to depict the linear relationship between a dependent variable and independent variables. A general form of the model can be described as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where

- $\mathbf{Y}$ is an $n \times 1$ matrix, where n is the number of observations.

- $\mathbf{X}$ is an $n \times (p+1)$ matrix, where p is the number of predictors.
- $\boldsymbol{\beta}$ is a $(p+1) \times 1$ matrix representing the regression coefficients, including both the slopes and the intercept.
- $\boldsymbol{\varepsilon}$ is an $n \times 1$ matrix (vector) representing the error term for each observation.

The coefficients $\boldsymbol{\beta}$ are optimized using ordinary least squares (OLS) method. This method minimizes the sum of squares of the difference between the observed data and predicted values as

$$J(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - x_i^T \boldsymbol{\beta})^2, \tag{2.19}$$

where:

- y_i is the i-th observation.
- x_i^T is the transpose of the i-th column of $\mathbf{X}$.

To find the values of $\boldsymbol{\beta}$ that minimize $J(\boldsymbol{\beta})$, we take the derivative of $J(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ and set it to zero. This leads to

$$\frac{\partial J}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) = 0. \tag{2.20}$$

Expanding and rearranging the terms, we get analytic solution of $\boldsymbol{\beta}$ as

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{Y},$$
$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y},$$

which provides the least squares estimates of the regression coefficients.

2.3.2 Update Coefficient with Gradient Descent Method

Gradient descent method is an optimizing algorithm that iteratively updates coefficients to obtain the estimation of model parameters. It is widely used in practice when analytic solutions are not feasible. By using a differentiable objective function, we could update the parameters such that the objective function moves towards its steepest descent. For linear regression models, people typically use the mean squared error (MSE) as in (2.19). The gradient descent algorithm updates the coefficients in the opposite direction of the gradient of the cost function. For each iteration, the coefficients are updated as follows based on

$$\boldsymbol{\beta} := \boldsymbol{\beta} - \alpha \frac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}},$$

where α is the learning rate, a hyperparameter that determines the size of the steps taken towards the minimum.

2.3.3 Feed-Forward Neural Network for Regression Problem

The feed-forward neural network (FFNN) is a classical artificial neural network that is widely used in practice. The term "feed-forward" means that there is no cycle within the structure of the network. As a result, it is widely used for regression problems in practice due to its simplicity and straightforwardness. The general architecture of an FFNN consists of an input layer, one or more hidden layers, and an output layer.

The network begins with an input matrix $\mathbf{X}$, which is of size $n \times p$, where n represents the number of samples in the input data and p is the dimension of the state. The network is composed of L hidden layers, with each layer containing some number of neurons. The l-th layer is denoted as $\mathbf{H}^{(l)}$.

At each layer of the network, there exists a linear transformation to the output of the previous layer, followed by a nonlinear activation function. Specifically, for the l-th layer, the process is characterized by the transformation $\mathbf{H}^{(l)} = f(\mathbf{W}^{(l)}\mathbf{H}^{(l-1)} + \mathbf{b}^{(l)})$, where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ represent the weight matrix and bias vector of the l-th layer, respectively, and f denotes the activation function.

This process accumulates till the last layer of the network, which is called the output layer. It is responsible for producing the predicted outputs, which is the network's estimation for regression purposes.

The network's learning process is the same as the linear regression, which involves updating the weights and biases to minimize the difference between the predicted outputs and the actual outputs. This is achieved by gradient descent method. The mean squared error (MSE) is a common cost function for regression, defined as

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2,$$

where $\hat{y}_i$ and y_i represent the predicted and actual output values for the i-th sample, respectively.

During training process, the weights $\mathbf{W}$ and biases $\mathbf{b}$ of the network are updated iteratively based on the update rule

$$\mathbf{W}^{(l)} := \mathbf{W}^{(l)} - \alpha \frac{\partial J}{\partial \mathbf{W}^{(l)}},$$
$$\mathbf{b}^{(l)} := \mathbf{b}^{(l)} - \alpha \frac{\partial J}{\partial \mathbf{b}^{(l)}},$$

for each layer l in the network.

This approach allows FFNNs to model complex relationships between input and output variables, making them a powerful tool for regression analysis in different problems.

2.3.4 Bayesian Learning and Bayesian Linear Regression

Bayesian learning is a probabilistic approach in machine learning that updates the model by integrating prior beliefs with evidence. The main asset behind Bayesian learning is Bayes' Theorem, which updates the probability of a hypothesis as more evidence becomes available according to

$$\mathbb{P}(H|D) = \frac{\mathbb{P}(D|H) \cdot \mathbb{P}(H)}{\mathbb{P}(D)},$$

where

- $\mathbb{P}(H|D)$ is the posterior probability of hypothesis H given data D.
- $\mathbb{P}(D|H)$ is the likelihood of data under hypothesis H.
- $\mathbb{P}(H)$ is the prior probability of hypothesis H.

- $\mathbb{P}(D)$ is the probability of the data.

The process of Bayesian learning involves starting with a prior distribution, observing data, and then updating this distribution to obtain the posterior distribution under all circumstances. However, in practice, this prior probability can be very subjective that relies on experience or previous studies. Additionally, calculating the likelihood of data can be challenging for complex models.

Bayesian linear regression is a statistical method that extends the classical linear regression framework by incorporating prior knowledge or beliefs about the parameters into the model. The main idea behind Bayesian linear regression is to update the model by using the above Bayes' Theorem. The parameters of the model are assumed with a prior distribution and will be updated when new observations are obtained. The key idea is to get the posterior distribution of parameters based on the prior distribution and likelihood of the observed data. Such posterior distribution provides a range of likely values for the parameters, taking into account both the prior distribution and the observed data.

Formally, the model incorporates a base function Φ, and a weight vector ϑ, where Φ is a matrix in $\mathbb{R}^{N\times M}$ and ϑ is a vector in $\mathbb{R}^{M\times 1}$. A Bayesian linear model is written as

$$Y = \Phi\vartheta + \varepsilon,$$

where ε is the noise component, following a Gaussian distribution $\mathcal{N}(0, \Psi)$, with Ψ being a diagonal matrix with the i-th element being σ^2/K. Initially, a Gaussian prior distribution with zero mean and covariance matrix Σ_0 is assigned to ϑ. The posterior distribution of ϑ is determined using Bayes' theorem

$$P(\vartheta|Y) \propto p(\vartheta)p(Y|\vartheta),$$

with $p(Y|\vartheta)$ as the likelihood function. Under the assumption of error terms being independent and identically distributed, the likelihood is

$$P(Y|\vartheta) = \mathcal{N}(\Phi\vartheta, \sigma^2 I/K),$$

where I is an n-dimensional identity matrix. The posterior distribution is computed as

$$P(\vartheta|Y) = \mathcal{N}(\bar{\vartheta}, \bar{\Sigma}),$$

with $\bar{\vartheta}$ and $\bar{\Sigma}$ being the MAP estimation and the posterior covariance matrix, respectively, calculated by

$$\bar{\vartheta} = (\Phi^T\Phi + \frac{\sigma^2}{K}\Sigma_0^{-1})^{-1}\Phi^T Y$$

and

$$\bar{\Sigma} = \frac{\sigma^2}{K}(\Phi^T\Phi + \frac{\sigma^2}{K}\Sigma_0^{-1})^{-1}.$$

Chapter 3

Safety-Critical Control with Imperfect Deterministic Systems

In practice, we my suffer from scenarios in which the information of the system is imperfect. This might be caused by inaccurate estimation of the model. As a result, in this chapter, we focus on deterministic systems with imperfect model.

3.1 Preliminary and Problem Definition

3.1.1 Model and Uncertainty

Throughout the chapter, we consider a SISO nonlinear control affine model

$$\begin{aligned} \dot{x} &= f(x) + g(x)u, \\ y &= h(x), \end{aligned} \tag{3.1}$$

such that $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times 1}$ are locally Lipschitz, $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state and $u \in \mathcal{U} \subseteq \mathbb{R}$ is the control and $h : \mathbb{R}^n \to \mathbb{R}$ is an r^{th}-order continuously differentiable function for some integer r.

Given a control signal $u : [0, \infty) \to \mathbb{R}$, we make a bit abuse of notation and denote the unique solution of (3.1) starting from x_0 and defined on the maximal interval of existence $\mathbb{T} := [0, \tau_{\max}(x_0, u))$ by $x(t; x_0, u)$. We may also use the shorthand notations $x(t; x_0)$ or $x(t)$ or x if the arguments are not emphasized under clear contexts.

We also consider a nominal model that estimates the dynamics of the true system (3.1) as

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

where $\hat{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\hat{g} : \mathbb{R}^n \to \mathbb{R}^{n\times 1}$ are locally Lipschitz continuous.

3.1.2 Problem Formulation

The objective of this chapter is to control a nonlinear system (3.1) with imperfect information to reach a given target set while ensuring safety, i.e., staying inside a safe set. We assume that the nominal model (3.2) is known and there exists a nominal feedback controller such that the closed-loop system can safely reach the target set. Then the problem is formally formulated as below.

Problem 1. *Given system as in (3.1), a goal region $\mathcal{X}_{goal} \subset \mathcal{X}_{safe}$, a safe set $\mathcal{X}_{safe} \subset \mathbb{R}^n$, a nominal controller $k(x)$, and an initial state x_0, design a feedback controller $u = \tilde{k}(x)$, where $\tilde{k} : \mathcal{X}_{safe} \to \mathbb{R}$, such that the solution of the closed-loop system satisfies that]*

3.1.3 Control Barrier Function

We consider a set $\mathcal{C}$ defined as a superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ such that

$$\begin{aligned} \mathcal{C} &= \{x \in \mathbb{R}^n : h(x) \geq 0\}, \\ \partial\mathcal{C} &= \{x \in \mathbb{R}^n : h(x) = 0\}, \\ \mathcal{C}^\circ &= \{x \in \mathbb{R}^n : h(x) > 0\}. \end{aligned} \tag{3.3}$$

We refer $\mathcal{C}$ as the safe set and safety can be framed in the context of enforcing invariance of $\mathcal{C}$. We can define a set to be forward invariant as below.

Definition 3.1.1. *Let $h : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function and $\mathcal{C} \subset \mathbb{R}^n$ be a zero superlevel set of h as defined in (3.3). The set is forward invariant with respect to input signal u if the resulting trajectory satisfies $x(t) \in \mathcal{C}$ for each $x_0 \in \mathcal{C}$ for all $t \in \mathbb{T}$. The system (3.1) is safe with respect to $\mathcal{C}$ if $\mathcal{C}$ is forward invariant.*

We denote by $\mathcal{K}_\infty$ the class of extended class $\mathcal{K}$ functions, which contains continuous functions $\alpha : \mathbb{R} \to \mathbb{R}$ that are strictly increasing and $\alpha(0) = 0$. Based on this, we can define the control barrier function as follows.

Definition 3.1.2. *Let $h : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function and $\mathcal{C} \subset \mathcal{D} \subset \mathbb{R}^n$ be a superlevel set of h as defined in (3.3). Then h is a control barrier function (CBF) if there exists an extended $\mathcal{K}_\infty$ function α such that for the control system (3.1),*

$$\sup_{u \in \mathcal{U}} [L_f h(x) + L_g h(x) u] \geq -\alpha(h(x))$$

for all $x \in \mathcal{D}$, where $L_f h(x) = f(x) \cdot \frac{\partial h}{\partial x}(x)$ and $L_g h(x) = g(x) \cdot \frac{\partial h}{\partial x}(x)$.

We can then consider the set consisting of all control values that render $\mathcal{C}$ to be safe [4]:

$$K_{\text{cbf}}(x) = \{u \in \mathcal{U} : L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0\}.$$

3.1.4 Safety-Critical Control

Suppose we are given a feedback controller $u = k(x)$ for the system (3.1) and we wish to control the system while guaranteeing safety. It may be the case that sometimes the feedback controller $u = k(x)$ is not safe, i.e., there exists some x such that $u(x) \notin K_{\text{cbf}}(x) = \{u \in \mathcal{U} : L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0\}$. We can use the following quadratic programming to find the safe control with minimum perturbation [8]:

$$\begin{aligned} u(x) = \arg\min_{u \in \mathcal{U}} \ & \frac{1}{2} ||u - k(x)||^2 \qquad \text{(CBF-QP)} \\ \text{s.t.} \quad & L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0. \end{aligned}$$

3.1.5 Relative Degree and Exponential Control Barrier Function

The relative degree of a continuous differentiable function h on a set with respect to a system as in (3.1) is the number of times we need to differentiate h along the dynamics of the system before the control input u explicitly appears. The formal definition of relative degree is as below.

Definition 3.1.3. *Given an r^{th}-order continuously differentiable function h, a set D and a system as defined in (3.1), we say h has a relative degree of r with respect to system (3.1) on D if $L_g L_f^{r-1} h(x) \neq 0$ and $L_g L_f h(x) = L_g L_f^2 h(x) = \cdots = L_g L_f^{r-2} h(x) = 0$ for $x \in D$, where $L_f^r h(x) = L_f L_f^{r-1} h(x)$.*

Remark 3.1.4. *In this chapter, we assume that h has a well-defined relative degree of r with respect to system (3.1) on a domain D of interest, similar to [150], where the author assumed $D = \mathbb{R}^n$.*

The r^{th}-order time-derivative of $h(x)$ is

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

and $h^{(r)}(x)$ is dependent on the control input u. The system is input-output linearizable if $L_g L_f^{r-1} h(x)$ is invertible. For a given control $\mu \in \mathbb{R}$, u can be chosen such that $L_f^r h(x) + L_g L_f^{r-1} h(x) u = \mu$. The control input u renders the input-output dynamics of the system linear. Defining a system with state

$$\eta(x) := \begin{bmatrix} h(x) \\ \dot{h}(x) \\ \vdots \\ h^{(r-1)}(x) \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{r-1} h(x) \end{bmatrix}, \tag{3.4}$$

we can then construct a state-transformed linear system

$$\begin{aligned} \dot{\eta}(x) &= F\eta(x) + G\mu, \\ h(x) &= C\eta(x), \end{aligned} \tag{3.5}$$

where

$$F = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$
$$C = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

The exponential control barrier function is defined below as in [101].

Definition 3.1.5. *Given an r^{th}-order continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ and a superlevel set $\mathcal{C}$ of h as defined in (3.3), then h is an exponential control barrier function (ECBF) if there exists a row vector $K = [k_0, k_1, \dots, k_{r-1}]$ such that*

$$\sup_{u \in \mathcal{U}} [L_f^r h(x) + L_g L_f^{r-1} h(x) u] \geq -K \cdot \eta(x). \tag{3.6}$$

for any $x \in \mathcal{C}$, where K is chosen such that the transformed system (3.5) is stable.

Remark 3.1.6. *It is explained in [101] that the ECBF with $r = 1$ is the same as the CBF as in Definition 3.1.2. The design of the ECBF, i.e, the selection of $k_0, k_1, \dots, k_{r-1}$ in K is also explained in [101] using state feedback control and pole placement.*

As a result, given an ECBF and a nominal controller $u = \kappa(x)$, we can consider the following quadratic programming problem to enforce the condition in Definition 3.1.5 with minimum perturbation

$$\begin{aligned} u(x) = \arg\min_{u \in \mathcal{U}} \frac{1}{2}||u - \kappa(x)||^2 \qquad \text{(ECBF-QP)} \\ \text{s.t.} \quad L_f^r h(x) + L_g L_f^{r-1} h(x) u \geq -K \cdot \eta(x). \end{aligned}$$

3.1.6 Higher-Order Control Barrier Function and Controlled Set Invariance

Exponential control barrier functions (ECBF) can be seen as a special case of higher-order control barrier functions (HOCBF) defined in [147]. In this section, we present some sufficient conditions on using HOCBF for enforcing set invariance. We first define a series of continuously differentiable function $b_0, b_j : \mathbb{R}^n \to \mathbb{R}$ for each $j = 1, 2, \ldots, r$ and corresponding superlevel sets $\mathcal{C}_j$ as

$$\begin{aligned} b_0(x) &= h(x), \\ b_j(x) &= \dot{b}_{j-1}(x) + c_j \alpha_j(b_{j-1}(x)), \end{aligned} \tag{3.7}$$

and

$$\mathcal{C}_j = \{x \in \mathbb{R}^n : b_{j-1}(x) \geq 0\}, \tag{3.8}$$

where $c_j > 0$ are constants and $\alpha_j(\cdot)$ are differentiable extended class $\mathcal{K}$ functions. We further assume that the interiors of the sets $\mathcal{C}_i$ are given by

$$\mathcal{C}_i^\circ = \{x \in \mathbb{R}^n : b_{j-1}(x) > 0\}.$$

Definition 3.1.7. *A continuously differentiable function h is an r^{th}-order control barrier function (HOCBF) for system (3.1), if there exists extended differentiable class $\mathcal{K}$ functions $\alpha_j(\cdot)$ for $j = 1, 2, ..., r$, such that for $b_j(x)$ defined in (3.7) with any arbitrary $c_j > 0$ and the corresponding superlevel sets $\mathcal{C}_j$ defined as in (3.8), the following*

$$\sup_{u \in \mathbb{R}} [L_f^r h(x) + L_g L_f^{r-1} h(x) u + \mathcal{O}(h)] \geq -c_r \alpha_r(b_{r-1}(x)) \tag{3.9}$$

holds for all $x \in \bigcap_{j=1}^r \mathcal{C}_j$, where $\mathcal{O}(h)$ denotes the Lie derivatives of h along f with degree up to $r - 1$.

Remark 3.1.8. *Note that $\mathcal{C}_1$ is uniquely defined, whereas $\mathcal{C}_2, \mathcal{C}_3, ..., \mathcal{C}_r$ is defined based on the choice of $c_1, c_2, ..., c_{r-1}$.*

Proposition 3.1.9. *Consider r^{th}-order HOCBF $h : \mathbb{R}^n \to \mathbb{R}$ with the associated α_j and sets $\mathcal{C}_i$ for $j \in \{1, 2, ..., r\}$. Suppose that h has relative degree r with respect to system (3.1) on a set D containing $\bigcap_{j=1}^{r} \mathcal{C}_j$. Then any Lipschitz continuous controller $u(x)$ that satisfies*

$$L_f^r h(x) + L_g L_f^{r-1} h(x) u(x) + \mathcal{O}(h) \geq -c_r \alpha_r(b_{r-1}(x)) \tag{3.10}$$

for all $x \in \bigcap_{j=1}^{r} \mathcal{C}_j^\circ$ renders the set $\bigcap_{j=1}^{r} \mathcal{C}_j^\circ$ forward invariant. Furthermore, given any functions α_j, $j \in \{1, 2, ..., r\}$, and any compact initial set $X_0 \subset \mathcal{C}_1^\circ$, there exist appropriate choices of $c_j > 0$ such that $X_0 \subset \bigcap_{j=1}^{r} \mathcal{C}_j^\circ$.

Before proceeding to the proof, we introduce technical tools to show how the invariance conditions is effective for first order barrier functions. We first cite a lemma from [59], which can be proved based on Lemma 4.4 in [71] and well-known comparison techniques [82].

Lemma 3.1.10. *[59] Let $z : [t_0, t_f) \to \mathbb{R}$ be a continuously differentiable function satisfying the differential inequality*

$$\dot{z}(t) \geq -\alpha(z(t)),\ \forall t \in [t_0, t_f), \tag{3.11}$$

where $\alpha : \mathbb{R} \to \mathbb{R}$ is a locally Lipschitz extended class $\mathcal{K}$ function. Then there exists a class $\mathcal{KL}$ function $\beta : [0, \infty) \times [0, \infty) \to [0, \infty)$ (only depending on α) such that

$$z(t) \geq \beta(z(t_0), t - t_0), \quad \forall t \in [t_0, t_f).$$

Corollary 3.1.11. *Given a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ and dynamics on $\mathbb{R}^n$*

$$\dot{x} = f(x) \tag{3.12}$$

such that $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz. Let $\mathcal{C} = \{x :\ h(x) \geq 0\}$, and $\mathcal{C}^\circ := \{x :\ h(x) > 0\}$. If the Lie derivative of h along the trajectories of x satisfies

$$\dot{h}(x) \geq -\alpha(h(x)),\ \forall x \in \mathcal{C}, \tag{3.13}$$

where α is a locally Lipschitz extended class $\mathcal{K}$ function, then the set $\mathcal{C}^\circ$ is forward invariant.

Proof. If $\mathcal{C}^\circ = \emptyset$, then it is invariant. Otherwise, we apply Lemma 3.1.10, it follows that if $x(t_0) \in \mathcal{C}^\circ$, then we have $h(x(t)) > 0$ for all $t \in [t_0, t_f)$, where $[t_0, t_f)$ is the maximal interval of existence for $x(t)$ starting from $x(t_0)$. □

Remark 3.1.12. *Note that the result cannot be extended to the invariance of the set $\mathcal{C}$, despite that it is widely stated so in the literature. A simple counterexample is when $\mathcal{C}^\circ = \emptyset$, we can define $h(x) = -x^2$ and therefore $\mathcal{C} = \{0\}$. Then for $\dot{x} = c \neq 0$, even though we have a satisfaction of (3.13) on $\mathcal{C} = \{0\}$, it is not invariant under the flow.*

Now assume $\mathcal{C}^\circ \neq \emptyset$, we also need to necessarily assume the locally Lipschitz continuity of α. As for a counter example, let $\dot{x} = -1$ and $h(x) = \frac{2\sqrt{2}}{3\sqrt{3}} x^{3/2}$ for $x \geq 0$. Then $\dot{B}(x) = -B^{1/3}(x)$ satisfies the condition on $\mathcal{C}$. However, the point 0 loses asymptotic behavior and $h(x)$ will reach 0 within finite time for any $x_0 > 0$.

Proof of Proposition 3.1.9:

By the choice of controller $u(x)$ in (3.10), we have

$$b_r(x) = \dot{b}_{r-1}(x) + c_r \alpha_r(b_{r-1}(x)) \geq 0 \tag{3.14}$$

for all $x \in \bigcap_{j=1}^r \mathcal{C}_j$. Suppose $x_0 \in \bigcap_{j=1}^r \mathcal{C}_j^\circ$. Then there exists a small time $\tau > 0$ such that the solution to (3.1) under the controller $u(x)$ is defined on $[0, \tau]$ and $x(t) \in \bigcap_{j=1}^r \mathcal{C}_j^\circ$ for all $t \in [0, \tau]$. The differentiability of α_j implies its local Lipschitz continuity. By (3.14) and Lemma 3.1.10, we have $b_{r-1}(x(t)) > 0$ for all $t \in [0, \tau]$. By the same argument, we can show that $b_j(x(t)) > 0$ for all $t \in [0, \tau]$ and all $j = 0, 1, \cdots, r-1$. To conclude that $x(t) \in \bigcap_{j=1}^r \mathcal{C}_j^\circ$ for all t in the maximal interval of existence of $x(t)$, we can use the fact that the $\mathcal{KL}$ lower bound given by Lemma 3.1.10 only depends on α_j's.

As for any given α_j and $X_0 \subset \mathcal{C}_j^\circ$, since $L_f^j h$, $\mathcal{O}_{j-1}(h)$, α_j for $j \in \{1, ..., r\}$ and $L_g L_f^{r-1} hu$ are all continuous functions, we can recursively define

$$c_j > \max(-\frac{L_f^j h(x_0) + \mathcal{O}_{j-1} h(x_0)}{\alpha_j(b_{j-1}(x_0))}, \delta_j)$$

for arbitrary $\delta_j > 0$ from $j = 1$ to $j = r-1$. Similarly, we choose

$$c_r > \max(-\frac{L_f^j h(x_0) + L_g L_f^{r-1} h(x_0) u(x_0) + \mathcal{O}_{r-1} h(x_0)}{\alpha_r(b_{r-1}(x_0))}, \delta_r).$$

The above choice of c_j for $j \in \{1, ..., r\}$ guarantees $b_j(x_0) = \dot{b}_{j-1}(x_0) + c_j \alpha_j(b_{j-1}(x_0)) > 0$, or equivalently $X_0 \subset \bigcap_{j=1}^r \mathcal{C}_j^\circ$. ■

Remark 3.1.13. *Sufficient conditions for enforcing set invariance using ECBF or HOCBF can be found in [101] and [147], respectively (see also [4]). The first part of Proposition 3.1.9 recaptures the results in [4, 101, 147], but we spell out the importance of the local*

Lipschitz condition on α_j's and the fact that the set $\bigcap_{j=1}^{r} \mathcal{C}_j$ itself may not be controlled invariant under the well-known (zeroing) CBF condition (see Remark 3.1.12 above) even for the case $r = 1$ without further assumptions. The second part of Proposition 3.1.9 is in case that by a given HOCBF h, the initial point $x_0 \notin \bigcap_{j=1}^{r} \mathcal{C}_j$. However, one can always rescale the existing α_j with proper choices of c_j to provide invariance conditions, such that controllers adjusted to the conditions will lead the trajectories starting from any compact initial set $X_0 \subset \mathcal{C}_1^\circ$ invariant within $\mathcal{C}_1$.

3.2 Model Uncertainty

In this section, we discuss how we deal with model uncertainty and guarantee safety of the real system using the nominal model. We consider a real model as in (3.1), where f and g are not known precisely in practice and a nominal model as in (3.2) that estimates the true dynamics of the system. Then we can rewrite the real model based on the nominal model as

$$\dot{x} = \hat{f}(x) + \hat{g}(x)u + b(x) + A(x)u, \tag{3.15}$$

where $b(x) = f(x) - \hat{f}(x)$ and $A(x) = g(x) - \hat{g}(x)$.

Proposition 3.2.1. *Given a nominal model and a real model as in (3.2) and (3.15), respectively, a control barrier function h with relative degree r on a set D, we have*

$$h^{(m)}(x) = \hat{h}^{(m)}(x) + \Delta_m(x), \quad x \in D, \tag{3.16}$$

for $m = 1, 2, \ldots, r-1$, where $\hat{h}^{(m)} = \nabla \hat{h}^{(m-1)} \cdot (\hat{f}(x) + \hat{g}(x)u)$ with $\hat{h}^{(0)} = h$, and $\Delta_m(x)$ is the remainder term. Both terms in (3.16) are independent of the control input u.

Proof. We use mathematical induction to prove the result.

For $m = 1$, we have

$$\begin{aligned} h^{(1)}(x) &= \frac{\partial h}{\partial x} \cdot (\hat{f}(x) + \hat{g}(x)u + b(x) + A(x)u) \\ &= L_{\hat{f}} h(x) + L_{\hat{g}} h(x)u + L_b h(x) + L_A h(x)u. \end{aligned}$$

Since h is with higher-order relative degree $r > 1$ with respect to the nominal model and uncertainty, we have $L_{\hat{g}} h(x) = L_A h(x) = 0$, as a result,

$$\begin{aligned} h^{(1)}(x) &= L_{\hat{f}} h(x) + L_b h(x) \\ &= \hat{h}^{(1)}(x) + L_b h(x) \\ &= \hat{h}^{(1)}(x) + \Delta_1(x), \end{aligned}$$

where $\Delta_1(x) = L_b h(x)$. We can see that for $m = 1$, (3.16) holds and $\Delta_1(x)$ is independent of u. Now assume that (3.16) holds for $m = k$ and $\Delta_m(x)$ is independent of u, then for $m = k + 1$, we have

$$\begin{aligned}
&h^{(k+1)}(x) \\
&\quad = \frac{(\partial \hat{h}^k(x) + \Delta_k(x))}{\partial x} \cdot (\hat{f}(x) + \hat{g}(x)u + b(x) + A(x)u) \\
&\quad = L_{\hat{f}}^{k+1} h(x) + L_{\hat{g}} L_{\hat{f}}^k h(x) u + L_b L_{\hat{f}}^k h(x) + L_A L_{\hat{f}}^k h(x) u \\
&\qquad\qquad + \frac{\partial \Delta_k(x)}{\partial x} \cdot (\hat{f}(x) + \hat{g}(x)u + b(x) + A(x)u)).
\end{aligned}$$

Since the relative degree of h with respect to the real model, nominal model and uncertainty are all r, for $m = k + 1 < r - 1$, we have $L_{\hat{g}} L_{\hat{f}}^k h(x) u = L_A L_{\hat{f}}^k h(x) u = 0$, $L_{\hat{f}}^{k+1} h(x) = \hat{h}^{(k+1)}$ and $L_{\hat{g}}(\frac{\partial \Delta_k(x)}{\partial x}) = L_A(\frac{\partial \Delta_k(x)}{\partial x}) = 0$. As a result,

$$h^{(k+1)}(x) = \hat{h}^{(k+1)}(x) + \Delta_{k+1}(x)$$

such that

$$\begin{aligned}
\Delta_{k+1}(x) &= L_b L_{\hat{f}}^k h(x) \\
&\quad + \frac{\partial \Delta_k(x)}{\partial x} \cdot (\hat{f}(x) + \hat{g}(x)u + b(x) + A(x)u) \\
&= L_b L_{\hat{f}}^k h(x) + \frac{\partial \Delta_k(x)}{\partial x} \cdot (\hat{f}(x) + b(x)) \\
&= L_b L_{\hat{f}}^k h(x) + L_{\hat{f}}(\frac{\partial \Delta_k(x)}{\partial x}) + L_b(\frac{\partial \Delta_k(x)}{\partial x}).
\end{aligned}$$

This means that for $m = k + 1$, the equation $h^{(k+1)}(x) = \hat{h}^{(k+1)}(x) + \Delta_{k+1}(x)$ also holds and $\Delta_{k+1}(x)$ is independent of control input u. □

The above proposition shows that for $m = 1, 2, \cdots, r - 1$, we can always separate the time derivative of the CBF for the real system into the time derivative of the CBF for the nominal system and a remainder. As a result, for $m = r$:

$$
\begin{aligned}
h^r(x,u) &= \frac{\partial(L_{\hat{f}}^{r-1}h(x) + \Delta_{r-1}(x))}{\partial x} \cdot (\hat{f}(x) + \hat{g}(x)u \\
&+ b(x) + A(x)u) \\
&= L_{\hat{f}}^{r}h(x) + L_{\hat{g}}L_{\hat{f}}^{r-1}h(x)u + L_b L_{\hat{f}}^{r-1}h(x) \\
&+ L_A L_{\hat{f}}^{r-1}h(x)u + \frac{\partial \Delta_{r-1}(x)}{\partial x} \cdot (\hat{f}(x) + b(x)) \\
&+ \frac{\partial \Delta_{r-1}}{\partial x}(x) \cdot (\hat{g}(x) + A(x))u \\
&= \hat{h}^{(r)}(x,u) + \Delta_r(x) + \Sigma_r(x)u,
\end{aligned}
\tag{3.17}
$$

where $\Delta_r(x) = \frac{\partial \Delta_{r-1}(x)}{\partial x} \cdot (\hat{f}(x) + b(x)) + L_b L_{\hat{f}}^{r-1}h(x)$ and $\Sigma_r(x) = L_A L_{\hat{f}}^{r-1}h(x) + \frac{\partial \Delta_{r-1}(x)}{\partial x} \cdot (\hat{g}(x) + A(x))$. According to the above conclusion, we know that the higher-order time derivative of the real CBF $h^{(r)}$ can be separated into the higher-order time derivative of the nominal CBF $\hat{h}^r$ and a remainder $\Delta_r + \Sigma_r u$. To sum up, we can write our conclusion in the following representation as

$$
\begin{bmatrix} h^{(1)}(x) \\ h^{(2)}(x) \\ \vdots \\ h^{(r-1)}(x) \\ h^{(r)}(x,u) \end{bmatrix} = \begin{bmatrix} \hat{h}^{(1)}(x) + \Delta_1(x) \\ \hat{h}^{(2)}(x) + \Delta_2(x) \\ \vdots \\ \hat{h}^{(r-1)}(x) + \Delta_{r-1}(x) \\ \hat{h}^{(r)}(x,u) + \Delta_r(x) + \Sigma_r(x)u \end{bmatrix}
\tag{3.18}
$$

.

3.3 Neural Control Barrier Function Scheme for Safety-Critical Control

In order to use the Proposition 3.1.9 and the controller that satisfies (3.10) or the special version (3.6) for the safety guarantees, we need to gather information of Δ_m for $m < r$ as well as Δ_r and Σ_r.

However, due to the lack of information of the system, we are unable to capture the correction information. We use a data-driven method to approximate the vector field $\varphi := [\Delta_1, \Delta_2, \cdots, \Delta_r, \Sigma_r]$, and impose similar barrier conditions on the approximated $\tilde{\varphi}$

for safety-critical control. Based on the partial observation of data, we show in this section that a degree of robustness in the barrier condition is necessary to balance the inaccuracy of data. We use the ECBF condition (3.6) for simplicity to illustrate the idea.

Let $\tilde{\varphi}$ be the approximation of φ based on some training data $\mathfrak{C} \subset \mathcal{C}$. Let $\|\cdot\|$ denote the infinity norm of a finite-dimensional vector space. We make the following extra assumptions for the rest of analysis.

Assumption 3.3.1. *We assume that*

(1) The safe set $\mathcal{C}$ is compact.

(2) For any $\varepsilon > 0$, there exists a training set $\mathfrak{C}$ that is sufficiently dense in $\mathcal{C}$, and a neural network $\tilde{\varphi}$ based on $\mathfrak{C}$ such that

$$\sup_{x\in\mathcal{C}} \|\varphi(x) - \tilde{\varphi}(x)\| \leq \varepsilon. \tag{3.19}$$

Furthermore, we assume that both $\tilde{\varphi}$ and φ are Lipschitz continuous on the compact set $\mathcal{C}$.

Proposition 3.3.2. *Let $h : \mathbb{R}^n \to \mathbb{R}$ be an r^{th}-order continuously differentiable function whose time-derivatives satisfy (3.18). Recall notations in (3.18). Let $\varphi := [\Delta_1, \Delta_2, \cdots, \Delta_r, \Sigma_r]$ be approximated by a neural network $\tilde{\varphi} := [\tilde{\Delta}_1, \tilde{\Delta}_2, \cdots, \tilde{\Delta}_r, \tilde{\Sigma}_r]$ such that Assumption 5.2.1 is held. Let $\tilde{h}^{(r)}(x,u) = \hat{h}^{(r)}(x,u) + \tilde{\Delta}_r + \tilde{\Sigma}_r u$. Then, h is an ECBF (recall Definition 3.1.5) if there exists a positive row vector $K = [k_0, k_1, \cdots, k_{r-1}]$ such that*

$$\sup_{u\in\mathcal{U}}[\tilde{h}^{(r)}(x,u)] \geq -K\cdot\tilde{\eta}(x) + \varepsilon\left(1 + \sum_{i=1}^{r-1} k_i\right) + \varepsilon \max_{u\in\mathcal{U}} |u|, \tag{3.20}$$

where $\tilde{\eta}(x) = [h(x), \tilde{h}^{(1)}(x), \tilde{h}^{(2)}(x), \cdots, \tilde{h}^{(r-1)}(x)]^T$, and ε is an arbitrarily small parameter characterizing the universal approximation as described in (2) of Assumption 5.2.1.

Proof. By the hypothesis, it follows that, for all $x \in \mathcal{C}$ and $u \in \mathbb{R}$,

$$\begin{aligned}
&h^{(r)}(x,u) - \tilde{h}^{(r)}(x,u) \\
\geq& - |\tilde{\Delta}_r(x) - \Delta_r(x)| - |\tilde{\Sigma}_r(x) - \Sigma_r(x)|u \\
=& - |\tilde{\Delta}_r(x) - \Delta_r(x)| - |(\tilde{\Sigma}_r(x) - \Sigma_r(x))u| \\
\geq& - |\tilde{\Delta}_r(x) - \Delta_r(x)| - |\tilde{\Sigma}_r(x) - \Sigma_r(x)| \cdot |u| \\
\geq& - \varepsilon - \varepsilon|u|
\end{aligned}$$

and

$$K \cdot (\eta(x) - \tilde{\eta}(x)) = \sum_{i=1}^{r-1} k_i(\Delta_i(x) - \tilde{\Delta}_i(x))$$
$$\geq - \sum_{i=1}^{r-1} k_i \cdot \sup_{x\in\mathcal{C}} \|\varphi(x) - \tilde{\varphi}(x)\|$$
$$\geq - \varepsilon \sum_{i=1}^{r-1} k_i.$$

Then, for all $x \in \mathcal{C}$ and $u \in \mathbb{R}$, we have

$$h^{(r)}(x, u) + K \cdot \eta(x)$$
$$\geq \tilde{h}^{(r)}(x, u) - \varepsilon \max_{u\in\mathcal{U}} |u| + K \cdot \tilde{\eta}(x) - \varepsilon \left(1 + \sum_{i=1}^{r-1} k_i\right).$$

Take supremum over $u \in \mathbb{R}$ on both sides, the conclusion follows. □

For systems with unknown dynamics, let the hypothesis in Proposition 3.3.2 be held, then by imposing condition (3.20) for controller synthesis, we can achieve safety-critical control of set $\mathcal{C}$ as long as we choose the training set sufficiently dense. We can derive a similar condition as (3.20), where an arbitrarily small robustness should be introduced.

3.3.1 Feasibility of Assumption 3.3.1

Note that the compactness assumption on the safe set is reasonable as per real-world application. We then verify the feasibility of (3) of Assumption 3.3.1.

Differentiation Error of φ Over a Finite Set

The functions Δ_i for $i \in \{1, 2, \cdots, r\}$ and Σ_r are the difference between the time derivatives $h^{(i)}$ and $\hat{h}^{(i)}$ for $i \in \{1, 2, \cdots, r\}$. To illustrate the basic idea of the scheme, we do not attempt to use the real-world data as the training set. Instead, we use an artificial model as the hypothetically real model and generate training data using numerical differentiation of the numerical models based on $\mathfrak{C}$. The numerical error is such that, for any $\delta > 0$, we can find a sufficiently dense finite set $\mathfrak{C} \subset \mathcal{C}$ such that

$$\sup_{x \in \mathfrak{C}} \|\bar{\varphi}(x) - \varphi(x)\| \leq \delta, \tag{3.21}$$

where $\bar{\varphi}(x)$ represents the numerical differentiation of φ at some $x \in \mathfrak{C}$.

Optimization Error

For any $\eta > 0$, we assume there exists an optimizer that can learn an approximation $\tilde{\varphi}$ based on data $\{\bar{\varphi}(x) : x \in \mathfrak{C}\}$ such that

$$\sup_{x \in \mathfrak{C}} \|\tilde{\varphi}(x) - \bar{\varphi}(x)\| < \eta. \tag{3.22}$$

Generalization Error

By continuity of $\tilde{\varphi}$ and φ, there exists some $x^* \in \mathcal{C}$ such that

$$\sup_{x \in \mathcal{C}} \|\tilde{\varphi}(x) - \varphi(x)\| = \|\tilde{\varphi}(x^*) - \varphi(x^*)\|.$$

For any $\vartheta > 0$, by tuning $\mathfrak{C}$ to be sufficiently dense in $\mathcal{C}$ and the Lipschitz continuity of $\tilde{\varphi}(x)$ and $\varphi(x)$ on $\mathcal{C}$, there exists some $y \in \mathfrak{C}$ such that

$$\|\tilde{\varphi}(x^*) - \tilde{\varphi}(y)\| \leq \vartheta, \quad \|\varphi(x^*) - \varphi(y)\| \leq \vartheta.$$

Consequently,

$$\begin{aligned}
&\sup_{x \in \mathcal{C}} \|\tilde{\varphi}(x) - \varphi(x)\| = \|\tilde{\varphi}(x^*) - \varphi(x^*)\| \\
=&\|\tilde{\varphi}(y) - \varphi(y) + \tilde{\varphi}(x^*) - \tilde{\varphi}(y) + \varphi(y) - \varphi(x^*)\| \\
\leq&\|\tilde{\varphi}(y) - \varphi(y)\| + 2\vartheta \\
=&\|\tilde{\varphi}(y) - \bar{\varphi}(y) + \bar{\varphi}(y) - \varphi(y)\| + 2\vartheta \\
\leq&\sup_{y \in \mathfrak{C}} \|\tilde{\varphi}(y) - \bar{\varphi}(y)\| + \|\bar{\varphi}(y) - \varphi(y)\| + 2\vartheta \\
\leq&\eta + \sup_{y \in \mathfrak{C}} \|\bar{\varphi}(y) - \varphi(y)\| + 2\vartheta \\
\leq&\eta + \delta + 2\vartheta \leq \varepsilon.
\end{aligned}$$

Provided that we can choose η, δ, and ϑ sufficiently small, Assumption 3.3.1-(2) is verified to be feasible.

Remark 3.3.3. *The final $\mathfrak{C}$ should be chosen based on all of the above criteria from the above such that (2) of Assumption 3.3.1 can be satisfied. We provide a detailed data collection algorithm in Section 3.4.*

3.4 Learning Framework for Uncertainty

In this section, we will discuss the process of collecting training data and training for approximating $[h^{(1)}(x), h^2(x), \ldots, h^{(r-1)}(x), h^{(r)}(x, u)]^T$.

3.4.1 Collecting Training Data

In order to collect training data, we sample initial states and let the system evolve under control inputs. To be more specific, for any $T > 0$, we fix a sufficiently refined partition $\{t_0, t_1, \cdots, t_N\}$ of the interval $[0, T]$ such that $N \in \mathbb{N}$ and $0 = t_0 < t_1 < \cdots < t_N = T$. Introduce the index set $\mathcal{I}_N := \{0, 1, \cdots, N\}$. Let $\mathfrak{x}_i := x(t_i; \mathfrak{x}_0)$ for the given sampled initial condition $\mathfrak{x}_0$ and for each $i \in \mathcal{I}_N$. Then, for each $m < r$, the input layer data is given as $\{\mathfrak{x}_i\}_{i \in \mathcal{I}_N}$. On the other hand, the data $h^{(r)}(\mathfrak{x}_i)$ for $i \in \mathcal{I}_N$ can be obtained using numerical differentiation of the (hypothetically) real model. Hence, the output layer data is given as $\{\Delta_m(\mathfrak{x}_i)\}_{i \in \mathcal{I}_N}$, where $\Delta_m(\mathfrak{x}_i) = h^{(m)}(\mathfrak{x}_i) - \hat{h}^{(m)}(\mathfrak{x}_i)$ for $i \in \mathcal{I}_N$.

However, for $m = r$, we need two arbitrary valid control inputs $u_{i,1}$ and $u_{i,2}$ to obtain the training data for Δ_r and Σ_r, where $u_{i,1} \neq u_{i,2}$. For each $i \in \mathcal{I}_N$, we can generate $h^{(r)}(\mathfrak{x}_i, u_{i,1})$ and $h^{(r)}(\mathfrak{x}_i, u_{i,2})$ by numerical differentiation. Let $e := h^{(r)} - \hat{h}^{(r)}$. Then, by Eq (3.17), we have

$$\begin{aligned} e(\mathfrak{x}_i, u_{i,1}) &= \Delta_r(\mathfrak{x}_i) + \Sigma_r(\mathfrak{x}_i) u_{i,1}, \\ e(\mathfrak{x}_i, u_{i,2}) &= \Delta_r(\mathfrak{x}_i) + \Sigma_r(\mathfrak{x}_i) u_{i,2}. \end{aligned} \tag{3.23}$$

Conversely, for $i \in \mathcal{I}_N$, given $\mathfrak{x}_i$ and two arbitrarily selected control inputs such that $u_{i,1} \neq u_{i,2}$, we have

$$\begin{aligned} \Delta_r(\mathfrak{x}_i) &:= \frac{u_{i,2} e(\mathfrak{x}_i, u_{i,1}) - u_{i,1} e(\mathfrak{x}_i, u_{i,2})}{u_{i,2} - u_{i,1}}, \\ \Sigma_r(\mathfrak{x}_i) &:= \frac{e(\mathfrak{x}_i, u_{i,2}) - e(\mathfrak{x}_i, u_{i,1})}{u_{i,2} - u_{i,1}}. \end{aligned} \tag{3.24}$$

Note that the above relation is by directly reversing (3.23) regardless of the choice of $(u_{i,1}, u_{i,2})$. The cancellation of the dependence on $(u_{i,1}, u_{i,2})$ for (Δ_r, Σ_r) is due to the special control affine form.

We store the information into training buffers $\mathcal{B} = \{(\mathfrak{x}_i, \mathfrak{y}_i)\}_{i \in \mathcal{I}_N}$, where

$$\mathfrak{y}_i = [\Delta_1(\mathfrak{x}_i), \Delta_2(\mathfrak{x}_i), \cdots, \Delta_r(\mathfrak{x}_i), \Sigma_r(\mathfrak{x}_i)]^{\mathrm{T}}, \quad i \in \mathcal{I}_N.$$

3.4.2 Learning Framework

After we collect the training data set, we can use a deep neural network for approximation. Define the network as $\tilde{\Delta}(\mathfrak{x}|w)$ with w be the weights of the network and denote $\tilde{\Delta}_i(\mathfrak{x}|w)$ as the i-th entry of the output of the network. By defining the loss function using mean square error (MSE) as

$$\mathcal{L}(w) = \frac{1}{N} \sum_{i=0}^{N} \big(y_i - \tilde{\Delta}_i(\mathfrak{x}|w)\big),$$

we have

$$\begin{aligned}
\frac{\partial \mathcal{L}(w)}{\partial w} &= \frac{\partial \mathcal{L}(w)}{\partial \tilde{\Delta}_i(\mathfrak{x}_i|w)} \cdot \frac{\partial \tilde{\Delta}(\mathfrak{x}_i|w)}{\partial w} \\
&= \frac{2}{N} \sum_{i=0}^{N} (\partial \tilde{\Delta}_i(\mathfrak{x}_i|w) - \mathfrak{y}_i) \cdot \frac{\hat{\Delta}_i(\mathfrak{x}_i|w)}{\partial w} \\
&= \frac{2}{N} \sum_{i=0}^{N} (\partial \tilde{\Delta}_i(\mathfrak{x}_i|w) - \mathfrak{y}_i) \cdot \mathfrak{x}_i
\end{aligned}$$

According to gradient decent method, we can update weights of the deep neural network as

$$w_{k+1} = w_k + \frac{2\alpha}{N} \sum_{i=0}^{N} (\partial \hat{\Delta}_i(\mathfrak{x}_i|w) - \mathfrak{y}_i) \cdot \mathfrak{x}_i$$

with α the learning rate. By writing $\bar{w}$ as the weight when the training process converges, we can the mth order derivative for each $m \in \{1, 2, \ldots, r\}$ as

$$\begin{bmatrix} h^{(1)}(x) \\ h^{(2)}(x) \\ \vdots \\ h^{(r-1)}(x) \\ h^{(r)}(x,u) \end{bmatrix} = \begin{bmatrix} \hat{h}^{(1)}(x) + \tilde{\Delta}_1(x|\bar{w}) \\ \hat{h}^{(2)}(x) + \tilde{\Delta}_2(x|\bar{w}) \\ \vdots \\ \hat{h}^{(r-1)}(x) + \tilde{\Delta}_{r-1}(x|\bar{w}) \\ \hat{h}^{(r)}(x,u) + \tilde{\Delta}_r(x|\bar{w}) + \tilde{\Delta}_{r+1}(x|\bar{w})u \end{bmatrix}. \tag{3.25}$$

$$\dot{\hat{E}}(x) = \dot{\hat{h}}^r(x,u) + \tilde{\Delta}_r(x|\bar{w}) + \tilde{\Delta}_{r+1}(x|\bar{w})u.$$

After we have identified the $\tilde{\Delta}(\mathfrak{x}|w)$, we can calculate $b_r(x)$ iteratively according to Eq (3.25). Then we can obtain safe control inputs by solving the quadratic programming problem

$$\begin{aligned} u(x) = \arg\min_{u\in\mathbb{R}} \frac{1}{2}||u - k(x)||^2, \\ \text{s.t.} \quad b_r(x,u) \geq 0. \end{aligned} \tag{3.26}$$

This quadratic programming problem helps to find a safe control that is nearest to the nominal control $k(x)$. The pseudo code for data collection is shown in Algorithm 1. After we collect the training data, a neural network is trained using the MSE loss function.

Algorithm 1 Algorithm for data collection

Require: A working space, an initial state $\mathfrak{x}$, a nominal CBF $\hat{h}$, buffer $\mathcal{B}$, a partition $\{t_0, t_1, \cdots, t_N\}$ of the time interval $[0, T]$.

for i in N **do**
- Sample two control inputs $u_{i,1}$ and $u_{i,2}$
- Get $\mathfrak{x}_{i+1,1}$ and $\mathfrak{x}_{i+1,2}$ from $\mathfrak{x}_i$ using $u_{i,1}$ and $u_{i,2}$
- Calculate $h(\mathfrak{x}_i)$, $h(\mathfrak{x}_{i+1,1})$ and $h(\mathfrak{x}_{i+1,2})$
- Estimate $h^{(1)}(\mathfrak{x}_i), h^{(2)}(\mathfrak{x}_i), \ldots, h^{(r)}(\mathfrak{x}_i, u_{i,1})$ and $h^{(r)}(\mathfrak{x}_i, u_{i,2})$ using numerical differentiation.
- Calculate $\hat{h}^{(1)}(\mathfrak{x}_i), \hat{h}^{(2)}(\mathfrak{x}_i), \ldots, \hat{h}^{(r)}(\mathfrak{x}_i, u_{i,1})$ and $\hat{h}^{(r)}(\mathfrak{x}_i, u_{i,2})$
- Calculate $\Delta_1(\mathfrak{x}_i), \Delta_2(\mathfrak{x}_i), ..., \Delta_{r-1}(\mathfrak{x}_i)$ according to Eq (3.18)
- Calculate $\Delta_r(\mathfrak{x}_i)$ and $\Sigma_r(\mathfrak{x}_i)$ according to Eq (3.23) and Eq (3.24)
- $\mathcal{B} \leftarrow \{\mathfrak{x}_i, [\Delta_1(\mathfrak{x}_i), \Delta_2(\mathfrak{x}_i), ..., \Delta_r(\mathfrak{x}_i), \Sigma_r(\mathfrak{x}_i)]^T\}$

end for

3.5 Simulation Result

3.5.1 Unknown Dubins Vehicle

In this section, we test our algorithm using a differential drive model as in [84] with

$$\begin{aligned}\dot{x} &= v\cos\vartheta,\\ \dot{y} &= v\sin\vartheta,\\ \dot{\vartheta} &= \omega,\end{aligned} \tag{3.27}$$

with $s = [x, y, \vartheta]^T$ the state where x and y are the planar positions of the center of the vehicle, ϑ is its orientation, v is the forward velocity and ω is the control input of the system. The uncertainty of the systems comes from the forward velocity of v.

Static Obstacle Avoidance

In the first case, the Dubins vehicle is required to travel within the circle $x^2 + y^2 < 1$. The system is of relative degree 2 w.r.t h so we select

$$h(s) = 1 - x^2 - y^2$$

as our HOCBF and we have

$$\begin{aligned}b_0 &= h(s) = 1 - x^2 - y^2,\\ b_1 &= \dot{b}_0 + p_1 b_0,\\ b_2 &= \dot{b}_1 + p_2 b_1.\end{aligned}$$

The Lie derivatives of h is calculated as

$$\begin{aligned}L_f h(s) &= -2v(x\cos\vartheta + y\sin\vartheta),\\ L_f^2 h(s) + L_f L_g h(s)u &= -2v^2 - 2v(y\cos\vartheta - x\sin\vartheta)w.\end{aligned}$$

The uncertainty of the model comes from inaccurate estimation of the forward velocity v such that the real velocity is $v_{\text{real}} = 0.8$ and the nominal forward velocity is $v_{\text{norm}} = 0.5$.

In each time step, we estimate the first and second order derivative of h numerically using Newton's method as

$$\begin{aligned}\dot{h}_t(s_t) &= \frac{h_t(s_t) - h_{t-1}(s_{t-1})}{\Delta t}, \\ \ddot{h}_t(s_t) &= \frac{h_{t+1}(s_{t+1}) + h_{t-1}(s_{t-1}) - 2h_t(s_t)}{\Delta t^2}.\end{aligned} \tag{3.28}$$

Then the training data is calculated using Eq (3.18). A two-layer neural network is used to learn the derivatives of h. The network has 3 inputs as the dimension of the state is three. The network has 3 outputs representing $\Delta_1(s), \Delta_2(s)$ and $\Sigma_2(s)$ as in Eq (3.18). We train the network with 500 epochs and the simulated trajectories of the system are shown as in Fig (3.1). The total simulation time is 50s, which is equally divided into 1000 steps with $\Delta t = 0.05s$. We test trajectories with $p_1 = 0.5, p_2 = 0.5$, $p_1 = 1, p_2 = 1$ and $p_1 = 5, p_2 = 5$. All the trajectories start from initial state $[0.9, 0.1, 0]$, which is marked as a red star. We can see that all the trajectories are safe under the learned derivatives of the HOCBF.

Moving Obstacle Avoidance

In the second case, we test our algorithm for moving obstacle avoidance. As is shown in Figure (3.2), the initial position of the vehicle is marked as the purple star. A moving obstacle moves along the x-axis to the right from $(-1, 0)$ with a speed of $0.5/s$. The radius of the obstacle is $r_O = 0.5$ and the goal is marked as the blue circle. We use

$$h(s) = (x - x_{O_t})^2 + (y - y_{O_t})^2 - r_O^2,$$

where x_{O_t} and y_{O_t} are x and y coordinate of the obstacle at time t. The first and second order derivative of $h(s)$ is calculated as

$$\begin{aligned}\dot{h}(s) &= (2(x - x_{O_t})\cos\vartheta + 2(y - y_{O_t})\sin\vartheta)v, \\ \ddot{h}(s) &= 2v^2 + 2vw((y - y_{O_t})\cos\vartheta - (x - x_{O_t})\sin\vartheta).\end{aligned}$$

The nominal speed and real speed of the vehicle is the same as Case 1. We use the same network structure and training scale to learn the derivatives of the HOCBF. Once the derivatives are learned, we use a PID controller as the nominal control to reach the target and use the HOCBF for safety guarantee. The simulation results are shown in Fig (3.2). The purple circle and star are the positions of the obstacle and the vehicle at initial position at time step $t = 0s$. The black circle and star are the positions of obstacle and vehicle at time $t = 4$ and the green circle and star are those for $t = 7$. We also plot the

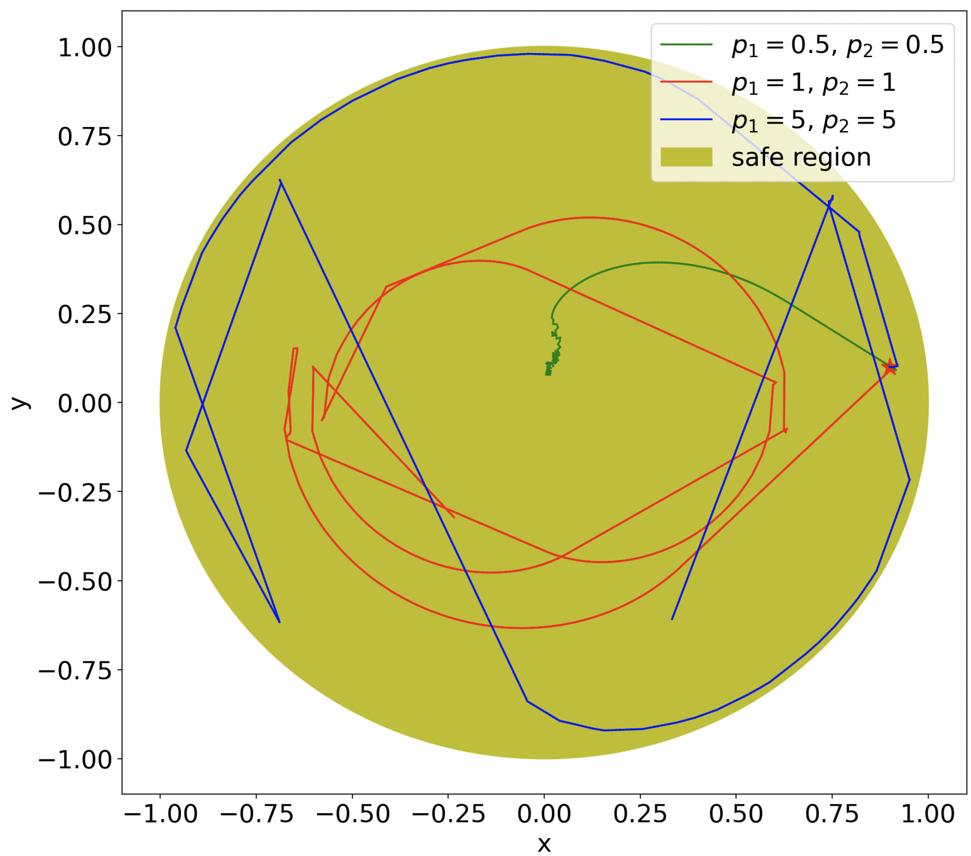
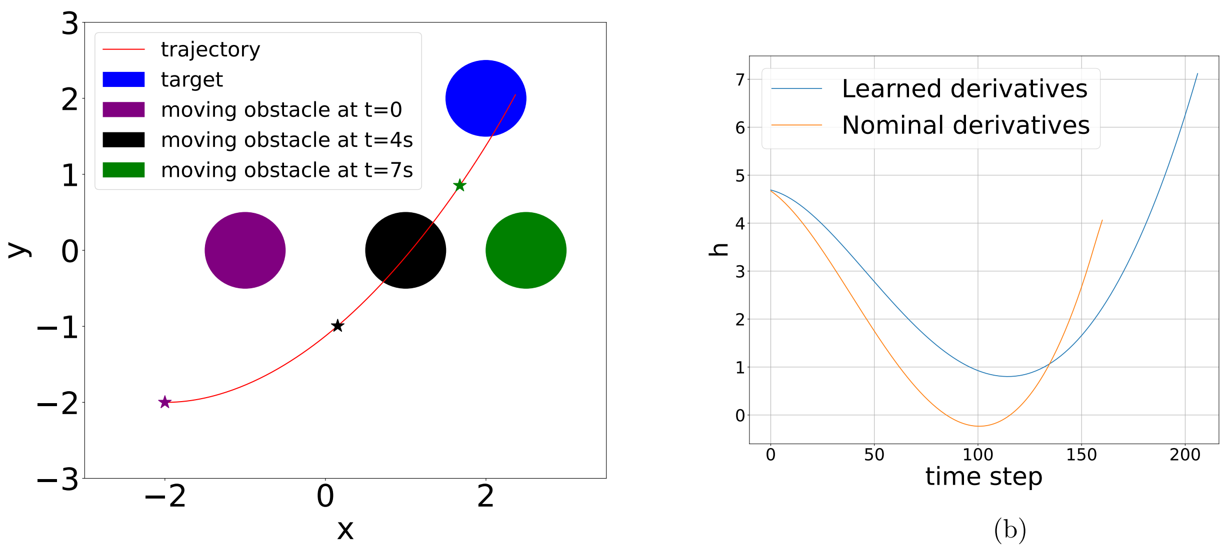


Figure 3.1. Simulation results of the Dubins vehicle using learned derivatives of HOCBF. The safe region is the yellow circle. All the trajectories start from the point marked as star at $(0.9, 0.1)$.

values of h during controlled process by using the nominal and learned derivatives of the HOCBF, respectively, as in Fig (3.2b). We see that the value of $h(s)$ is always positive using the learned derivatives while the value drops below 0 for using the nominal derivatives of HOCBF. This implies that the vehicle avoids the moving obstacle successfully when we use the learned HOCBF while it collides with the obstacle when we use the nominal derivatives. The reason that using the learned derivatives has more time steps as in Fig (3.2b) is that the vehicle slows down when approaching the obstacle so it takes longer to achieve the target.



(a)

(b)

Figure 3.2. Simulation results for avoiding moving obstacle: the blue circle is the goal region and the red curve is the trajectory of the vehicle using the learned derivatives of the HOCBF. (a): The initial position is at $(-2, -2, 0)$ marked as the purple star. The obstacle is marked with the purple circle initially at $(-1, 0)$ and moves right with a speed of $0.5/s$. The position of the obstacle at time $t = 4s$ and $t = 7s$ are marked as black and green circle. The corresponding position of the vehicle as $t = 4s$ and $t = 7s$ are spot as the black and green star. (b): The value of h during simulation. The blue curve is the h value using the learned derivatives of the HOCBF and the orange curve is the h value of using the nominal derivatives of the HOCBF.

3.5.2 Altitude Safety Control of Quadrotor

In the second experiment, we test our method for altitude control of a quadrotor with safety constraints. The full dynamics of a quadrotor is a 12-dimensional system with the state $\boldsymbol{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \varphi, \vartheta, \psi, \dot{\varphi}, \dot{\vartheta}, \dot{\psi}]$ where x, y, z are the positions along x-axis, y-axis and z-axis and φ, ϑ, ψ are the roll, pitch and yaw angles. According to [14], a simplified state-space model of a quadrotor can be written as

$$\begin{cases} \ddot{x} = (\cos\varphi \sin\vartheta \cos\psi + \sin\varphi \sin\psi)\frac{U_1}{m}, \\ \ddot{y} = (\cos\varphi \sin\vartheta \sin\psi - \sin\varphi \cos\psi)\frac{U_1}{m}, \\ \ddot{z} = -g + (\cos\varphi \cos\vartheta)\frac{U_1}{m}, \\ \ddot{\psi} = \dot{\varphi}\dot{\vartheta}(\frac{I_x - I_y}{I_z}) + \frac{U_4}{Iz}, \\ \ddot{\vartheta} = \dot{\varphi}\dot{\psi}(\frac{I_z - I_x}{I_y}) + \frac{l}{I_y}U_3, \\ \ddot{\varphi} = \dot{\vartheta}\dot{\psi}(\frac{I_y - I_z}{I_x}) + \frac{l}{I_x}U_2, \end{cases} \tag{3.29}$$

and

$$\begin{cases} U_1 = k_t(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2), \\ U_2 = k_t(\Omega_4^2 - \Omega_2^2), \\ U_3 = k_t(\Omega_1^2 - \Omega_3^2), \\ U_4 = k_d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2), \end{cases} \tag{3.30}$$

where m is the mass of the quadrotor, Ω_i, $i = 1, 2, 3, 4$ denote rotation speed of the four rotors, k_t and k_d are the drag force coefficient and reverse moment coefficient, respectively, l is the distance from the center of the quadrotor to the center of the rotors. The altitude along z-axis is controlled only by thrust U_1 and as a result, a subsystem

$$\begin{cases} \dot{z} = v_z, \\ \ddot{z} = -g + (\cos\varphi \cos\vartheta)\frac{U_1}{m} \end{cases} \tag{3.31}$$

can be decoupled for the altitude control. The function

$$h(z) = 1 - (\frac{z - c}{p})^4 \geq 0 \tag{3.32}$$

is selected to be our HOCBF and we have

$$\begin{aligned} b_0 &= h(z), \\ b_1 &= \dot{b}_0 + p_1 b_0, \\ b_2 &= \dot{b}_1 + p_2 b_1. \end{aligned}$$

The Lie derivatives of $h(z)$ are calculated as

$$\begin{aligned} L_f h(z) &= \frac{-4(z-c)^3}{p^4}\dot{z}, \\ L_f^2 h(z) &= \frac{-4(z-c)^3}{p^4}g - \frac{12(z-c)^2}{p^4}\dot{z}, \\ L_g L_f h(z) &= \frac{4(z-c)^3}{mp^4}\cos\varphi\cos\vartheta. \end{aligned} \tag{3.33}$$

The uncertainty of the model comes from inaccuracy of the mass of the quadrotor. In our experiment, the nominal model has $m_n = 0.4$ where the real mass of the quadrotor is $m = 0.6$. We select $p = 2$ to construct the safe region. In order to learn the derivatives of the HOCBF, we need two non-zero control inputs at each time step to calculate $\Delta_2(z)$ and $\Sigma_2(z)$ as in Eq (3.23). We set two goals for $z_{ref_{t,1}}$ and $z_{ref_{t,2}}$ and use a PID controller as in [73] to obtain such controls $U_{t,1}$ and $U_{t,2}$. Then we have

$$\begin{aligned} \ddot{h}_{t,1}(z_t, U_{t,1}) &= \ddot{\hat{h}}(z_t, U_{t,1}) + \Delta_2(z_t) + \Sigma_2(z_t)U_{t,1}, \\ \ddot{h}_{t,2}(z_t, U_{t,2}) &= \ddot{\hat{h}}(z_t, U_{t,2}) + \Delta_2(z_t) + \Sigma_2(z_t)U_{t,2}, \end{aligned} \tag{3.34}$$

according to Eq (3.17) and

$$\begin{aligned} \Delta_2(z_t) &= \frac{E(z_t, U_{t,1})U_{t,2} - E(z_t, U_{t,2})U_{t,1}}{U_{t,2} - U_{t,1}}, \\ \Sigma_2(z_t) &= \frac{E(z_t, U_{t,2}) - E(z_t, U_{t,1})}{U_{t,2} - U_{t,1}} \end{aligned} \tag{3.35}$$

according to Eq (3.24), where

$$\begin{aligned} E(z_t, U_{t,1}) &= \ddot{h}(z_t, U_{t,1}) - \ddot{\hat{h}}(z_t, E(z_t, U_{t,1}), \\ E(z_t, U_{t,2}) &= \ddot{h}(z_t, U_{t,2}) - \ddot{\hat{h}}(z_t, E(z_t, U_{t,2}). \end{aligned} \tag{3.36}$$

Again, $\ddot{\hat{h}}(z_t, U_{t,1})$ and $\ddot{\hat{h}}(z_t, U_{t,2})$ are estimated numerically and $\ddot{h}(z_t, U_{t,1})$ and $\ddot{h}(z_t, U_{t,2})$ are calculated analytically given z_t, $U_{t,1}$ and $U_{t,2}$ using Eq (3.33). We use a two-layer neural network with 100 and 30 nodes in each layer and train the network with 500 epoch as well. We test the performance of the learning using a sinusoidal reference trajectory as $z = 2.5\sin t + 5$. We select $p_1 = 2, p_2 = 8$ as in our simulation and the result is shown as in Fig (3.3). As we can see in the result, using the nominal derivatives of the HOCBF, the system will violate safety guarantee while using the learned derivatives, the system is safe in z-direction.

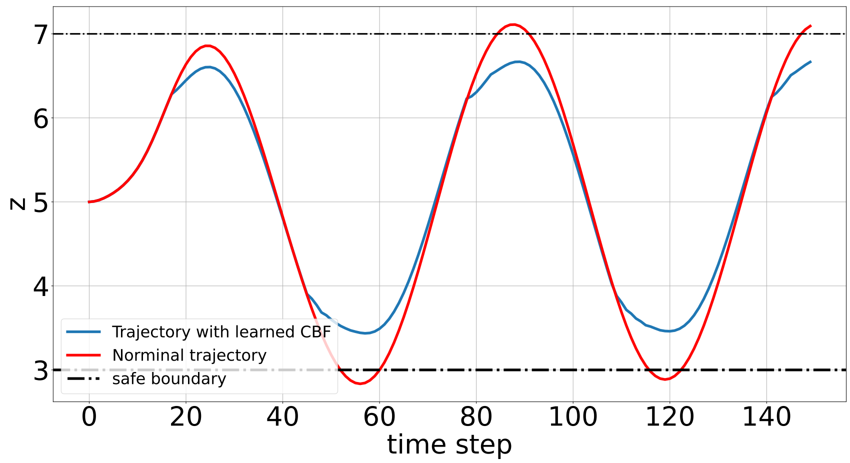


Figure 3.3. Simulation result for altitude control of quadrotor under reference trajectory $z = 2.5\sin t + 5$. The initial height of the quadrotor is $z = 5$. The red curve is the trajectory with nominal HOCBF while the blue curve is the trajectory using the learned HOCBF. The horizontal dashed line is the safe boundary for z.

3.6 Conclusion

In this Chapter, we discuss safety-critical control for unknown deterministic systems. We present a framework for learning the derivatives of HOCBF for system with uncertainty. We first provide sufficient conditions on controllers via HOCBF for set invariance. Then we show that the real derivatives of the HOCBF can be learned from that of the nominal derivatives and remainders by using neural networks and the resulting control frame work is also control affine. We also show that under some reasonable assumptions, the learned derivatives of the HOCBF will provide a robust safety guarantee for the systems. We show in simulation that our method can handle model uncertainty using a Dubins vehicle model and a quadrotor model.

Chapter 4

Safety-Critical Control with Gaussian Noise

Imperfect models can sometimes lead to severe consequences in system control. Therefore, synthesizing controllers addressing uncertainty is crucial. In this Chapter, we consider stochastic systems with completely known drift and diffusion. We begin by analyzing the pros and cons of various types of stochastic certificates and then propose of our notion of Stochastic Control Barrier Functions (SCBFs).

4.1 Preliminary and Problem Definition

4.1.1 System Description

Given a filtered probability space $(\Omega, \mathscr{F}, \{\mathscr{F}_t\}, \mathbb{P})$ with a natural filtration, a state space $\mathcal{X} \subseteq \mathbb{R}^n$, a (compact) set of control values $\mathcal{U} \subset \mathbb{R}^p$, consider a continuous-time stochastic process $X : [0, \infty) \times \Omega \to \mathcal{X}$ that solves the SDE

$$dX_t = (f(X_t) + g(X_t)u(t))dt + \sigma(X_t)dW_t, \tag{4.1}$$

where $u : \mathbb{R}_{\geq 0} \to \mathcal{U}$ is a bounded measurable control signal; W is a d-dimensional standard $\{\mathscr{F}\}_t$-Brownian motion; $f : \mathcal{X} \to \mathbb{R}^n$ is a nonlinear vector field; $g : \mathcal{X} \to \mathbb{R}^{n\times p}$ and $\sigma : \mathcal{X} \to \mathbb{R}^{n\times d}$ are smooth mappings.

Assumption 4.1.1. *We make the following assumptions on system (4.1) for the rest of this chapter:*

(i) There is a $\xi \in \mathcal{X}$ such that $\mathbb{P}[X_0 = \xi] = 1$;

(ii) The mappings f, g, σ satisfy local Lipschitz continuity and a linear growth condition.

Definition 4.1.2 (Strong solutions)**.** *A stochastic process X is said to be a strong solution to (4.1) if it satisfies the following integral equation*

$$X_t = \xi + \int_0^t (f(X_s) + g(X_s)u(s))ds + \int_0^t \sigma(X_s)dW_s, \tag{4.2}$$

where the stochastic integral is constructed based on the given Brownian motion W.

Remark 4.1.3. *(i) Under Assumption 4.1.1, the SDE (4.1) admits a unique strong solution.*

(ii) Weak solutions are in the sense that Brownian motions are constructed posterior for the stochastic integrals.

We exclude the consideration of weak solutions in this chapter to guarantee that Lyapunov-type analysis on sample paths behaviors is based on the same Brownian motion.

Definition 4.1.4 (Infinitesimal generator of X_t)**.** *Let X be the strong solution to (4.1), the infinitesimal generator $\mathcal{A}$ of X_t is defined by*

$$\mathcal{A}h(x) = \lim_{t \downarrow 0} \frac{\mathbb{E}^x[h(X_t)] - h(x)}{t}; \quad x \in \mathbb{R}^n, \tag{4.3}$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is in a set $\mathcal{D}(\mathcal{A})$ (called the domain of the operator $\mathcal{A}$) of functions such that the limit exists at x.

Proposition 4.1.5 (Dynkin)**.** *Let X solve (4.1). If $h \in C_0^2(\mathbb{R}^n)$ then $h \in \mathcal{D}(\mathcal{A})$ and*

$$\mathcal{A}h(x) = \frac{\partial h}{\partial x}(f(x) + g(x)u(t)) + \frac{1}{2}\sum_{i,j} \left(\sigma\sigma^T\right)_{i,j}(x)\frac{\partial^2 h}{\partial x_i \partial x_j}. \tag{4.4}$$

Remark 4.1.6. *The solution X to (4.1) is right continuous and satisfies strong Markov properties, and for any finite stopping time τ and $h \in C_0^2(\mathbb{R}^n)$, we have the following Dynkin's formula*

$$\mathbb{E}^\xi[h(X_\tau)] = h(\xi) + \mathbb{E}^\xi\left[\int_0^\tau \mathcal{A}h(X_s)ds\right],$$

and therefore

$$h(X_\tau) = h(\xi) + \int_0^t \mathcal{A}h(X_s)ds + \int_0^\tau \frac{\partial h(X_s)}{\partial x}dW_s.$$

The above is an analogue of the evolution of h along trajectories

$$h(x(t)) = h(\xi) + \int_0^t [L_f h(x(s)) + L_g h(x(s))u(s)]ds$$

driven by deterministic dynamics $\dot{x} = f(x) + g(x)u$, where $L_f h(x) = \frac{\partial h(x)}{\partial x} \cdot f(x)$ and $L_g h(x) = \frac{\partial h(x)}{\partial x} \cdot g(x)$.

4.1.2 Set Invariance and Control

In deterministic settings, a set $\mathcal{C} \subseteq \mathcal{X}$ is said to be invariant for a dynamical system $\dot{x} = f(x)$ if, for all $x(0) \in \mathcal{C}$, the solution $x(t)$ is well defined and $x(t) \in \mathcal{C}$ for all $t \geq 0$. As for stochastic analogies, we have the following probabilistic characterization of set invariance.

Definition 4.1.7 (Probabilistic set invariance). *Let X be a stochastic process. A set $\mathcal{C} \subset \mathcal{X}$ is said to be invariant w.r.t. a tuple (x, T, p) for X, where $x \in \mathcal{C}$, $T \geq 0$, and $p \in [0, 1]$, if $X_0 = x$ a.s. implies*

$$\mathbb{P}_x[X_t \in \mathcal{C},\ 0 \leq t \leq T] \geq p. \tag{4.5}$$

Moreover, if $\mathcal{C} \subset \mathcal{X}$ is invariant w.r.t. $(x, T, 1)$ for all $x \in \mathcal{C}$ and $T \geq 0$, then $\mathcal{C}$ is strongly invariant for X.

For stochastic dynamical systems with controls such as system (4.1), we would like to define similar probabilistic set invariance property for the controlled processes. Before that, we first define the following concepts.

Definition 4.1.8 (Control strategy). *A control strategy is a set-valued function*

$$\kappa : \mathcal{X} \to 2^{\mathcal{U}}. \tag{4.6}$$

We use a boldface $\mathbf{u}$ to indicate a set of constrained control signals. A special set of such signals is given by a control strategy as defined below.

Definition 4.1.9 (State-dependent control). *We say that a control signal u conforms to a control strategy κ for (4.1), and writes $u \in \mathbf{u}_\kappa$, if*

$$u(t) \in \kappa(X_t), \quad \forall t \geq 0, \tag{4.7}$$

where X satisfies (4.1) with u as input. The set of all control signals that confirm to κ is denoted by $\mathbf{u}_\kappa$.

Definition 4.1.10 (Controlled probabilistic invariance)**.** *Given system* (4.1) *and a set of control signals* $\mathbf{u}$*, a set* $\mathcal{C} \subset \mathcal{X}$ *is said to be controlled invariant under* $\mathbf{u}$ *w.r.t. a tuple* (x, T, p) *for system* (4.1)*, if there exists a control input* $u \in \mathbf{u}$*,* $\mathcal{C}$ *is invariant w.r.t.* (x, T, p) *for* X*, where* X *is the solution to* (4.1) *with* u *as input.*

Similarly, $\mathcal{C} \subset \mathcal{X}$ *is strongly controlled invariant under* $\mathbf{u}$ *if* $\mathcal{C} \subset \mathcal{X}$ *is controlled invariant under* $\mathbf{u}$ *w.r.t.* $(x, T, 1)$ *for all* $x \in \mathcal{C}$ *and* $T \geq 0$*.*

4.1.3 Problem Definition

For the rest of this chapter, we consider a safe set of the form

$$\mathcal{C} := \{x \in \mathcal{X} : h(x) \geq 0\}, \tag{4.8}$$

where $h : \mathcal{X} \to \mathbb{R}$ is a higher-order continuously differentiable function. We also define the boundary and interior of $\mathcal{C}$ explicitly as below

$$\partial\mathcal{C} := \{x \in \mathcal{X} : h(x) = 0\}, \tag{4.9}$$

$$\mathcal{C}^\circ := \{x \in \mathcal{X} : h(x) > 0\}. \tag{4.10}$$

Problem 4.1.11 (Probabilistic set invariance control)**.** *Given a compact set* $\mathcal{C} \subset \mathcal{X}$ *defined in* (4.8)*, a point* $\xi \in \mathcal{C}^\circ$*, and a tuple* (ξ, T, p)*, design a control strategy* κ *such that under* $\mathbf{u}_\kappa$*, the interior* $\mathcal{C}^\circ$ *is controlled invariant w.r.t.* (ξ, T, p) *for the resulting strong solutions to* (4.1)*.*

4.2 Safety-Critical Control Design via Barrier Function

In this section, we propose stochastic barrier certificates that can be used to design a control strategy κ for Problem 4.1.11. Before proceeding, it is necessary to review (stochastic) control barrier functions to interpret (probabilistic) set invariance. Note that we consider the safe set as constructed in (4.8), where the function h is given a priori.

4.2.1 Stochastic Reciprocal and Zeroing Barrier Function

Similar to the terminology for deterministic cases [8], we introduce the construction of stochastic control barrier functions as follows.

Definition 4.2.1 (SRCBF)**.** *A function $B : \mathcal{C}^\circ \to \mathbb{R}$ is called a stochastic reciprocal control barrier function (SRCBF) for system* (4.1) *if $B \in \mathcal{D}(\mathcal{A})$ and satisfies the following properties:*

(i) there exist class-$\mathcal{K}$ functions α_1, α_2 such that for all $x \in \mathcal{X}$ we have

$$\frac{1}{\alpha_1(h(x))} \le B(x) \le \frac{1}{\alpha_2(h(x))}; \tag{4.11}$$

(ii) there exists a class-$\mathcal{K}$ function α_3 such that

$$\inf_{u \in \mathcal{U}} [\mathcal{A}B(x) - \alpha_3(h(x))] \le 0. \tag{4.12}$$

We refer to the control strategy generated by (4.12) *as*

$$\varrho(x) := \{u \in \mathcal{U} : \mathcal{A}B(x) - \alpha_3(h(x)) \le 0\} \tag{4.13}$$

and the corresponding control constraint as $\mathbf{u}_\varrho$ (see in Definition 4.1.9).

Proposition 4.2.2 ([33])**.** *Suppose that there exists an SRCBF for system* (4.1)*. If $u(t) \in \mathbf{u}_\upsilon$, then for all $t \ge 0$ and $X_0 = \xi \in \mathcal{C}^\circ$, we have $\mathbb{P}_\xi[X_t \in \mathcal{C}^\circ] = 1$ for all $t \ge 0$.*

Remark 4.2.3. *The result admits a $\mathbb{P}$-a.s. controlled invariant set for the marginals of X, and is easily extended to a pathwise $\mathbb{P}$-a.s. controlled set invariance. Note that the strong solution is right continuous. Let $\{t_n,\ n = 1, 2, ...\}$ be the set of all rational numbers in $[0, \infty)$, and put*

$$\Omega^* := \bigcap_{1 \le n < \infty} \{\omega : X_{t_n} \in \mathcal{C}^\circ\},$$

then $\Omega^* \in \mathscr{F}$ (a σ-algebra is closed w.r.t. countable intersections). Since $\mathbb{Q}$ is dense in $\mathbb{R}$, X is right continuous, and h is continuous, we have

$$\Omega^* := \{\omega : X_t \in \mathcal{C}^\circ,\ \forall t \in [0, \infty)\}.$$

Note that $\mathbb{P}_\xi[\Omega^*] \equiv 1$ from the marginal result.

Definition 4.2.4 (SZCBF)**.** *A function $B : \mathcal{C} \to \mathbb{R}$ is called a stochastic zeroing control barrier function (SZCBF) for system* (4.1) *if $B \in \mathcal{D}(\mathcal{A})$ and*

(i) $B(x) \geq 0$ *for all* $x \in \mathcal{C}$*;*

(ii) $B(x) < 0$ *for all* $x \notin \mathcal{C}$*;*

(iii) there exists an extended $\mathcal{K}_\infty$ function α such that

$$\sup_{u\in\mathcal{U}} [\mathcal{A}B(x) + \alpha(B(x))] \geq 0. \tag{4.14}$$

We refer the control strategy generated by (4.14) *as*

$$\varkappa(x) := \{u \in \mathcal{U} : \mathcal{A}B(x) + \alpha(B(x)) \geq 0\} \tag{4.15}$$

and the corresponding set of constrained control signals as $\mathbf{u}_\varkappa$*.*

Proposition 4.2.5 (Worst-case probabilistic quantification)**.** *Suppose the mapping h is an SZCBF with linear function kx as the class-$\mathcal{K}$ function (where $k > 0$), and the control strategy as $\varkappa(x) = \{u \in \mathcal{U} : \mathcal{A}h(x) + kh(x) \geq 0\}$. Let $c = \sup_{x\in\mathcal{C}} h(x)$ and $X_0 = \xi \in \mathcal{C}^\circ$, then under any $u \in \mathbf{u}_\varkappa$ we have the following worst-case probability estimation:*

$$\mathbb{P}_\xi [X_t \in \mathcal{C}^\circ,\ 0 \leq t \leq T] \geq \left(\frac{h(\xi)}{c}\right) e^{-cT}. \tag{4.16}$$

Proof. Let $s = c - h(\xi)$ and $V(x) = c - h(x)$, then $V(x) \in [0, c]$ for all $x \in \mathcal{C}$. It is clear that $\mathcal{A}V(x) = -\mathcal{A}h(x)$. For $u(t) \in \mathbf{u}_\varkappa$ for all $t \in [0, T]$, we have

$$\mathcal{A}V(x) \leq -kV(x) + kc.$$

By [80, Theorem 3.1],

$$\mathbb{P}_\xi \left[\sup_{t\in[0,T]} V(X_t) \geq c\right] \leq 1 - \left(1 - \frac{s}{c}\right) e^{-cT}. \tag{4.17}$$

The result follows directly after this. □

In proposing stochastic control barrier functions for higher-order control systems, the above SRCBF and SZCBF are building blocks. The authors in [119] constructed higher-order SRCBF and have found the sufficient conditions to guarantee pathwise set invariance with probability 1. While the results seem strong, they come with significant costs. At the safety boundary, the control inputs need to be unbounded (as shown in the motivating example below and in the numerical experiments). On the other hand, the synthesis of controller for a higher-order system via SZCBF is with mild constraints. The trade-off is that the probability estimation of set invariance is of low quality (note that the worst-case probability estimation using first-order barrier function is already lower bounded by a small value over a relatively long time period). We propose higher-order stochastic control barrier functions in subsection C in order to reduce the high control efforts and improve the worst-case quantification. Before proceeding, we illustrate the above motivation through.

4.2.2 A Motivating Example

In this section, we will discuss how reciprocal control barrier functions (RCBFs) and SRCBFs perform differently around the boundary of the safe set. We show that for deterministic systems, RCBFs can guarantee safety with bounded control while for stochastic systems, SRCBFs require unbounded control in order to keep systems safe. We use the following two simple one-dimensional systems:

$$\dot{x} = x + u,$$

and

$$dx = (x + u)dt + \sigma dW,$$

for the comparison. Suppose that our safe set is $\{x \in \mathbb{R} \big| x < 1\}$ so that we can use $h(x) = 1 - x$. Accordingly, a RCBF for the deterministic system is $B = \frac{1}{h}$. We choose $\gamma = 1$ as in [8] and [33] and, as a result, the condition using the RCBF is

$$\begin{aligned} L_f B(x) + L_g B(x)u = \frac{1}{h^2}(x + u) \leq h, \\ u \leq (1 - x)^3 - x. \end{aligned}$$

It means that we can control the system safely using a control bounded by $(1 - x)^3 - x$ when $x \to 1$. However, for the stochastic system, we have $\frac{\partial^2 B}{\partial x^2} = \frac{2}{h^3}$. Then the SRCBF condition is

$$\begin{aligned} \mathcal{A}B = \frac{1}{h^2}(x + u) + \frac{\sigma^2}{h^3} \leq h, \\ u \leq h^3 - x - \frac{\sigma^2}{h}. \end{aligned}$$

As x approaches 1, the control approaches $-\infty$. This implies that in order to guarantee safety, we requires an unbounded control around the boundary of the safe set for stochastic systems, which can be difficult to satisfy for some practical applications.

4.2.3 Higher-Order Stochastic Control Barrier Function

To obtain non-vanishing worst-case probability estimation (compared to SZCBF), we propose a safety certificate via a stochastic Lyapunov-like control barrier function [80].

Definition 4.2.6 (Stochastic control barrier functions)**.** *A continuously differentiable function $B : \mathbb{R}^n \to \mathbb{R}$ is said to be a stochastic control barrier function (SCBF) if $B \in \mathcal{D}(\mathcal{A})$ and the following conditions are satisfied:*

(i) $B(x) \geq 0$ *for all* $x \in \mathcal{C}$*;*

(ii) $B(x) < 0$ *for all* $x \notin \mathcal{C}$*;*

(iii) $\sup_{u \in \mathcal{U}} \mathcal{A}B(x) \geq 0$.

We refer the control strategy generated by (iii) as

$$\upsilon(x) := \{u \in \mathcal{U} : \mathcal{A}B(x) \geq 0\} \tag{4.18}$$

and the corresponding set of constrained control signals as $\mathbf{u}_\upsilon$.

Remark 4.2.7. *Condition (iii) of the above definition is an analogue of* $\sup_{u \in \mathcal{U}}[L_f B(x) + L_g B(x)u] \geq 0$ *for the deterministic settings. The consequence is such that* $\mathbb{E}^\xi[B(x)] \geq B(x)$ *for all* $x \in \mathcal{C}$*. A relaxation of condition (iii) is given in the deterministic case such that the set invariance can still be guaranteed [137],*

$$\sup_{u \in \mathcal{U}}[L_f B(x) + L_g B(x)u] \geq -\alpha(B(x)),$$

where α is a class-$\mathcal{K}$ function. If $\alpha(x) = kx$ where $k > 0$, under the stochastic settings, the condition formulates an SZCBF and provides a much weaker quantitative estimation of the lower bound of satisfaction probability. In comparison with SZCBF, we provide the worst-case quantification in the following proposition.

Proposition 4.2.8. *Suppose the mapping h is an SCBF with the corresponding control strategy $v(x)$. Let $c = \sup_{x\in\mathcal{C}} h(x)$ and $X_0 = \xi \in \mathcal{C}^\circ$, then under the set of constrained control signals $\mathbf{u}_v$, we have the following worst-case probability estimation:*

$$\mathbb{P}_\xi \left[X_t \in \mathcal{C}^\circ,\ 0 \le t < \infty\right] \ge \frac{h(\xi)}{c}.$$

Proof. Let $V = c - h(x)$, then $V(x) \ge 0$ for all $x \in \mathcal{X}$ and $\mathcal{A}V \le 0$. Let $s = c - h(\xi)$ then $s \le c$ by definition. The result for every finite time interval $t \in [0,T]$ is followed by [80, Lemma 2.1],

$$\mathbb{P}_\xi \left[X_t \in \mathcal{C}^\circ,\ 0 \le t < T\right] \ge 1 - \frac{s}{c}.$$

The result follows by letting $T \to \infty$. □

Definition 4.2.9. *A function $B : \mathcal{X} \to \mathbb{R}$ is called a stochastic control barrier function with relative degree r for system (4.1) if $B \in \mathcal{D}(\mathcal{A}^r)$, and $\mathcal{A} \circ \mathcal{A}^{r-1}h(x) \neq 0$ as well as $\mathcal{A} \circ \mathcal{A}^{j-1}h(x) = 0$ for $j = 1, 2, \ldots, r-1$ and $x \in \mathcal{C}$.*

If the system (4.1) is an r^{th}-order stochastic control system, to steer the process X to satisfy probabilistic set invariance w.r.t. $\mathcal{C}$, we recast the mapping h as an SCBF with relative degree r. For $h \in \mathcal{D}(\mathcal{A}^r)$, we define a series of functions $b_0, b_j : \mathcal{X} \to \mathbb{R}$ such that for each $j = 1, 2, \ldots, r$ $b_0, b_j \in \mathcal{D}(\mathcal{A})$ and

$$\begin{aligned} b_0(x) &= h(x), \\ b_j(x) &= \mathcal{A} \circ \mathcal{A}^{j-1} b_0(x). \end{aligned} \tag{4.19}$$

We further define the corresponding superlevel sets $\mathcal{C}_j$ for $j = 1, 2, \ldots, r$ as

$$\mathcal{C}_j = \{x \in \mathbb{R}^n : b_j(x) \ge 0\}. \tag{4.20}$$

Theorem 4.2.10. *If the mapping h is an SCBF with relative degree r, the corresponding control strategy is given as $v(x) = \{u \in \mathcal{U} : \mathcal{A}^r h(x) \ge 0\}$. Let $c_j =: \sup_{x\in\mathcal{C}_j} b_j(x)$ for each $j = 0, 1, ..., r$ and $X_0 = \xi \in \bigcap_{j=0}^{r} \mathcal{C}_j^\circ$. Then under the set of constrained control signals $\mathbf{u}_v$, we have the following worst-case probability estimation:*

$$\mathbb{P}_\xi[X_t \in \mathcal{C}^\circ,\ 0 \le t < \infty] \ge \prod_{j=0}^{r-1} \frac{b_j(\xi)}{c_j}.$$

Proof. We introduce the notations $p_j := \mathbb{P}_\xi[X_t \in \mathcal{C}_j^\circ,\ 0 \le t < \infty]$, $\hat{p}_j := \mathbb{P}_\xi[X_t \in \mathcal{C}_j^\circ,\ 0 \le t < \infty \mid \mathcal{A}b_j \ge 0]$.

The control signal $u(t) \in \{u \in \mathcal{U} : \mathcal{A}^r b(x) \ge 0\}$ for all $t \ge 0$ provides $\mathcal{A}b_{r-1} \ge 0$. By Proposition 4.2.8,

$$\begin{aligned} p_{r-1} &= \mathbb{P}_\xi[X_t \in \mathcal{C}_{r-1}^\circ,\ 0 \le t < \infty] \\ &= \mathbb{P}_\xi[X_t \in \mathcal{C}_{r-1}^\circ,\ 0 \le t < \infty \mid \mathcal{A}b_{r-1} \ge 0] \\ &= \hat{p}_{r-1}. \end{aligned} \tag{4.21}$$

For $j = 0, 1, ..., r-2$, and $0 \le t < \infty$, we have the following recursion:

$$\begin{aligned} p_j &= \mathbb{P}_\xi[X_t \in \mathcal{C}_j^\circ] \\ &= \mathbb{E}^\xi[\mathbb{1}_{\{X_t \in C_j^\circ\}} \mathbb{1}_{\{X_t \in C_{j+1}^\circ\}}] + \mathbb{E}^\xi[\mathbb{1}_{\{X_t \in C_j^\circ\}} \mathbb{1}_{\{X_t \notin C_{j+1}^\circ\}}] \\ &= \mathbb{P}_\xi[X_t \in C_j^\circ \mid \mathbb{1}_{\{X_t \in C_{j+1}^\circ\}}] \cdot \mathbb{P}_\xi[X_t \in C_{j+1}^\circ] \\ &+ \mathbb{P}_\xi[X_t \in C_j^\circ \mid \mathbb{1}_{\{X_t \notin C_{j+1}^\circ\}}] \cdot \mathbb{P}_\xi[X_t \notin C_{j+1}^\circ], \end{aligned} \tag{4.22}$$

where we have used shorthand notations $\{X_t \in \mathcal{C}_j\} := \{X_t \in \mathcal{C}_j,\ 0 \le t < \infty\}$ and $\{X_t \notin C_{j+1}\} := \{X_t \notin C_{j+1} \text{ for some } 0 \le t < \infty\}$. Indeed, we have

$$X_t = X_t \mathbb{1}_{\{X_t \in C_{j+1}^\circ\}} + X_t \mathbb{1}_{\{X_t \notin C_{j+1}^\circ\}},$$

then

$$\mathbb{E}[X_t] = \mathbb{E}[X_t \mathbb{1}_{\{X_t \in C_{j+1}^\circ\}}] + \mathbb{E}[X_t \mathbb{1}_{\{X_t \notin C_{j+1}^\circ\}}],$$

and (4.22) follows. Note that

$$\begin{aligned} &\mathbb{P}_\xi[X_t \in C_j^\circ \mid \mathbb{1}_{\{X_t \in C_{j+1}^\circ\}}] \\ &\ge \mathbb{P}_\xi[X_t \in C_j^\circ \mid \mathcal{A}b_j \ge 0] = \hat{p}_j, \end{aligned} \tag{4.23}$$

and therefore

$$\mathbb{P}_\xi[X_t \in C_j^\circ \mid \mathbb{1}_{\{X_t \in C_{j+1}^\circ\}}] \cdot \mathbb{P}_\xi[X_t \in C_{j+1}^\circ] \ge \hat{p}_j p_{j+1}.$$

Now define stopping times $\tau_j = \inf\{t : b_j(X_t) \le 0\}$ for $j = 0, 1, ..., r-2$, then $b_j(X_{t \wedge \tau_j}) \ge 0$ a.s.. In addition,

$$X_{t \wedge \tau_j} = \mathbb{1}_{\{\tau_j \le \tau_{j+1}\}} X_{t \wedge \tau_j} + \mathbb{1}_{\{\tau_j > \tau_{j+1}\}} X_{t \wedge \tau_j}$$

Assume the worst scenario, which is for all $t \ge \tau_{j+1}$, we have $\mathcal{A}b_j \le 0$. On $\{\mathcal{A}b_j < 0\} \cap \{\tau_j > \tau_{j+1}\}$, we have $b_j(X_{\tau_{j+1}}) > 0$ and $\mathbb{E}^\xi[b_j(X_{t \wedge \tau_j})] \le b_j(X_{\tau_{j+1}}) - \int_{\tau_{j+1}}^t \varepsilon(s) ds$ for some

$\varepsilon : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ and $t \geq \tau_{j+1}$. Therefore, the process $b_j(X_{t\wedge\tau_j})$ is a nonnegative supermartingale and $\mathbb{P}_\xi[\sup_{T\leq t<\infty} b_j(X_{t\wedge\tau_j}) \geq \lambda] \leq \frac{b_j(X_{\tau_{j+1}})-\int_{\tau_{j+1}}^{T}\varepsilon(s)ds}{\lambda}$ by Doob's supermartingale inequality. For any $\lambda > 0$, we can find a finite $T \geq \tau_{j+1}$ such that $\mathbb{P}_\xi[\sup_{T\leq t<\infty} b_j(X_{t\wedge\tau_j}) \geq \lambda] = 0$. Since λ is arbitrarily selected, we must have $\mathbb{P}_\xi[\sup_{T\leq t<\infty} b_j(X_{t\wedge\tau_j}) > 0] = 0$, which means τ_j is triggered within finite time. On the other hand, on $\{\mathcal{A}b_j < 0\} \cap \{\tau_j \leq \tau_{j+1}\}$, τ_j has been already triggered. Therefore, $\{X_t \notin \mathcal{C}_j^\circ$ for some $0 \leq t < \infty\}$ a.s. given $\{\mathcal{A}b_j < 0\}$. Hence,

$$
\begin{aligned}
&\mathbb{P}_\xi[X_t \in C_j^\circ \,|\mathbb{1}_{\{X_t\notin C_{j+1}^\circ\}}] \\
&\geq \mathbb{P}_\xi[X_t \in C_j^\circ \,|\mathbb{1}_{\{X_t\notin C_{j+1}^\circ,\,\forall t\geq\tau_{j+1}\}}] \\
&\geq \mathbb{P}_\xi[X_t \in C_j^\circ \,|\mathbb{1}_{\{\mathcal{A}b_j<0,\,\forall t\geq\tau_{j+1}\}}] = 0
\end{aligned}
\tag{4.24}
$$

Combining the above, for $j = 0, 1, ..., r-2$, we have

$$
p_j \geq \hat{p}_j p_{j+1},
$$

and ultimately $p_0 \geq \prod_{j=0}^{r-1} \hat{p}_j$. □

Remark 4.2.11. *The above result estimates the lower bound of the safety probability given the constrained control signals $\mathbf{u}_v$. Based on recursion (4.22), we can easily obtain the same result by dropping the last term. However, we argued that under some extreme conditions the worst case may happen. Indeed, we have assumed that $t \geq \tau_{j+1} \implies \mathcal{A}b_j \leq 0$. This conservative assumption is made such that within finite time X will cross the boundary of each $\mathcal{C}_j$.*

Another implicit condition may cause the worst-case lower bound as well, that is when $\bigcup_{j=0}^{r-1}\{\mathcal{A}b_j = 0,\ 0 \leq t < \infty\}$ is a $\mathbb{P}_\xi$-null set. This, however, is practically possible since the controller indirectly influences the value of $\mathcal{A}b_j$ for all $j < r$, the strong invariance of the level set $\{\mathcal{A}b_j = 0\}$ is not guaranteed using QP scheme.

Remark 4.2.12. *A nice selection of controller is to implicitly reduce the total time a sample path spends in $\{\mathcal{A}b_j \leq 0\}$ for each j. However, this is a challenging task by only steering the bottom-level flow, which in turn gives us a future research direction.*

Based on the above discussions, the advantages of our SCBF are mainly in the following two aspects:

- SCBF requires milder control compare to SRCBF, which makes it more applicable in some practical scenarios where safety is as important as feasibility of control actions.

- Compared to SZCBF, it guarantees a better worst-case safety estimation.

However, we should also admit that the SCBF sacrifices some safety probability to compromise with the requirement of milder control. We will use more examples to validate this in the following simulations.

4.3 Simulation Result

In this section, we use two examples to validate our result. We show that the proposed SCBFs have smaller control effort compared to SRCBFs and higher safe probability compared to SZCBFs.

4.3.1 SCBF for Adaptive Cruise Control Model

In the first example, we use an automatic cruise control example as in [33] and [8]. The model is given by the following three-dimensional system:

$$d\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -F_r(x)/M \\ 0 \\ x_2 - x_1 \end{bmatrix} dt + \begin{bmatrix} 1/M \\ 0 \\ 0 \end{bmatrix} udt + \Sigma dW,$$

where x_1 and x_2 denote the velocity of the following vehicle and leading vehicle, respectively, and x_3 is the distance between two vehicles and

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The aerodynamic drag is $F_r(x) = f_0 + f_1 x_1 + f_2 x_1^2$ with $f_0 = 0.1$, $f_1 = 5$, $f_2 = 0.25$ and the mass of the vehicle is $M = 1650$. The initial state is chosen as $[x_1, x_2, x_3] = [15, 20, 150]^T$ and W is a three-dimensional Brownian motion representing uncertainty existing between the speed of two vehicles. The goal of the following vehicle is to achieve a desired velocity $x_d = 22$ while keeping the collision constraint $h(x) = x_3 - \tau x_1 > 0$. The setup of the problem suggests that the following vehicle is achieving a desired speed of 22 while the leading vehicle is blocking in front with a lower speed of 20. The example shows that the controller will prioritize safety in practice as a hard constraint. We use a Lyapunov function

$V(x) = (x_1 - x_d)^2$ to control the velocity of the following vehicle. We use $B(x) = \frac{1}{h(x)}$ as the SRCBF and $h(x)$ as the SCBF. We solve QP problems using SRCBF as

$$
\begin{aligned}
&[u^*, \delta^*] = \arg\min_{u,\delta} \frac{1}{2}(u^2 + \delta^2) \quad \text{s.t.} \\
&\frac{\partial V(x)}{\partial x}(f(x) + g(x)u) + \frac{1}{2}\operatorname{tr}\left(\Sigma^T \Sigma \frac{\partial^2 V(x)}{\partial x^2}\right) \le \delta, \\
&\frac{\partial B(x)}{\partial x}(f(x) + g(x)u) + \frac{1}{2}\operatorname{tr}\left(\Sigma^T \Sigma \frac{\partial^2 B(x)}{\partial x^2}\right) \le \frac{\gamma}{B(x)},
\end{aligned}
$$

and SCBF as

$$
\begin{aligned}
&[u^*, \delta^*] = \arg\min_{u,\delta} \frac{1}{2}(u^2 + \delta^2) \quad \text{s.t.} \\
&\frac{\partial V(x)}{\partial x}(f(x) + g(x)u) + \frac{1}{2}\operatorname{tr}\left(\Sigma^T \Sigma \frac{\partial^2 V(x)}{\partial x^2}\right) \le \delta, \\
&\frac{\partial h(x)}{\partial x}(f(x) + g(x)u) + \frac{1}{2}\operatorname{tr}\left(\Sigma^T \Sigma \frac{\partial^2 h(x)}{\partial x^2}\right) \ge 0.
\end{aligned}
$$

We present the simulation results by showing the control effort $J = u^2$ for SCBF and SRCBF as in Figure 4.1. We choose noise level to be $\sigma_1 = 1$. In Figure 4.1, the red curve is for the SRCBF and the blue curve is for the SCBF. We can find out that SRCBF requires a higher control effort to make the system safe. In practice, this implies that we need larger acceleration in order to keep the system staying within the safe set. As a result, run 50 simulations using both barrier functions and we bound the control and show the safe probability as in Table 4.1. The table shows that for unbounded control, the safe probability using SRCBF is 92% compared to 80% by using SCBF. However, the safe probability drops to only 25% when we use SRCBF while the safe probability is 70% for SCBF under bounded control input. As a result, we can see that the safety probability obtained by SCBF is more robust to saturation of control inputs.

4.3.2 SCBF for Dubins Vehicle with Disturbance

In the second example, we test our SCBF using a differential drive model as in [84]:

$$
ds = d\begin{bmatrix} x \\ y \\ \vartheta \end{bmatrix} = \begin{bmatrix} \cos\vartheta & 0 \\ \sin\vartheta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} dt + \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} dW,
$$

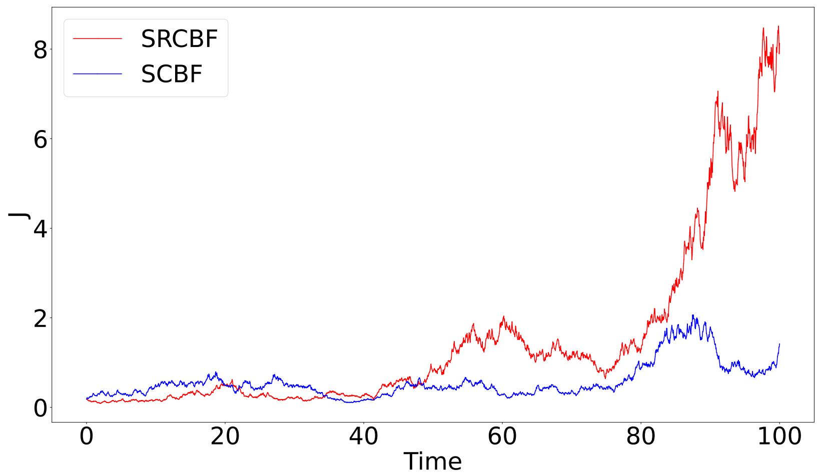


Figure 4.1. Plot of control effort $J = u^2$ for Adaptive Cruise Control.

	SRCBF	SCBF
Unbounded control	92%	80%
Bounded control	25%	70%

Table 4.1: Safe probability of adaptive cruise control between SRCBF and SCBF under bounded and unbounded control inputs. We sample 50 trajectories for each case and calculate the safe probability. The simulation step time is chosen to be $t = 0.05s$. The total simulation time is 100s.

where x and y are the planar positions of the center of the vehicle, ϑ is its orientation, $v = 2$ is its forward velocity, the angular velocity w is the control of the system and W is a standard Brownian motion representing uncertainty in x and y. The working space of the vehicle is a circle centered at $(0, 0)$ with a radius of $r = 3$. Our objective is to control the vehicle within the working space and as a result, the safety requirement can be encoded using $h(s) = r^2 - x^2 - y^2$. We solve QP problems using constraints that are obtained from Theorem 4.2.10 as

$$\mathcal{A} \circ \mathcal{A}(9 - x^2 - y^2) \geq 0 \tag{4.25}$$

We first test safe probability for different initial positions. We use our SCBF to test this probability under different noise levels within $[0, 0.3]$. For simplicity, we set $\sigma_1 = \sigma_2 = \sigma$. For each value of σ, we sample 1000 trajectories and calculate the safe probability. We also compare the result between SCBF and a higher-order SZCBF, which is an extension of the work [118] as

$$h_1(s) = \mathcal{A}h(s) + \alpha_1 h(s),$$
$$h_2(s) = \mathcal{A}h_1(s) + \alpha_2 h(s).$$

The constraints from SZCBF that $h_2(s) \geq 0$ are used to solve QP problems. We first compare the safe probability between SCBF and SZCBF for different noise level within $[0, 0.2]$. For each value of σ, we randomly sample 1000 initial points and generate 1000 trajectories accordingly. We calculate safe probability over this 1000 trajectories and plot the result as in Figure 4.2. We can find out that SCBF has a better safe probability over SZCBF. In another experiment, we randomly sample 10 initial points. Then we generate 500 trajectories using SCBF and SZCBF for each initial point. We fix the noise to be $\sigma = 0.2$. The result is shown as in Figure 4.3. From the figure, we can find out that SCBF has a overall better performance than SZCBF under randomly sampled initial points.

4.4 Conclusion

This chapter considers the pros and cons of the existing formulations for stochastic barrier functions. We propose stochastic control barrier functions (SCBFs) for safety-critical control of stochastic systems and extend the worst-case safety probability estimation to higher-order SCBFs. Theoretically, we show that the proposed SCBFs provide good trade-offs between the imposed control constraints and the conservatism in the estimation of safety probability. The proposed scheme is also utilized to control an automatic cruise control model and a Dubins vehicle model in simulation.

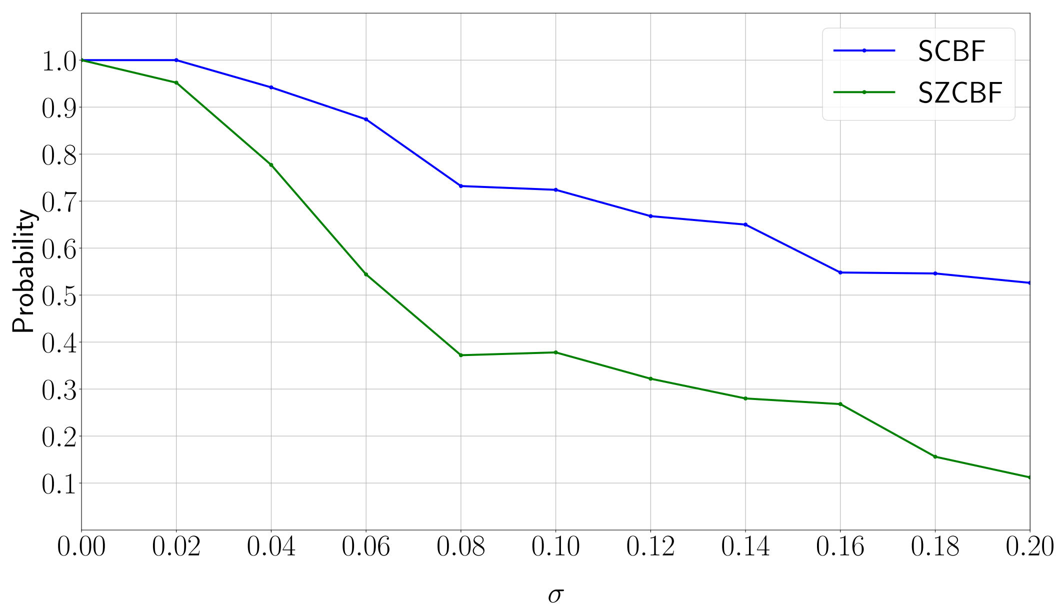


Figure 4.2. Safe probability between SCBF and SZCBF. We compare noise level within [0,0.2]. For each value of σ, we sample 1000 initial points to calculate safe probability.

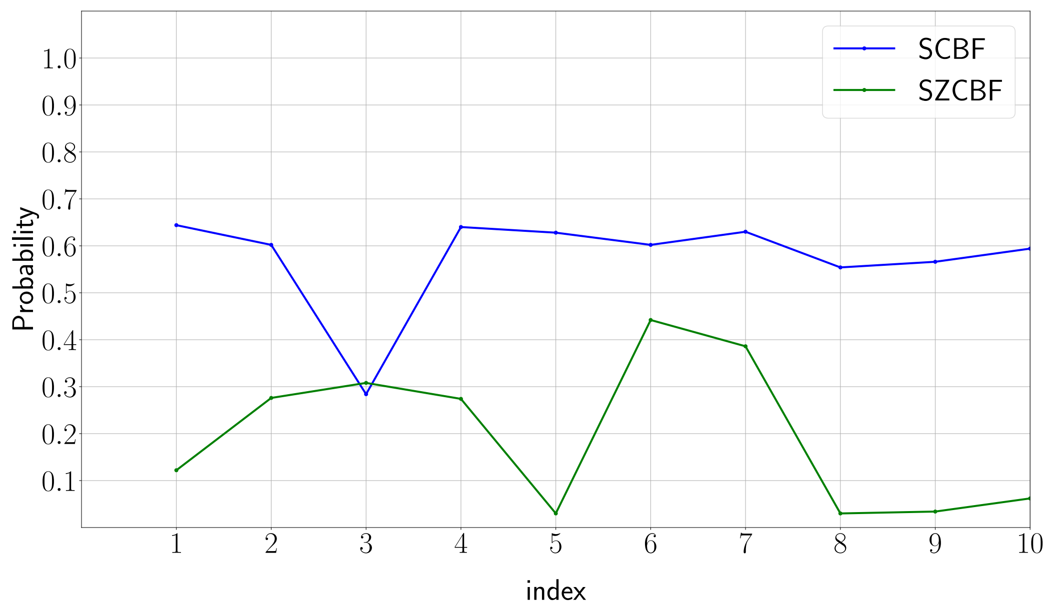


Figure 4.3. Safe probability of 10 randomly sample initial points. For each initial point, we sample 500 trajectories using SCBF and SZCBF respectively. The horizontal axis represents the index of the initial points.

Chapter 5

Data-Driven Learning of Safety-Critical Control with Stochastic Control Barrier Function

In Chapter 4, we discuss stochastic control barrier functions to provide safety guarantee for SDEs driven by Brownian motions. However, in some practical scenarios, we do not have precise information about the stochastic systems. As a result, in this chapter, we study safety-critical control problems for stochastic systems with unknown diffusion terms.

5.1 Preliminary and Problem Definition

5.1.1 System Description

Given a state space $\mathcal{X} \subseteq \mathbb{R}^n$ and a (compact) set of control values $\mathcal{U} \subset \mathbb{R}^p$, consider a continuous-time stochastic dynamical system

$$dX_t = (f(X_t) + g(X_t)u(t))dt + \sigma(X_t)dW_t, \; X_0 = x, \tag{5.1}$$

where $u : \mathbb{R}_{\geq 0} \to \mathcal{U}$ is a bounded measurable control signal; W represents a d-dimensional standard Wiener process; $f : \mathcal{X} \to \mathbb{R}^n$ is a locally Lipschitz nonlinear vector field; $g : \mathcal{X} \to \mathbb{R}^{n\times p}$ and $\sigma : \mathcal{X} \to \mathbb{R}^{n\times d}$ are smooth mappings.

Definition 5.1.1 (Weak solutions)**.** *For each fixed signal u, the system* (5.1) *admits a weak solution if there exists a filtered probability space $(\Omega^\dagger, \mathscr{F}^\dagger, \{\mathscr{F}_t^\dagger\}, \mathbb{P}^\dagger)$, where a Wiener process W is defined and a pair (X^u, W) is adapted, such that X^u solves the SDE* (5.1)*.*

The infinitesimal generator of X is defined as in Definition 4.1.4 and can be calculated according to (4.4).

5.1.2 Problem Formulation

For the rest of this chapter, we consider a safe set of the form

$$\mathcal{C} := \{x \in \mathcal{X} : h(x) \geq 0\}, \tag{5.2}$$

where $h \in C^2(\mathbb{R}^n)$. We also define the boundary and interior of $\mathcal{C}$ explicitly as below

$$\partial\mathcal{C} := \{x \in \mathcal{X} : h(x) = 0\}, \tag{5.3}$$

$$\mathcal{C}^\circ := \{x \in \mathcal{X} : h(x) > 0\}. \tag{5.4}$$

The objective of this chapter is to control the stochastic system (5.1) with an unknown diffusion term to stay inside the safe set. Given the definition of controlled probabilistic invariance as in Definition 4.1.10, the problem is defined as follows.

Problem 5.1.2. *Given system* (5.1) *with the b unknown, a compact set $\mathcal{C} \subseteq \mathcal{X}$ defined by* (6.2)*, a point $x \in \mathcal{C}^\circ$, and a $p \in [0,1]$, design a control strategy κ such that under $\mathbf{u}_\kappa$, the interior $\mathcal{C}^\circ$ is controlled p-invariant for the resulting solutions to* (5.1)*.*

Remark 5.1.3. *In this chapter, we assume we have full knowledge of the drift term of the system, i. e., f and g in* (5.1)*. For uncertainty in drift term, refer [129, 137] for detail.*

5.2 Worst-Case Probabilistic Quantification via Stochastic Control Barrier Function

There witnesses a surge of applications of control synthesis for probabilistic safety problems using stochastic control barrier functions. Two commonly-used types of stochastic control barrier functions, reciprocal type (SRCBF) [33] and zeroing type (SZCBF) [118], are investigated. The recent work [138] identified the pros and cons of SRCBF and SZCBF and proposed a middle-ground type SCBF[1] as in [138, Definition III.6].

[1]We use this acronym for the specified notion in [138, Definition III.6] rather than the general type of stochastic control barrier functions.

Note that the function h in (5.2) is already a potential SCBF candidate. We need to further impose a condition on the drift term of its Itô derivative along the sample paths, i.e. $\mathcal{A}h$, to make it effective.

Due to the lack of information of the diffusion term in (5.1), we are unable to capture the correction term in the Itô derivative of the nominal barrier function h along sample paths. We use a data-driven method to approximate the function $\mathcal{A}h$, and impose similar barrier conditions on the approximated $\hat{\mathcal{A}}h$ for safety-critical control. Based on the partial observation of data, we show in this section that a degree of robustness in the barrier condition is necessary to balance the inaccuracy of data. A similar approach can be applied to derive the robustness for the other types of stochastic control barrier functions.

We suppose that data is sampled without control inputs. Then for each x, the law $\mathbf{P}^x$ process X is independent of u. We further define the stopping time

$$\tau := \inf\{t \geq 0 : X_t \in \partial\mathcal{C}\}$$

for each sampled process. Let $\mathfrak{C}$ denote a finite subset of $\mathcal{C}$.

5.2.1 Probability Estimation Based on Partially Observed Data

We make the following assumptions for the rest of derivation. We show in the next subsection that the assumptions are feasible for compact $\mathcal{C}$.

Assumption 5.2.1. *Let $\hat{\mathcal{A}}h$ be the approximation of $\mathcal{A}h$ based on the training set $\mathfrak{C}$. We assume that*

(i) For any $y \in \mathcal{C}$ and any $\varepsilon > 0$, there exists an $x \in \mathfrak{C}$ such that[2]

$$\begin{aligned}&\mathbf{E}^{y,u} \sup_{t\in[0,\tau]} |\hat{\mathcal{A}}h(X_t^u) - \mathcal{A}h(X_t^u)| \\ \leq &\mathbf{E}^{x,u} \sup_{t\in[0,\tau]} |\hat{\mathcal{A}}h(X_t^u) - \mathcal{A}h(X_t^u)| + \varepsilon.\end{aligned} \tag{5.5}$$

(ii) For any $\varsigma \in (0,1]$, there exists a probability measure $\mathbb{P}$ with marginals $\mathbf{P}^x$ for all $x \in \mathfrak{C}$ such that

$$\mathbb{E} \sup_{x\in\mathcal{C}} |\mathcal{A}h(x) - \hat{\mathcal{A}}h(x)| \leq \varsigma. \tag{5.6}$$

[2]Note that $\tau < \tau_{\text{ex}}$ with probability 1.

Furthermore, we assume that both $\hat{\mathcal{A}}h$ and $\mathcal{A}h$ are Lipschitz continuous on the compact set $\mathcal{C}$.

We apply the approximated function $\hat{\mathcal{A}}h$ and show the worst-case safety probability of the controlled process under policy generated by the following robust scheme.

Proposition 5.2.2. *Suppose we are given arbitrary $\varsigma > 0, \varepsilon > 0$ and training set $\mathfrak{C}$. Let $\hat{\mathcal{A}}h$ be generated as in Assumption 5.2.1. Suppose that $\sup_{u\in\mathcal{U}} \hat{\mathcal{A}}h(x) \geq \varsigma + \varepsilon$ for all $x \in \mathcal{C}$. Let $\upsilon(x) = \{u \in \mathcal{U} : \hat{\mathcal{A}}h(x) \geq \varsigma + \varepsilon\}$. Then for any $x \in \mathcal{C}^\circ$ and $u \in \mathbf{u}_\upsilon$, we have*

$$\mathbf{P}^{x,u}[X_t^u \in \mathcal{C}^\circ,\ 0 \leq t < \infty] \geq \frac{h(x)}{\sup_{y\in\mathcal{C}} h(y)}$$

Proof. Let $c = \sup_{y\in\mathcal{C}} h(y)$ and set $V = c - h$. Then for all $x \in \mathcal{C}^\circ$, we have $V(x) > 0$ and $\hat{\mathcal{A}}V(x) \leq -(\varsigma + \varepsilon)$. Note that

$$\mathbf{E}^{x,u}[V(X_{\tau\wedge t}^u)] = V(x) + \mathbf{E}^{x,u}\left[\int_0^{\tau\wedge t} \mathcal{A}V(X_s^u) ds\right] \tag{5.7}$$

and by assumption,

$$\begin{aligned}
&\mathbf{E}^{x,u}\left[\int_0^{\tau\wedge t} \mathcal{A}V(X_s^u) ds\right] \\
=&\mathbf{E}^{x,u}\left[\int_0^{\tau\wedge t} \mathcal{A}V(X_s^u) - \hat{\mathcal{A}}V(X_s^u)\ ds\right] \\
&+ \mathbf{E}^{x,u}\left[\int_0^{\tau\wedge t} \hat{\mathcal{A}}V(X_s^u)\ ds\right] \\
\leq& \int_0^{\tau\wedge t} \mathbf{E}^{x,u}|\mathcal{A}V(X_s^u) - \hat{\mathcal{A}}V(X_s^u)|\ ds - (\varsigma+\varepsilon)\cdot(\tau\wedge t) \\
\leq& \int_0^{\tau\wedge t} \mathbb{E} \sup_{s\in[0,\tau]} |\mathcal{A}V(X_s^u) - \hat{\mathcal{A}}V(X_s^u)|\ ds - \varsigma\cdot(\tau\wedge t) \\
\leq& \int_0^{\tau\wedge t} \mathbb{E}\sup_{x\in\mathcal{C}} |\mathcal{A}V(x) - \hat{\mathcal{A}}V(x)|\ ds - \varsigma\cdot(\tau\wedge t) \leq 0,
\end{aligned} \tag{5.8}$$

where the fifth line of the above is to transfer information from arbitrary $x \in \mathcal{C}$ to the data used in $\mathfrak{C}$. The mismatch of measure provides an extra error of ε. Hence, by (5.7), we have

$$\mathbf{E}^{x,u}[V(X_{\tau\wedge t}^u)] \leq V(x), \quad \forall t \geq 0. \tag{5.9}$$

On the other hand, for all $t \geq 0$,

$$\begin{aligned} \mathbf{E}^{x,u}[V(X^u_{\tau\wedge t})] &\geq \mathbf{E}^{x,u}[\mathbb{1}_{\{\tau\leq t\}}V(X^u_{\tau\wedge t})] \\ &\geq \mathbf{P}^{x,u}[\tau \leq t]\cdot \mathbf{E}^{x,u}[V(X^u(\tau)] \\ &> c\cdot \mathbf{P}^{x,u}[\tau\leq t]. \end{aligned} \tag{5.10}$$

Therefore, by (5.9) and (5.10), we have

$$\mathbf{P}^{x,u}[\tau \leq t] < \frac{V(x)}{c},\ \forall t \geq 0. \tag{5.11}$$

Sending $t \to \infty$ we get $\mathbf{P}^{x,u}[\tau < \infty] \leq \frac{V(x)}{c}$ for all $x \in \mathcal{C}^\circ$. Rearranging this we can obtain the conclusion. □

5.2.2 Feasibility of Assumption

Note that for the compact set $\mathcal{C}$ and for sufficiently dense training data, the conditions in Assumption 5.2.1 can be satisfied theoretically. We will show that both (i) and (ii) of Assumption 5.2.1 require the selection of the training data but separately. Before proceeding to the explanation, we introduce the following concepts.

Definition 5.2.3. ***(Weak convergence of measures and processes):*** *Given any separable metric space $(\mathcal{S}, \varrho)$, a sequence of probability measure $\{\mathbf{P}^n\}$ on $\mathscr{B}(\mathcal{S})$ is said to weakly converge to $\mathbf{P}$ on $\mathscr{B}(\mathcal{S})$, denoted by $\mathbf{P}^n \rightharpoonup \mathbf{P}$, if for all $f \in C_b(\mathcal{S})$ we have $\lim_{n\to\infty}\int_{\mathcal{S}} f\ d\mathbf{P}^n = \int_{\mathcal{S}} f\ d\mathbf{P}$. A sequence $\{X^n\}$ of continuous processes X^n with law $\mathbf{P}^n$ is said to weakly converge (on $[0,T]$) to a continuous process X with law $\mathbf{P}$, denoted by $X^n \rightharpoonup X$, if for all $f \in C_b(C([0,T];\mathbb{R}^n))$ we have $\lim_{n\to\infty}\mathbf{E}^n[f(X^n)] = \mathbf{E}[f(X)]$.*

The following proposition demonstrates a compactness of weak solutions starting from a compact set in a weak sense as in Definition 5.2.3. We provide the rephrased version based on [76, Theorem 1] and [77, Corollary 1.1, Chap 3] as follows. A detailed explanation can be found in [91].

Proposition 5.2.4. *Given any compact set $\mathcal{C}$ and its associated first-hitting time τ, given any sequence of stopped weak solutions $\{(X^n)^\tau\}_{n=1}^\infty$ with $X^n(0) = x_n$, there exists a subsequence $\{(X^{n_k})^\tau\}$ and a process X with $X(0) = x$ such that $x_{n_k} \to x$ and $(X^{n_k})^\tau \rightharpoonup X^\tau$.*

Justification of Assumption 5.2.1(i)

We observe that for each x in a compact set $\mathcal{C}$, for any fixed $T > 0$, the quantity $\sup_{t\in[0,\tau\wedge T]} |\mathcal{A}h(\cdot) - \hat{\mathcal{A}}h(\cdot)|$ is a bounded function on the canonical space generated by $\mathcal{C}$ with measure $\mathbf{P}^x$. In view of Definition 5.2.3 and Proposition 5.2.4, the quantity

$$\left\{\mathbf{E}^x \sup_{t\in[0,\tau\wedge T]} |\mathcal{A}h(X_t^u) - \hat{\mathcal{A}}h(X_t^u)|\right\}_{x\in\mathcal{C}}$$

forms a compact set (in the conventional sense). By the boundedness assumption on $\mathcal{C}$, we have $\tau < \infty$ $\mathbf{P}^x$-a.s. for every $x \in \mathcal{C}$. Therefore, sending T to infinity, we still have the compactness for

$$\left\{\mathbf{E}^x \sup_{t\in[0,\tau]} |\mathcal{A}h(X_t^u) - \hat{\mathcal{A}}h(X_t^u)|\right\}_{x\in\mathcal{C}}.$$

By choosing $\mathfrak{C}$ sufficiently dense in $\mathcal{C}$, for each given $\varepsilon > 0$, we are able to build the ε-net with centers in $\mathfrak{C}$ such that for any arbitrary $y \in \mathcal{C}$, there exists an $x \in \mathfrak{C}$ such that $\mathcal{A}h - \hat{\mathcal{A}}h$ are weakly ε-close to each other in the sense of (5.5).

We then verify the feasibility of (ii) of Assumption 5.2.1.

Approximating $\mathcal{A}h$ over a finite set

Note that, following the procedure as in [98], we are able to approximate $\mathcal{A}h$ by some $\tilde{\mathcal{A}}h$ at one single point $x \in \mathbb{R}^n$ at a time, whose precision is measured under the corresponding probability $\mathbb{P}^x := \otimes_{i=1}^{\infty}\mathbf{P}^x$. However, to fit the assumption, we need the precision to be measured in L_1 sense.

By [98, Theorem 6], for each $x \in \mathbb{R}^n$, we can utilize Lipschitz continuity of f, g, b and the relation

$$\tilde{\mathcal{A}}_1 h(x) = \frac{\mathbf{E}^x[h(X_{\tau_s}^u)] - h(x)}{\tau_s}$$

at some deterministic sampling time τ_s to obtain the first-step approximation

$$|\tilde{\mathcal{A}}_1 h(x) - \mathcal{A}h(x)| \le \delta, \tag{5.12}$$

where $\delta = C_1\tau_s + C_2\sqrt{\tau_s}$, and $C_1, C_2 > 0$ are constants generated by Lipschitz continuity. The precision δ can be arbitrarily small.

Since $\tilde{\mathcal{A}}_1 h(x)$ has used $\mathbf{E}^x[h(X^u_{\tau_s})]$, the authors in [98] then applied the law of large numbers (LLN) to approximate $\mathbf{E}^x[h(X^u_{\tau_s})]$ by $\frac{1}{n}\sum_{i=1}^n h(X^{u,(i)}_{\tau_s})$ with i.i.d. $h(X^{u,(i)}_{\tau_s})$ draw from $\mathbf{P}^x$ at the marginal time τ_s. The approximation

$$\tilde{\mathcal{A}}h = \frac{\frac{1}{n}\sum_{i=1}^n h(X^{u,(i)}_{\tau_s}) - h(x)}{\tau_s}$$

creates errors in probability w.r.t. $\mathbb{P}^x$ as in [98, Theorem 12], i.e., for each $\beta \in (0,1]$, there exists a $\tilde{\delta}$ such that

$$\mathbb{P}^x[|\mathcal{A}h(x) - \tilde{\mathcal{A}}h(x)| \leq \tilde{\delta}] > 1 - \beta.$$

Note that the only place that we introduce $\mathbb{P}^x$ is when we use LLN. We need to leverage the convergence in the L_1 sense, i.e.,

$$\mathbb{E}^x \left| \frac{1}{n}\sum_{i=1}^n h(X^{u,(i)}_{\tau_s}) - \mathbf{E}^x[h(X^u_{\tau_s})] \right| \to 0. \tag{5.13}$$

This is indeed the case as an existing result, even though it is seldom mentioned. Combining (5.13) and (5.12), we can easily obtain that for each $x \in \mathbb{R}^n$, for any $\delta > 0$, there exists a sufficiently large n such that

$$\mathbb{E}^x \left| \tilde{\mathcal{A}}h(x) - \mathcal{A}h(x)) \right| \leq \delta. \tag{5.14}$$

We provide the proof for the L_1 convergence of LLN in the Appendix.

Repeating the same process for x over a finite set $\mathfrak{C}$ gives

$$\sup_{x\in\mathfrak{C}} \mathbb{E}\left[|\mathcal{A}h(x) - \tilde{\mathcal{A}}h(x)| \right] \leq \delta, \tag{5.15}$$

where $\mathbb{E}$ is the associated expectation w.r.t. $\mathbb{P} := \otimes_{x\in\mathfrak{C}}\mathbb{P}^x$.

Optimization Error

For any $\eta > 0$, we assume there exists an optimizer that can learn an approximation $\hat{\mathcal{A}}h$ based on data $\left\{ \tilde{\mathcal{A}}h(x) : \ x \in \mathfrak{C} \right\}$ such that

$$\sup_{x\in\mathfrak{C}} |\hat{\mathcal{A}}h(x) - \tilde{\mathcal{A}}h(x)| < \eta. \tag{5.16}$$

Generalization Error

By continuity of $\hat{\mathcal{A}}h(x)$ and $\mathcal{A}h(x)$, there exists some $x^* \in \mathcal{C}$ such that

$$\sup_{x\in\mathcal{C}} |\hat{\mathcal{A}}h(x) - \mathcal{A}h(x)| = |\hat{\mathcal{A}}h(x^*) - \mathcal{A}h(x^*)|.$$

For any $\vartheta > 0$, by choosing $\mathfrak{C}$ to be sufficiently dense in $\mathcal{C}$ and the Lipschitz continuity of $\hat{\mathcal{A}}h(x)$ and $\mathcal{A}h(x)$ on $\mathcal{C}$, there exists some $y \in \mathfrak{C}$ such that

$$|\hat{\mathcal{A}}h(x^*) - \hat{\mathcal{A}}h(y)| \le \vartheta, \quad |\mathcal{A}h(x^*) - \mathcal{A}h(y)| \le \vartheta.$$

It follows that

$$\begin{aligned}
&\mathbb{E}\left[\sup_{x\in\mathcal{C}} |\hat{\mathcal{A}}h(x) - \mathcal{A}h(x)|\right] \\
=&\mathbb{E}\left[|\hat{\mathcal{A}}h(x^*) - \mathcal{A}h(x^*)|\right] \\
=&\mathbb{E}\left[|\hat{\mathcal{A}}h(y) - \mathcal{A}h(y) + \hat{\mathcal{A}}h(x^*) - \hat{\mathcal{A}}h(y)\right. \\
&\qquad\qquad\qquad \left. + \mathcal{A}h(y) - \mathcal{A}h(x^*)|\right] \\
\le&\mathbb{E}\left[|\hat{\mathcal{A}}h(y) - \mathcal{A}h(y)|\right] + 2\vartheta \\
=&\mathbb{E}\left[|\hat{\mathcal{A}}h(y) - \tilde{\mathcal{A}}h(y) + \tilde{\mathcal{A}}h(y) - \mathcal{A}h(y)|\right] + 2\vartheta \\
\le&\mathbb{E}\left[\sup_{y\in\mathfrak{C}} |\hat{\mathcal{A}}h(y) - \tilde{\mathcal{A}}h(y)|\right] + \mathbb{E}\left[|\tilde{\mathcal{A}}h(y) - \mathcal{A}h(y)|\right] + 2\vartheta \\
\le&\eta + \sup_{y\in\mathfrak{C}} \mathbb{E}\left[|\tilde{\mathcal{A}}h(y) - \mathcal{A}h(y)|\right] + 2\vartheta \\
\le&\eta + \delta + 2\vartheta \le \varsigma,
\end{aligned}$$

where ς is from Assumption 5.2.1(ii), provided that we choose η, δ, and ϑ sufficiently small.

Remark 5.2.5. *The final $\mathfrak{C}$ should be chosen based on all of the above criteria such that (i) and (ii) of Assumption 5.2.1 can both be satisfied.*

5.3 Data-Driven Stochastic Control Barrier Function Scheme for Safety-Critical Control

In this section, we describe how do we use supervised learning to implement the Data-driven Stochastic Control Barrier Function (DDSCBF) scheme and do safety-critical control for stochastic systems with an unknown diffusion part. We use a neural network to approximate the infinitesimal generator of a SCBF. The detail of data collecting and training is explained below.

Given an SDE as in (5.1), we have

$$\mathcal{A}h(x) = L_f h(x) + L_g h(x)u + \frac{1}{2}\operatorname{tr}\left[(\sigma\sigma^T)(x)\cdot h_{xx}(x)\right].$$

Since the only unknown part in $\mathcal{A}h(x)$ is only $\frac{1}{2}\operatorname{tr}\left[(\sigma\sigma^T)(x)\cdot h_{xx}(x)\right]$. As we can see, this term is a function of x only, so we can define a function $\Delta(x) = \frac{1}{2}\operatorname{tr}\left[(\sigma\sigma^T)(x)\cdot h_{xx}(x)\right]$ and accordingly

$$\mathcal{A}h(x) = L_f h(x) + L_g h(x)u + \Delta(x).$$

We use supervised learning to learn $\Delta(x)$ (and hence some $\hat{A}h(x)$) that approximates $\mathcal{A}h(x)$.

Next we describe how to obtain the training data. We use a sampling method to collect data in order to learn $\Delta(x)$. First we sample a set with N initial points $\{x_1, x_2, ..., x_N\}$. At the initial point x_i for $i \in \{1, 2, ..., N\}$, we sample n one-step transitions and reach to the next stage x_{ij} for $j \in \{1, 2, ..., n\}$. We reset the point back to x_i after each one-step transition. According to [98], $\tilde{\mathcal{A}}h(x_i)$ can be estimated numerically by

$$\tilde{\mathcal{A}}h(x_i) = \frac{\frac{1}{n}\sum_{j=0}^{n} h(x_{ij}) - h(x_i)}{\Delta t}. \tag{5.17}$$

As a result, obtain $\tilde{\Delta}(x_i)$ by

$$\tilde{\Delta}(x_i) = \tilde{\mathcal{A}}h(x_i) - (L_f h(x_i) + L_g h(x_i)u).$$

We then add $\{x_i, \tilde{\Delta}(x_i)\}$ into a data set D, constructing a data set D is of dimension N. Next we use learning to fit the data set. The process of collecting training data is shown as in shown as in Algorithm 2.

Once we have collected the data set D, we construct a neural network $\mathcal{N}(x)$ and specify a loss function $\mathcal{L}$ using minimum square error (MSE). We use supervised learning to find the parameters of the network such that the $\frac{1}{N}\sum_i^N \mathcal{L}(\mathcal{N}(x_i), \tilde{\Delta}(x_i))$ is minimized. This implies that the neural network $\mathcal{N}(x)$ will approximate the function $\Delta(x)$. So the derivative of the SCBF $\mathcal{A}h(x)$ will be approximated by $\hat{\mathcal{A}}h(x) := L_f h(x) + L_g h(x)u + \mathcal{N}(x)$. As a result, we can use this approximated derivative of SCBF as QP constraints to guarantee safety-critical control for stochastic systems with unknown diffusion part as in [138]. The overall theoretical analysis of guarantees is shown in 5.2.2 under Assumption 5.2.1.

Algorithm 2 Data-driven learning algorithm of SCBF

Require: An SDE as in (5.1), the number of initial points N, the number of trajectories sampled at each initial point n, an empty data-set D, a time step Δt, an initial neural network $\mathcal{N}(x)$.

```
Initialize neural network
Sample N initial points {x_1, x_2, ..., x_N}
for i in N do
    for j in n do
        Get x_ij from x_i using Euler-Maruyama method according to (5.1) using Δt [56]
        Calculate h(x_ij)
    end for
    Estimate Ãh(x_i) using (5.17)
    Calculate training data using
```

$$\tilde{\Delta}(x_i) = \tilde{\mathcal{A}}h(x_i) - (L_f h(x_i) + L_g h(x_i)u) \tag{5.18}$$

```
    Add training data into data-set, D ← {x_i, Δ̃(x_i)}
end for
```

5.4 Simulation Result

5.4.1 Inverted Pendulum with Unknown Diffusion

In the first example, we test our result using an inverted pendulum. The system is an SDE of the form

$$d\begin{bmatrix}\vartheta \\ \dot{\vartheta}\end{bmatrix} = \begin{bmatrix}\dot{\vartheta} \\ \frac{g}{l}\sin\vartheta\end{bmatrix}dt + \begin{bmatrix}0 \\ \frac{1}{ml^2}\end{bmatrix}udt + \begin{bmatrix}0.1\vartheta \\ 0\end{bmatrix}dW,$$

with the state $x = [\vartheta, \dot{\vartheta}]^T$, gravitational acceleration $g = 10$ and length $\ell = 0.7$. We assume that the diffusion part $\sigma(x) = [0.1\vartheta, 0]^T$ is unknown to us. Consider the control barrier function

$$h(x) = c - x^T P x,$$

where

$$P = \begin{bmatrix}\sqrt{3} & 1 \\ 1 & \sqrt{3}\end{bmatrix}.$$

So

$$h = 0.2 - \sqrt{3}\vartheta^2 - 2\vartheta\dot{\vartheta} - \sqrt{3}\dot{\vartheta}^2.$$

Accordingly, we have

$$\begin{aligned} L_f h(x) + L_g h(x) u = &-(2\sqrt{3}\vartheta + 2\dot{\vartheta})\dot{\vartheta} \\ &- (2\vartheta + 2\sqrt{3}\dot{\vartheta}) \cdot \frac{g}{l} \sin\vartheta - \frac{2\vartheta + 2\sqrt{3}\dot{\vartheta}}{ml^2} u. \end{aligned}$$

We follow Algorithm 2 to obtain training data. We randomly sample 200 points within state space and at each point x_i, we simulate 50000 one-step transitions to get to the next point x_{ij}. The time step of the transition is $t = 0.01s$. Then we estimate $\hat{\mathcal{A}}h(x_i)$ using (5.17). As a result, the training data is obtained according to (5.18). We use a neural network with two hidden layers, with 100 and 30 nodes for each layer, respectively, to fit the training data. We train the network with 500 epochs and compare the training result with the analytic result calculated as

$$\begin{aligned} \frac{1}{2}\operatorname{tr}\left[(\sigma\sigma^T)(x) \cdot h_{xx}(x)\right] &= \frac{1}{2} tr\Bigg(\begin{bmatrix} 0.1\vartheta & 0 \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} -2\sqrt{3} & -2 \\ -2 & -2\sqrt{3} \end{bmatrix} \cdot \begin{bmatrix} 0.1\vartheta \\ 0 \end{bmatrix} \Bigg) \\ &= -\sqrt{3} \cdot (0.1\vartheta)^2. \end{aligned} \tag{5.19}$$

The result of learning is shown in the Figure 5.1. The black dots are the training data, the yellow curve is the analytic result calculated as in (5.19) and the red dots are the neural network output for validation after training.

We also test the control result of applying the DDSCBF scheme. We compare the safe rate using the real SCBF, the DDSCBF scheme and CBF on the unknown system. For each case, we randomly sample 1000 trajectories and compute the safe rate. As shown in the Table 5.1, the system is sensitive to the noise that all the trajectories are unsafe when using CBF. But after applying the DDSCBF scheme, the success rate of the system is over 90%.

5.4.2 A Nonlinear Numerical Model with Unknown Diffusion

In the second example, we test our result using a nonlinear system given by the following stochastic differential equation:

$$d\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -0.6x_1 - x_2 \\ x_1^3 \end{bmatrix} dt + \begin{bmatrix} 0 \\ x_2 \end{bmatrix} udt + \begin{bmatrix} 0 \\ \sigma(x_2) \end{bmatrix} dW.$$

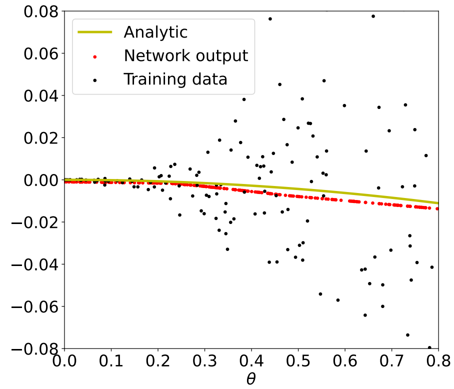


Figure 5.1. Training result of $\frac{1}{2}\operatorname{tr}\left[(\sigma\sigma^T)(x)\cdot h_{xx}(x)\right]$. The black dots are the training data. The yellow curve is the analytic result, which is the true value $-\sqrt{3}\cdot(0.1\vartheta)^2$ and the red dots are the output of the neural network after training.

	Success rate
SCBF	92%
DDSCBF	91%
CBF	0%

Table 5.1: The success rate of using SCBF, DDSCBF scheme and CBF for pendulum system over 1000 runs.

The control objective is to reach the origin $(0, 0)$ and the safe region is defined as

$$h = -x_2^2 - x_1 + 1 > 0.$$

The generator of h is calculated as

$$\mathcal{A}(h) = 0.6x_1 + x_2 - 2x_1^3 x_2 - 2x_2^2 u - \sigma(x_2)^2.$$

We use the same number of sample points and number of transitions at each point as in the first example. The structure of the neural network is also the same as in the first example. The training result is shown in the Figure 5.2.

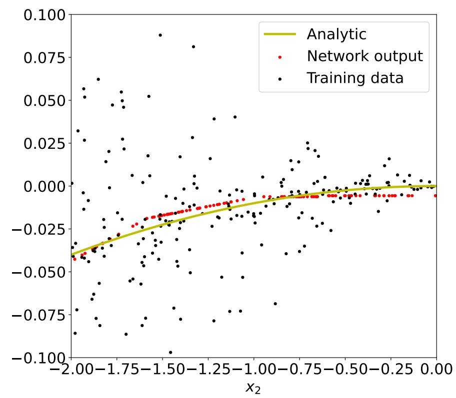


Figure 5.2. Training result of $\frac{1}{2}\operatorname{tr}\left[(\sigma\sigma^T)(x) \cdot h_{xx}(x)\right]$ for $\sigma(x_2) = 0.1x_2$. The black dots are the training data. The yellow curve is the analytic value and the red dots are the output of the neural network after training.

	$\sigma(x_2) = 0.1x_2$	$\sigma(x_2) = 0.15x_2$
SCBF	86.8%	84.3%
DDSCBF	85.2%	83%
CBF	77.3%	57.5%

Table 5.2: The success rate of using SCBF, learned SCBF and CBF for nonlinear system over 1000 runs under different noise with $\sigma(x_2) = 0.1x_2$ and $\sigma(x_2) = 0.15x_2$.

We use a CLF to control the deterministic system, i.e., $\sigma(x_2) = 0$ and the result is shown in Figure 5.3a. Also the control using CLF and CBF for the deterministic system is shown in Figure 5.3b. We can see that CBF will guarantee a safe trajectory for the deterministic system. However, when the system has a diffusion part of $b(x_2) = 0.1x_2$, the noise will make the trajectory unsafe using CBF as shown in Figure 5.3c. By using our DDSCBF scheme, the trajectory is within the safe region as shown in Figure 5.3d.

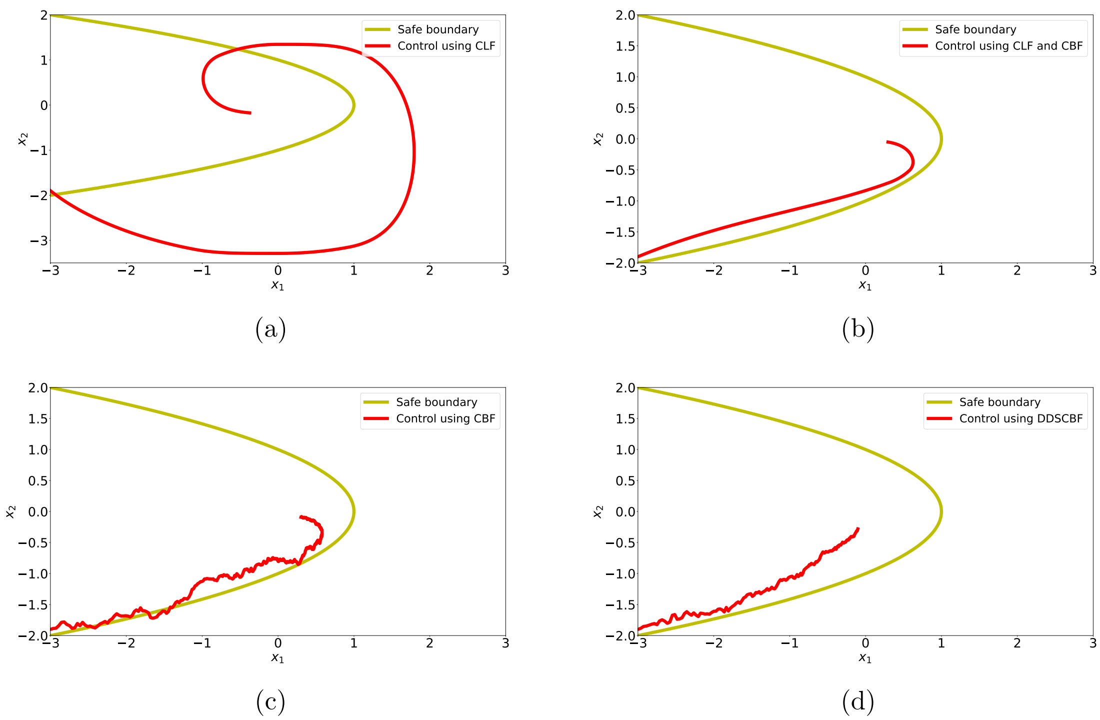


Figure 5.3. Simulation results of nonlinear system. (a): Control of system using CLF with $\sigma(x_2) = 0$. (b): Control of deterministic system using CLF and CBF with $\sigma(x_2) = 0$. (c): Sample trajectory of uncertain system using CBF with $\sigma(x_2) = 0.1x_2$. (d): Sample trajectory of uncertain system using DDSCBF scheme with $\sigma(x_2) = 0.1x_2$.

As in the first example, in order to test the performance of our DDSCBF scheme, we randomly sample 1000 trajectories and compute the safe rate under different noise for $\sigma(x_2) = 0.1x_2$ and $\sigma(x_2) = 0.15x_2$. The result is presented in Table 5.2.

5.5 Conclusion

In this chapter, we study safety-critical control problems for stochastic systems with unknown diffusion parts. We show that we can learn the infinitesimal generator of the SCBFs using data-driven method. We use supervised learning to approximate the generator of SCBFs for safety control of SDEs. We also validate our result using two nonlinear SDEs. However, one of the potential bottleneck of such method might be time-consuming for more complicated systems with higher relative degree. As a result, this bottleneck will be discussed into more detail in the following Chapter.

Chapter 6

Safety-Critical Control with Unknown Drift and Diffusion

In this chapter, we still consider safety-critical control for stochastic system driven by Gaussian noise. But unlike above chapters, we assume that we do not have either the drift term or the diffusion term of the system. We use data from observation to address safety guarantee of the system.

6.1 Preliminary and Problem Definition

6.1.1 System Description

We consider the same system as in Chapter 4 for a continuous-time stochastic process $X : [0, \infty) \times \Omega \to \mathcal{X}$ that solves the SDE

$$dX_t = (f(X_t) + g(X_t)u(t))dt + \sigma(X_t)dW_t, \tag{6.1}$$

where $u : \mathbb{R}_{\geq 0} \to \mathcal{U}$ is a bounded measurable control signal; W is a d-dimensional standard $\{\mathcal{F}\}_t$-Brownian motion; $f : \mathcal{X} \to \mathbb{R}^n$ is a nonlinear vector field; $g : \mathcal{X} \to \mathbb{R}^{n \times p}$ and $\sigma : \mathcal{X} \to \mathbb{R}^{n \times d}$ are smooth mappings.

Let X be a strong solution of the SDE defined as 4.1.2, then the infinitesimal generator of X can be defined according to (4.1.4) and calculated based on (4.4).

6.1.2 Problem Formulation

Similar as previous chapters, we consider a safe set of the form

$$\mathcal{C} := \{x \in \mathcal{X} : h(x) \geq 0\}, \tag{6.2}$$

where $h \in C^2(\mathbb{R}^n)$. We also define the boundary and interior of $\mathcal{C}$ explicitly as below

$$\partial\mathcal{C} := \{x \in \mathcal{X} : h(x) = 0\}, \tag{6.3}$$

$$\mathcal{C}^\circ := \{x \in \mathcal{X} : h(x) > 0\}. \tag{6.4}$$

The objective of this chapter is to control the stochastic system (6.1) with unknown drift and diffusion terms to stay inside the safe set. Given the definition of controlled p-invariant according to Definition 4.1.7, the problem is defined as follows.

Problem 6.1.1. *Given system (6.1) with f, g and σ unknown, a compact set $\mathcal{C} \subseteq \mathcal{X}$ defined by (6.2), a point $x \in \mathcal{C}^\circ$, and a $p \in [0,1]$, design a (deterministic) control strategy κ such that under $\mathbf{u}_\kappa$, the interior $\mathcal{C}^\circ$ is controlled p-invariant for the resulting solutions to (6.1).*

Note that in Chapter 4, we assume that we have full knowledge of the SDE and in Chapter 5, we make the assumption that the diffusion term of the SDE is unknown. In this chapter, we further extend our assumption that both the drift and diffusion terms are unknown.

6.2 Data Collection and System Identification

In Chapter 4, we define our stochastic control barrier function (SCBF) as in Definition 4.2.6 and we show that our SCBF can guarantee a worst-case safety probability according to Theorem 4.2.10. As a result, in this section, we will address the process of identifying the system with data-driven method using Bayesian inference and then, then propose our QP-based control framework for safety-critical control of the unknown SDE with learned dynamics using SCBF

6.2.1 Data Collection for Drift Term

In order to collect training data to estimate $f(x)$ and $g(x)$, we sample an initial state and let the system evolve under control inputs. More specifically, given some $T > 0$, we divide the time interval $[0, T]$ into a sufficiently refined partition $\{t_0, t_1, ..., t_N\}$ with $N \in \mathbb{N}$ such that $0 = t_0 < t_1 < \cdots < t_N = T$. Define the index set as $\mathcal{I} = \{0, 1, \ldots, N\}$. At each $i \in \mathcal{I}$, we simulate the system with K times from x_i with two control inputs $u_{i,1}$ and $u_{i,2}$ such that $u_{i,1} \neq u_{i,2}$. For each $u_{i,j}$ with $j \in \{1, 2\}$, we will get $x_{i,j}^k$ for $k \in \{1, 2, ..., K\}$. Applying CLT to get

$$dx_{i,j} = \sum_{k=1}^{K} \frac{[x_{i+1,j} - x_i | x_i]}{K}.$$

Accordingly, we can calculate the target value for $f(x_i)$ and $g(x_i)$ as

$$\begin{aligned} y_{f(x_i)} &= \frac{dx_{i,1} \cdot u_{i,2} - dx_{i,2} \cdot u_{i,2}}{(u_{i,2} - u_{i,1})(t_{i+1} - t_i)}, \\ y_{g(x_i)} &= \frac{dx_{i,1} - dx_{i,2}}{(u_{i,1} - u_{i,2})(t_{i+1} - t_i)}. \end{aligned} \tag{6.5}$$

Finally we will get training data as

$$\begin{aligned} \mathbf{X} &= [x_0, \ldots, x_i, \ldots, x_N]^T, \\ \mathbf{Y}_f &= [y_{f(x_0)}, \ldots, y_{f(x_i)}, \ldots, y_{f(x_N)}]^T, \\ \mathbf{Y}_g &= [y_{g(x_0)}, \ldots, y_{g(x_i)}, \ldots, y_{g(x_N)}]^T. \end{aligned} \tag{6.6}$$

6.2.2 Identification of Drift Term

We use Bayesian linear regression [15] to identify the drift term of the system. Define Φ as the base function and ϑ as the weight vector where $\Phi \in \mathbb{R}^{N \times M}$ and $\vartheta \in \mathbb{R}^{M \times 1}$. Here, M is the number of base function. We can write the regression equation as

$$Y = \Phi\vartheta + \varepsilon.$$

The noise term ε is assumed to follow a Gaussian distribution $\mathcal{N}(0, \Psi)$ where Ψ is a diagonal matrix with the i-th element as σ^2/K. Initially, we impose a Gaussian prior $p(\vartheta)$ with zero mean and covariance matrix Σ_0 on weight vector ϑ. According to Bayes' theorem, the posterior distribution of ϑ is

$$P(\vartheta|Y) \propto p(\vartheta)p(Y|\vartheta),$$

where $p(Y|\vartheta)$ is the likelihood function. Assuming that the error term is independently and identically distributed, the likelihood function can be written

$$P(Y|\vartheta) = \mathcal{N}(\Phi\vartheta, \sigma^2 I/K),$$

where I is the n dimensional identity matrix. Then the posterior distribution can be calculated analytically as

$$P(\vartheta|Y) = \mathcal{N}(\bar{\vartheta}, \bar{\Sigma})$$

with $\bar{\vartheta}$ the maximum a posterior estimation (MAP) as

$$\bar{\vartheta} = (\Phi^T\Phi + \frac{\sigma^2}{K}\Sigma_0^{-1})^{-1}\Phi^T Y \tag{6.7}$$

and $\bar{\Sigma}$ as the posterior covariance matrix as

$$\bar{\Sigma} = \frac{\sigma^2}{K}(\Phi^T\Phi + \frac{\sigma^2}{K}\Sigma_0^{-1})^{-1}.$$

As a result, given training data X, Y_f and Y_g as in Equation 6.6, we can approximate the drift terms $\hat{f}(x)$ and $\hat{g}(x)$. By properly choosing the base functions $\Phi_{\hat{f}}$ and $\Phi_{\hat{g}}$, we can construct models for both $\hat{f}(x)$ and $\hat{g}(x)$ according to Equation 6.2.2 as

$$\hat{f}(x) = \Phi_f\vartheta_f + \varepsilon_f,$$
$$\hat{g}(x) = \Phi_g\vartheta_g + \varepsilon_g,$$

and calculate weights $\bar{\vartheta}_f$ and $\bar{\vartheta}_g$ based on Equation 6.7.

6.2.3 Data Collection for Diffusion Term

After we estimate the drift terms, we can use the estimated $f(x)$ and $g(x)$ to identify the diffusion term. Similar as above, we also randomly sample an initial state and let the system evolve under some control u_i with $i \in \mathcal{I}$ at each time step given the refined partition $\{t_0, t_1, ..., t_N\}$. Specifically, at each time step, we calculate the ξ_i using estimated drift term as

$$\xi_i = x_{i+1} - x_i - \hat{f}(x_i)\Delta t - \hat{g}(x_i)u_i\Delta t.$$

Then we will get the dataset $\mathcal{D}_\xi$ containing $[\xi_0, \cdots, \xi_{N-1}]^T$ to estimate σ.

6.2.4 Identification of Diffusion Term

The likelihood function given dataset $\mathcal{D}$ is

$$L(\mathcal{D}|\sigma) = \Pi_{i=0}^{N-1} P(\xi_i|\sigma),$$

where

$$P(\xi_i|\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{\xi_i^2}{\sigma^2}}.$$

Given a prior distribution $P(\sigma)$, the posterior distribution of σ is

$$P(\sigma|\mathcal{D}) \propto L(\mathcal{D}|\sigma) \cdot P(\sigma). \tag{6.8}$$

Then we use the MAP method to estimate σ by maximizing the log-posterior distribution. According to Eq (6.8), we can get

$$\log(P(\sigma|\mathcal{D})) = \log L(\sigma|\mathcal{D}) + \log(P(\sigma)). \tag{6.9}$$

By using the inverse-gamma distribution

$$P(\sigma) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{1}{\sigma^{\alpha+1}} e^{-\frac{\beta}{\sigma}} \tag{6.10}$$

as the prior distribution, we can get

$$\log L(\sigma|\mathcal{D}) = \sum_{i=0}^{N-1} \log \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{\xi_i^2}{\sigma^2}},$$
$$\log P(\sigma) = \log \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{1}{\sigma^{\alpha+1}} e^{-\frac{\beta}{\sigma}}.$$

Simplifying the above equation, we have

$$\log L(\sigma|\mathcal{D}) = -(N-1)(\log\sigma + \log\sqrt{2\pi}) - \frac{1}{2\sigma^2}\sum_{i=0}^{N-1}\xi_i^2,$$
$$\log P(\sigma) = \alpha\log\beta - \log\Gamma(\alpha) - (\alpha+1)\log\sigma - \frac{\beta}{\sigma}.$$

Combining the above two expressions, we can get the log-posterior as

$$\log P(\sigma)|\mathcal{D} = -(N+\alpha)\log\sigma - \frac{1}{2\sigma^2}\sum_{i=0}^{N-1}\xi_i^2 - \frac{\beta}{\sigma} + C$$

with

$$C = \alpha \log \beta - \log \Gamma(\alpha) - (N-1) \log \sqrt{2\pi}$$

a constant. The MAP estimation of σ is obtained by taking the derivative of the log-posterior with respect to σ as

$$\frac{d \log P(\sigma|\mathcal{D})}{d\sigma} = 0. \tag{6.11}$$

Note: solving the analytic solution of Eq (6.11) is intractable in practice so we can use numerical optimization method to find out the value of σ that maximize the log-posterior distribution.

6.2.5 QP-based Control Framework for Learned Dynamics with SCBF

Based on the above identification process, we can make prediction using the learned model to approximate the dynamics of an unknown SDE. For drift terms, we can make prediction as $\hat{f}(x) = \Phi_f \bar{\vartheta}_f$ and $\hat{g}(x) = \Phi_g \bar{\vartheta}_g$ and for diffusion term, we estimate noise level using $\hat{\sigma}$ as in Equation 6.11. As a result, we can control the unknown SDE using a QP-based control framework with SCBF as

$$\begin{aligned} u(x) = \arg\min_{u \in \mathbb{R}} \frac{1}{2}||u||^2, \\ \text{s.t.} \quad \hat{\mathcal{A}}B(x) \geq 0, \end{aligned} \tag{6.12}$$

where

$$\hat{\mathcal{A}}B(x) = \frac{\partial B}{\partial x}(\hat{f}(x) + \hat{g}(x)u + \frac{1}{2}\sum_{i,j} \left(\hat{\sigma}\hat{\sigma}^T\right)_{i,j}(x)\frac{\partial^2 B}{\partial x_i \partial x_j}.$$

6.3 Simulation Result

6.3.1 Unknown Nonlinear Model with System Identification and Control

In the first example, we test our result using the following model as

$$d\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -0.6x_1 - x_2 \\ x_1^3 \end{bmatrix} dt + \begin{bmatrix} 0 \\ x_2 \end{bmatrix} udt + \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} dW.$$

The safe region is defined as

$$h(x) = -x_2^2 - x_1^2 + 1 > 0.$$

The generator of h is calculated as

$$\mathcal{A}(h) = 1.2x_1^2 + x_1x_2 - 2x_1^3x_2 - 2x_2^2u - (\sigma_2^2 + \sigma_2^2),$$

and we select

$$b_0(x) = h(x),$$
$$b_1(x) = \mathcal{A}(x).$$

In order to identify $f(x)$ in the drift term, we first randomly sample 10 initial points and at each initial point, we set $u = 0$ and simulate the system with $\Delta t = 0.01$. For any state $x_t = \xi$, we apply the control to the system at state ξ for K times to calculate the expectation of $f(x_t)$ using the central limit theorem as

$$f(\xi) + \varepsilon \approx \sum_{j=0}^{K} \frac{[x_{t_K+1} - x_t | x_t = \xi]}{K\Delta t}. \tag{6.13}$$

The we will get the training data for $f(x)$ as

$$\mathbf{X} = [\xi_1, \xi_2, ..., .\xi_N]^T,$$
$$\mathbf{Y} = [f(\xi_1), f(\xi_2), ..., f(\xi_N)].$$

and estimate $f(x)$ using linear regression. After $f(x)$ is estimated, we use the estimated function $\hat{f}(x)$ to identify $g(x)$ in the drift term. We set control input as $u = 0.1$ since the system is control affine and $g(x)$ is independent of u. We collect the training data similarly as in estimating $f(x)$. Through out this example of the simulation, we use polynomial base functions as

$$\Phi = [1, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_2^3]$$

for both $f(x)$ and $g(x)$. The result of estimation for $f(x)$ is shown in Fig 6.1. We also calculate the mean square error (MSE) for $f_1(x)$ and $f_2(x)$ over 100 randomly sample points and show the result in Table 6.1.

After we estimate the drift term, we collect data using the estimated $\hat{f}(x)$ and $\hat{g}(x)$ for 100 initial points and 300 simulation steps for each initial point. At each time step, we calculate the noise data using

$$\varphi_t = x_{t+1} - x_t - \hat{f}(x_t) \cdot dt - \hat{g}(x_t) \cdot u \cdot dt$$

Table 6.1: MSE for estimation of $f_1(x)$ and $f_2(x)$ with $K = 10, 30, 100$ over 100 randomly sample points.

	$f_1(x)$	$f_2(x)$
K=10	e^{-2}	$5e^{-2}$
K=30	$2e^{-3}$	$8e^{-3}$
K=100	$6e^{-4}$	$6e^{-4}$

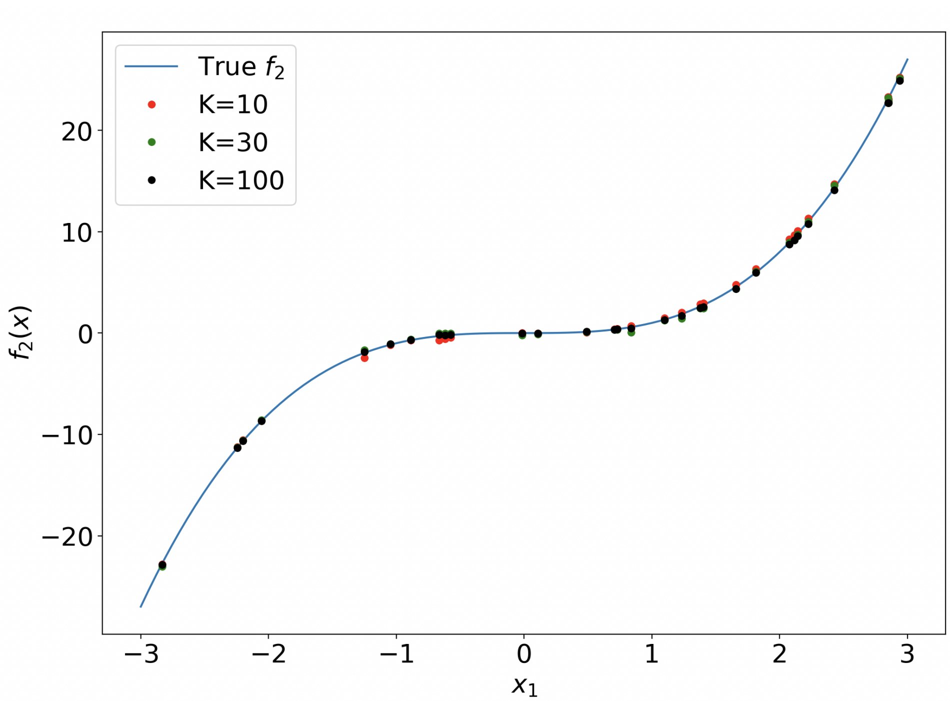


Figure 6.1. Estimation of $f_2(x) = x_1^3$ using $K = 10$, $K = 30$ and $K = 100$. The estimated result is compared with the true value of the function using 30 randomly sampled points. The diffusion of the system for all the cases is $[0.2, 0.2]$.

and add it into the data set $\mathcal{D}$. We use inverse gamma distribution as in Equation (6.10) with $\alpha = 1$ and $\beta = 1$.

We use log-likelihood function as

$$\log L(\sigma|\mathcal{D}) = \sum_{i=1}^{n} \left[-\frac{1}{2}\log(2\pi\sigma^2) - \frac{\varphi_i^2}{2\sigma^2} \right],$$

where n is the number of data in $\mathcal{D}$. We estimate a $\hat{\sigma}$ such that the log-posterior distribution

$$\log P(\hat{\sigma}|\mathcal{D}) = \log P(\sigma) + \log L(\hat{\sigma}|\mathcal{D})$$

is maximized. The distribution over 10000 random sample from posterior distribution of σ_1 and σ_2 is shown in Fig 6.2.

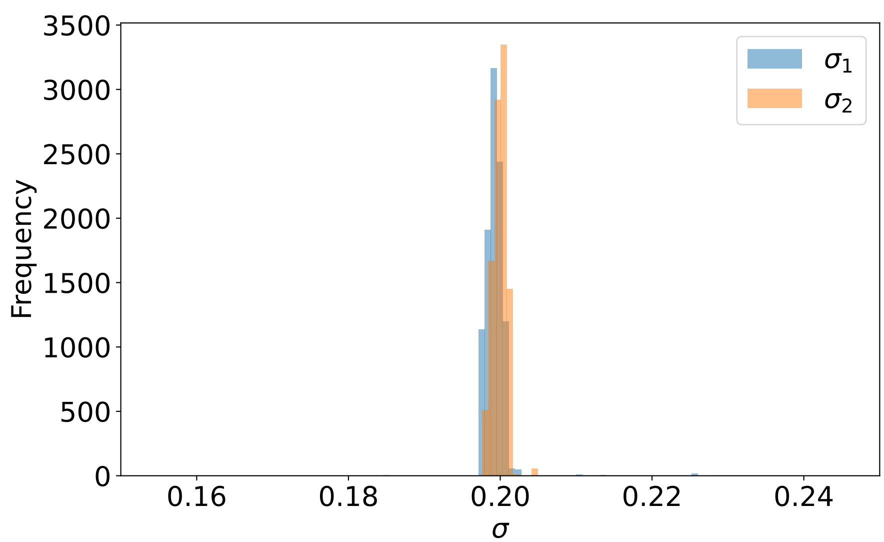


Figure 6.2. Posterior distribution of σ_1 and σ_2 over 10000 random samples for nonlinear model.

In order to analyze the performance of safety control w.r.t the estimated system, we count the number of safe trajectories over 1000 simulations under different initial points between the SCBF, DDSCBF [139] and Bayesian SCBF. For SCBF, we assume that we have knowledge of true system and for DDSCBF, we assume that we know the drift of the system and only the diffusion is unknown to us. The statistical results at initial state $[-0.1, 0.7]^T$ and $[-0.1, 0.8]^T$ are shown in Table 6.2. The analytical result in the table is the worst-case safe probability calculated based on [138, Theorem III.8]. Since the relative degree if of $r = 1$, the worst-case probability is calculated as $P = \frac{b_0(x_0)}{c_0}$ where $c_0 = \sup_{x \in \mathcal{C}} h(x)$ with $\mathcal{C}$ as the safe set. It is easy to find that $c_0 = 1$ so the worst-case probability is calculated as $P = 0.5$ and $P = 0.35$ for initial state $[-0.1, 0.7]^T$ and $[-0.1, 0.8]^T$, respectively.

Table 6.2: Safety ratio over 1000 simulation trajectories at initial state $[-0.1, 0.7]^T$ and $[-0.1, 0.8]^T$ with diffusion $\sigma_1, \sigma_2 = 0.2$.

	Safety ratio	
	$(-0.1, 0.7)$	$(-0.1, 0.8)$
SCBF [138]	91%	55%
DDSCBF [139]	88%	45%
Bayesian SCBF	90%	43%
Analytical [138]	50%	35%

6.3.2 Adaptive Cruise Control Model with System Identification and Control

In the second example, we use an adaptive cruise control (ACC) model as in Chapter 4 to validate the estimation of the system and safety control. The model is

$$d \begin{bmatrix} v \\ z \end{bmatrix} = \begin{bmatrix} -F_r(x)/M \\ v_f - v \end{bmatrix} dt + \begin{bmatrix} 1/M \\ 0 \end{bmatrix} udt + \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} dW,$$

where $x = [v, z]$ is the state of the system representing the speed of the behind vehicle and the distance between two vehicles, respectively. v_f is the speed of the front vehicle. The aerodynamic drag is $F_r(x) = f_0 + f_1 x_1 + f_2 x_1^2$ with $f_0 = 0.1,\ f_1 = 5,\ f_2 = 0.25$ and the mass of the vehicle is $M = 1650$. We require the behind vehicle to reach a desired speed $v_d = 22$ while keeping a minimum distance with $D = 10$ from the front vehicle. Since the second-order generator of $h(x)$ is

$$\mathcal{A}\mathcal{A}h(x) = L_f^2 h(x) + L_g L_f h(x) u + \Sigma^T \frac{\partial^4 h(x)}{\partial x^4} \Sigma$$

with

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix},$$

we select $h(x) = (z - D)^5$ with a non-zero fourth-order derivative. As a result, select

$$\begin{aligned}
b_0(x) &= h(x), \\
b_1(x) &= \mathcal{A}h(x) = 5(z-D)^4(v_f - v), \\
b_2(x) &= \frac{5(z-D)^4}{M} \cdot F + 20(z-D)^3 \cdot (v_f - v)^2 \\
&- \frac{(z-D)^4}{M} \cdot u + 120(z-D)(\sigma_1^2 + \sigma_2^2).
\end{aligned}$$

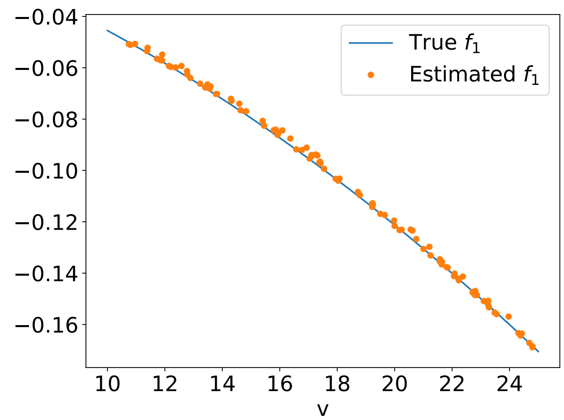
We use the same base functions, prior distribution and likelihood function as in the first example for system estimation. The result of estimation for $f(x)$ is displayed as in Fig 6.3 and the distribution over 10000 random samples from the posterior distribution of σ_1 and σ_2 is shown in Fig 6.4.

To control the vehicle to reach the desired speed, we use control Lyapunov function as in [138]. We calculate the safety ratio over 1000 simulations between SCBF, DDSCBF and Bayesian SCBF. Note that the SCBF is a non-convex function in this example, we can not find the supreme value of c_0 as in the first example. So we only compare simulated safety-probability and the result is shown in Table 6.3.

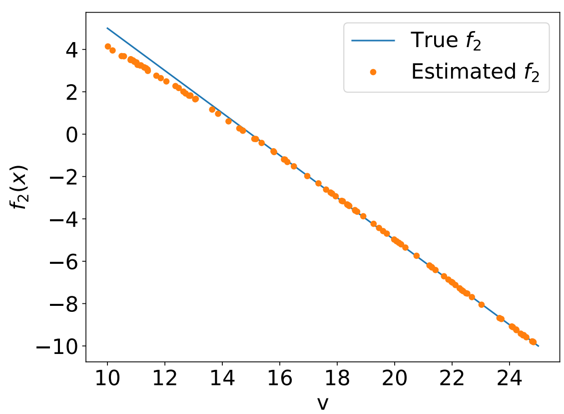
Table 6.3: Safety ratio over 1000 simulation trajectories with diffusion $\sigma_1, \sigma_2 = 0.5$. The initial state is $[v, z]^T = [10, 15]^T$.

	Safety ratio
SCBF [138]	83%
DDSCBF [139]	65%
Bayesian SCBF	78%

Also, as stated in [139], the bottleneck of the DDSCBF is that when the relative degree is higher than 1, we have to recursively apply this method in each order to calculate higher-order generators. As in this example, we have relative degree $r = 2$, we need to learn the first-order generator, and then use the estimated first-order generator to sample data for estimating the second order generator. In the process of collecting training data for second order generator using DDSCBF, we have to sample N points and at each point, sample K simulations and apply CLT to calculate training data. However, in each simulation, we need to get the value of first-order generator through network, which is very time consuming. As a result, we compare the running time between DDSCBF and Bayesian estimated SCBF. The runtime of both methods are displayed in Table 6.4. In the simulation of DDSCBF, we select $N = 300$ and $K = 10000$.



(a)



(b)

Figure 6.3. Estimation of $f(x)$ for adaptive cruise control model. The result is validated using 100 randomly sampled states. (a): Estimation of $f_1(x)$. (b): Estimation of $f_2(x)$.

Table 6.4: Runtime of learning process

	Runtime
DDSCBF [139]	over 5 hours
Bayesian SCBF	15s

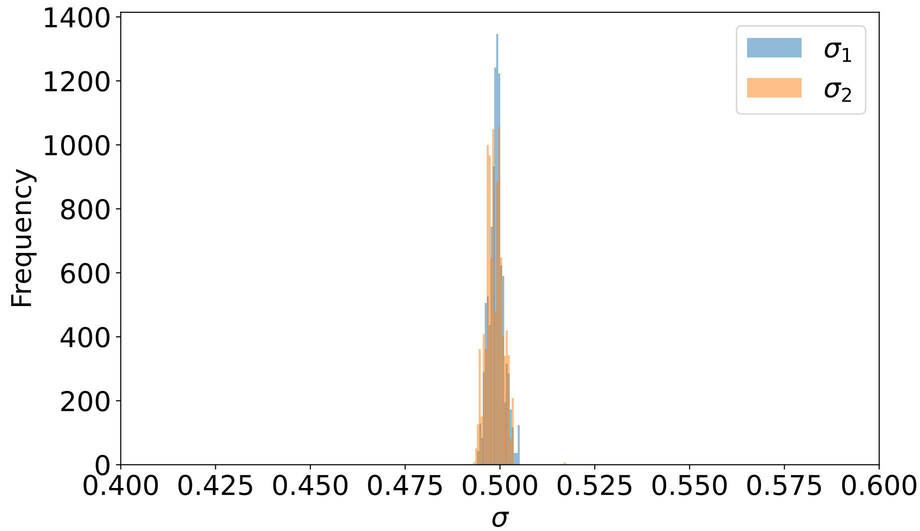


Figure 6.4. Posterior distribution of σ_1 and σ_2 over 10000 random samples for adaptive cruise control model. The true values are $\sigma_1 = 0.5$ and $\sigma_2 = 0.5$.

6.4 Conclusion

In this chapter we address safety-critical control problem for fully unknown SDEs. We use Bayesian inference to estimate both the drift and diffusion terms of the system and use SCBF to guarantee safety for the learned system. We validate safety ratio with statistical results using nonlinear models and compare with analytical safety probabilities.

Chapter 7

Conclusion and Future Work

This thesis mainly focuses on safety guarantee of systems under uncertainties. Specifically, we address deterministic systems, stochastic systems and stochastic systems with unknown parameters. Throughout the paper, we discuss different levels of uncertainty and provide different learning approaches. For deterministic systems, we use learning methods to approximate the dynamics of the higher order control barrier functions. We show that the error in the training and generalization process can be bounded by choosing parameters small enough during the process, so that the learned function is also a higher order control barrier function to guarantee safety for such deterministic systems. For stochastic systems with partial information, we can not calculate infinitesimal generator directly. As a result, we show that we could approximate the generator given collected data. We also show that by carefully selecting training parameters, the approximated generator can also guarantee a worst-case safety estimation. Finally, we discuss fully known SDEs. Due to the fact that the method in Chapter 5 is computationally demanding when applied iteratively for higher order degree cases, we propose a method to identify the system first and then control the system using the identified system. However, we do not provide further safety guarantee based on such method, and this would be an interesting direction for future work. Below is a more detailed conclusion of our work in each chapter.

7.1 Conclusion

In Chapter 4, we consider stochastic systems driven by Gaussian noise. We consider the pros and cons of the existing formulations for stochastic barrier functions (such as frequently used SRCBFs and SZCBFs), and propose stochastic control barrier functions

(SCBFs) for safety-critical control of stochastic systems and extend the worst-case safety probability estimation to higher-order SCBFs. We show that the proposed SCBFs provide good trade-offs between the imposed control constraints and the conservatism in the estimation of safety probability, which are demonstrated both theoretically and empirically. Our simulations with the automatic cruise control model and the Dubins vehicle model have underscored the practical applicability of our approach. These applications not only validate our theoretical findings but also give insights for implementing SCBFs in real-world scenarios, such as autonomous vehicle navigation and robotic control systems.

In Chapter 3, we consider safety-control for deterministic systems with unknown parameters. We propose a learning framework to show that the derivatives of CBFs can be obtained using the nominal system and observations from the simulations of the systems. We extend previous work for CBFs with relative degree one into HOCBFs. We show that under some reasonable assumptions, the learned derivatives of the HOCBF provide a robust safety guarantee for the system.

In Chapter 5, we discuss stochastic systems with unknown diffusion. We take advantage of previous work to estimate infinitesimal generators of Lyapunov functions with respect to SDEs and extend the idea to the whole state space. We show that we can use observed data to estimate the generator of SCBF when we do not have full knowledge of the stochastic systems. Additionally, we provide safety guarantee using the learned generator for such systems.

In Chapter 6, we expand our study of system uncertainty to address that both the drift and diffusion of the SDE are unknown. We propose a two-stage method by using Bayesian inference to estimate the system first and synthesize safe controllers using SCBFs. We show that the learned system can achieve similar safety ratio compared to the real systems. We also compare the runtime with the method proposed in Chapter 5, showing that identifying the system first dramatically reduces runtime, especially for cases with a higher relative degree.

7.2 Future Work

In this thesis, we discuss problems of safety-critical control under uncertainties. Although this thesis covers safety-critical control under uncertainties, including parametric and stochastic uncertainty, there remains several unexplored topics in both theory and practice, particularly for more complex systems or demanding working environments. Therefore, while we have made significant progress, the field of safety-critical control for uncertain systems still holds many opportunities for further research work. As a result, we will next propose some open problems in the field of safety-critical control for systems with uncertainty based on the discussion of this thesis.

Safety-critical Control for Unknown System with Limited Data

In Chapter 5 and 6, we identify the dynamics of the generators an the dynamics of the SDEs, respectively, in order to control stochastic systems safely. The identification relies totally on the observations from the simulation and consequently, the robustness is heavily based on the quality of observations we get. However, it is not always the case that we will get enough data to satisfy our assumptions of theoretical analysis. Also, our methods rely on sample paths that is obtained on feasible policies and sparse data cannot always guarantee the accessibility of such policies. Recent work propose learning schemes to identify stochastic systems given limited amount of data as in [159,161]. Given such motivations, it is worth studying safety-critical control for systems that are identified from limited amount of data. It is also interesting to establish the convergent rate of identification given different observation size.

Safety Guarantee of Identified Stochastic System

To the best of our knowledge, we are the first one to discuss safety-critical control of stochastic systems with completely unknown drift and diffusion as in Chapter 6. We apply our SCBFs on the identified system and compare with results of the real system. However, we haven't address the analytical robustness of the system under current learning scheme. The performance of the controllers on the identified system is validated only on numerical analysis, which would be limited. It would be interesting to study such robustness of stochastic systems under different identifying schemes, such as Gaussian process or Bayesian inferences. The study of robustness in identified stochastic systems would not only improve the theoretical understanding of these systems but also have practical implications in areas such as robotic control and autonomous driving.

More Complicated Specification in Practice

Current scope of safety control research often focuses on relatively simple objectives, such as obstacle avoidance or reaching a single target. These scenarios only require some straightforward design of CBFs. However, real-world applications frequently face complicated scenarios and requirements. This increases difficulty not only in designing CBFs, but also in synthesizing reliable controllers. Designing CBFs for such complex requirements is usually challenging, especially when multiple CBFs are necessary. Additionally, when these CBFs are integrated into Quadratic Programming (QP) problems architecture, multiple constraints can result in the situation in which the solution of the QP is infeasible. Future research could be conducted towards developing methods that enable the design of CBFs for these complicated scenarios. Furthermore, in Chapter 4, we propose our SCBFs only addressing safety guarantee of stochastic systems. However, in practice, not only do we need to provide safety, but also require the (stochastic) systems to satisfy different specifications. One of the potential direction in the future is to study reach-avoid problems combining Lyapunov analysis with our SCBFs. More complicated specifications also need to be considered such as linear temporal combining our theoretical analysis for SCBFs.

References

[1] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Ng. An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, 19, 2006.

[2] Hassan Almubarak, Manan Gandhi, Yuichiro Aoyama, Nader Sadegh, and Evangelos A Theodorou. Gaussian process barrier states for safe trajectory optimization and control. *arXiv preprint arXiv:2212.00268*, 2022.

[3] Bowen Alpern and Fred B Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 1985.

[4] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *Proc. of ECC*, pages 3420–3431. IEEE, 2019.

[5] 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.

[6] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *Proc. of CDC*, pages 6271–6278. IEEE, 2014.

[7] Aaron D Ames and Matthew Powell. Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs. In *Control of Cyber-Physical Systems: Workshop held at Johns Hopkins University, March 2013*, pages 219–240. Springer, 2013.

[8] 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.

[9] Cedric Archambeau, Dan Cornford, Manfred Opper, and John Shawe-Taylor. Gaussian process approximations of stochastic differential equations. In *Gaussian Processes in Practice*, pages 1–16. PMLR, 2007.

[10] Andrea Bacciotti and Lionel Rosier. *Liapunov functions and stability in control theory*. Springer Science & Business Media, 2005.

[11] 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.

[12] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 30, 2017.

[13] Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[14] Samir Bouabdallah and Roland Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 2247–2252. IEEE, 2005.

[15] George EP Box and George C Tiao. *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011.

[16] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[17] 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. PMLR, 2022.

[18] Zhiyuan Cai, Huanhui Cao, Wenjie Lu, Lin Zhang, and Hao Xiong. Safe multi-agent reinforcement learning through decentralized multiple control barrier functions. *arXiv preprint arXiv:2103.12553*, 2021.

[19] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

[20] Peter J Campo and Manfred Morari. Robust model predictive control. In *1987 American control conference*, pages 1021–1026. IEEE, 1987.

[21] 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 *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6762–6769. IEEE, 2021.

[22] Ismaël Castillo, Johannes Schmidt-Hieber, and Aad Van der Vaart. Bayesian linear regression with sparse priors. 2015.

[23] Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural lyapunov control. *Advances in neural information processing systems*, 32, 2019.

[24] Hong Chen and Frank Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[25] Yu Chen, Chao Shang, Xiaolin Huang, and Xiang Yin. Data-driven safe controller synthesis for deterministic systems: A posteriori method with validation tests. *arXiv preprint arXiv:2304.00729*, 2023.

[26] 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, 2017.

[27] 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 *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 777–783. IEEE, 2020.

[28] 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 *Proc. of AAAI*, volume 33, pages 3387–3395, 2019.

[29] Jason Choi, Fernando Castañeda, 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.

[30] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

[31] Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

[32] Girish Chowdhary, Hassan A Kingravi, Jonathan P How, and Patricio A Vela. Bayesian nonparametric adaptive control using gaussian processes. *IEEE Transactions on Neural Networks and Learning Systems*, 26(3):537–550, 2014.

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

[34] David W Clarke, Coorous Mohtadi, and P Simon Tuffs. Generalized predictive control—part i. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

[35] David W Clarke, Coorous Mohtadi, and P Simon Tuffs. Generalized predictive control—part ii extensions and interpretations. *Automatica*, 23(2):149–160, 1987.

[36] Max H Cohen and Calin Belta. High order robust adaptive control barrier functions and exponentially stabilizing adaptive control lyapunov functions. *arXiv preprint arXiv:2203.01999*, 2022.

[37] Ersin Daş and Richard M Murray. Robust safe control synthesis with disturbance observer-based control barrier functions. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 5566–5573. IEEE, 2022.

[38] Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods. *arXiv preprint arXiv:2202.11762*, 2022.

[39] Grace S Deaecto and José C Geromel. Stability analysis and control design of discrete-time switched affine systems. *IEEE Transactions on Automatic Control*, 62(8):4058–4065, 2016.

[40] Vikas Dhiman, Mohammad Javad Khojasteh, Massimo Franceschetti, and Nikolay Atanasov. Control barriers in bayesian learning of system dynamics. *IEEE Transactions on Automatic Control*, 2021.

[41] T Dierks and Sarangapani Jagannathan. Optimal control of affine nonlinear continuous-time systems. In *Proceedings of the 2010 American control conference*, pages 1568–1573. IEEE, 2010.

[42] Warren E. Dixon, Erkan Zergeroglu, Darren M. Dawson, and Bret T Costic. Repetitive learning control: A lyapunov-based approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 32(4):538–545, 2002.

[43] Ludvig Doeser, Petter Nilsson, Aaron D Ames, and Richard M Murray. Invariant sets for integrators and quadrotor obstacle avoidance. In *2020 American Control Conference (ACC)*, pages 3814–3821. IEEE, 2020.

[44] Dynkin. *Markov processes.* Springer, 1965.

[45] Yousef Emam, Paul Glotfelter, Zsolt Kira, and Magnus Egerstedt. Safe model-based reinforcement learning using robust control barrier functions. *arXiv preprint arXiv:2110.05415*, 2021.

[46] Mahato Endo, Tatsuya Ibuki, and Mitsuji Sampei. Collision-free formation control for quadrotor networks based on distributed quadratic programs. In *2019 American Control Conference (ACC)*, pages 3335–3340. IEEE, 2019.

[47] Peyman Mohajerin Esfahani, Debasish Chatterjee, and John Lygeros. The stochastic reach-avoid problem and set characterization for diffusions. *Automatica*, 70:43–56, 2016.

[48] 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 *Proc. of ICRA*, pages 4093–4099. IEEE, 2020.

[49] Janick V Frasch, Andrew Gray, Mario Zanon, Hans Joachim Ferreau, Sebastian Sager, Francesco Borrelli, and Moritz Diehl. An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles. In *2013 European Control Conference (ECC)*, pages 4136–4141. IEEE, 2013.

[50] Randy A Freeman and Petar V Kokotovic. Inverse optimality in robust stabilization. *SIAM journal on control and optimization*, 34(4):1365–1391, 1996.

[51] Randy A Freeman and James A Primbs. Control lyapunov functions: New ideas from an old source. In *Proceedings of 35th IEEE conference on decision and control*, volume 4, pages 3926–3931. IEEE, 1996.

[52] Radian Furqon, Ying-Jen Chen, Motoyasu Tanaka, Kazuo Tanaka, and Hua O Wang. An sos-based control lyapunov function design for polynomial fuzzy control of nonlinear systems. *IEEE Transactions on Fuzzy Systems*, 25(4):775–787, 2016.

[53] Nathan Gaby, Fumin Zhang, and Xiaojing Ye. Lyapunov-net: A deep neural network architecture for lyapunov function approximation. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2091–2096. IEEE, 2022.

[54] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[55] Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24:81–108, 2005.

[56] Iosif Ilyich Gikhman and Anatoli Vladimirovich Skorokhod. Stochastic differential equations. In *The theory of stochastic processes III*, pages 113–219. Springer, 2007.

[57] Jeremy H Gillula and Claire J Tomlin. Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor. In *2012 IEEE International Conference on Robotics and Automation*, pages 2723–2730. IEEE, 2012.

[58] Paul Glotfelter, Ian Buckley, and Magnus Egerstedt. Hybrid nonsmooth barrier functions with applications to provably safe and composable collision avoidance for robotic systems. *IEEE Robotics and Automation Letters*, 4(2):1303–1310, 2019.

[59] 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.

[60] Ruben Grandia, Andrew J Taylor, Andrew Singletary, Marco Hutter, and Aaron D Ames. Nonlinear model predictive control of robotic systems with control lyapunov functions. *arXiv preprint arXiv:2006.01229*, 2020.

[61] Lars Grüne. Computing lyapunov functions using deep neural networks. *arXiv preprint arXiv:2005.08965*, 2020.

[62] Wei He, Zhao Yin, and Changyin Sun. Adaptive neural network control of a marine vessel with constraints using the asymmetric barrier lyapunov function. *IEEE transactions on cybernetics*, 47(7):1641–1651, 2016.

[63] Matthias Heger. Consideration of risk in reinforcement learning. In *Machine Learning Proceedings 1994*, pages 105–111. Elsevier, 1994.

[64] Shao-Chen Hsu, Xiangru Xu, and Aaron D Ames. Control barrier function based quadratic programs with application to bipedal robotic walking. In *Proc. of ACC*, pages 4542–4548. IEEE, 2015.

[65] John Jackson, Luca Laurenti, Eric Frew, and Morteza Lahijanian. Safety verification of unknown dynamical systems via gaussian process regression. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 860–866. IEEE, 2020.

[66] Pushpak Jagtap, George J Pappas, and Majid Zamani. Control barrier functions for unknown nonlinear systems using gaussian processes. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3699–3704. IEEE, 2020.

[67] R Kashyap. Maximum likelihood identification of stochastic linear systems. *IEEE Transactions on Automatic Control*, 15(1):25–34, 1970.

[68] S Sathiya Keerthi and Elmer G Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of optimization theory and applications*, 57:265–293, 1988.

[69] Hassan K Khalil. Lyapunov stability. *Control systems, robotics and automation*, 12:115, 2009.

[70] Hassan K Khalil. *Nonlinear control*, volume 406. Pearson New York, 2015.

[71] Hassan K Khalil and Jessy W Grizzle. *Nonlinear Systems*, volume 3. Prentice Hall, 2002.

[72] Mouhyemen Khan, Tatsuya Ibuki, and Abhijit Chatterjee. Gaussian control barrier functions: A non-parametric paradigm to safety. *arXiv preprint arXiv:2203.15474*, 2022.

[73] Mouhyemen Khan, Munzir Zafar, and Abhijit Chatterjee. Barrier functions in cascaded controller: Safe quadrotor control. In *2020 American Control Conference (ACC)*, pages 1737–1742. IEEE, 2020.

[74] S Mohammad Khansari-Zadeh and Aude Billard. Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6):752–765, 2014.

[75] 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*, pages 781–792. PMLR, 2020.

[76] Michał Kisielewicz. Stochastic differential inclusions and diffusion processes. *Journal of mathematical analysis and applications*, 334(2):1039–1054, 2007.

[77] Michał Kisielewicz et al. *Stochastic differential inclusions and applications*. Springer, 2013.

[78] Shumon Koga and Miroslav Krstic. Safe pde backstepping qp control with high relative degree cbfs: Stefan model with actuator dynamics. *IEEE Transactions on Automatic Control*, 2023.

[79] Yogesh Kumar, SB Roy, and PB Sujit. Barrier lyapunov function based trajectory tracking controller for autonomous vehicles with guaranteed safety bounds. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 722–728. IEEE, 2020.

[80] Harold J Kushner. Stochastic stability and control. Technical report, Brown Univ Providence RI, 1967.

[81] Joseph P La Salle. *The stability of dynamical systems*. SIAM, 1976.

[82] Vangipuram Lakshmikantham and Srinivasa Leela. *Differential and Integral Inequalities: Theory and Applications: Volume I: Ordinary Differential Equations*. Academic Press, 1969.

[83] Leslie Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, (2):125–143, 1977.

[84] Steven M LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[85] Jay H Lee, Manfred Morari, and Carlos E Garcia. State-space interpretation of model predictive control. *Automatica*, 30(4):707–717, 1994.

[86] José María Manzano, Daniel Limon, David Muñoz de la Peña, and Jan-Peter Calliess. Robust learning-based mpc for nonlinear constrained systems. *Automatica*, 117:108948, 2020.

[87] Zahra Marvi and Bahare Kiumarsi. Safe reinforcement learning: A control barrier function optimization approach. *International Journal of Robust and Nonlinear Control*, 31(6):1923–1940, 2021.

[88] David Q Mayne and Hannah Michalska. Receding horizon control of nonlinear systems. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 464–465. IEEE, 1988.

[89] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[90] Rayan Mazouz, Karan Muvvala, Akash Ratheesh Babu, Luca Laurenti, and Morteza Lahijanian. Safety guarantees for neural network dynamic systems via stochastic barrier functions. *Advances in Neural Information Processing Systems*, 35:9672–9686, 2022.

[91] Yiming Meng and Jun Liu. Sufficient conditions for robust probabilistic reach-avoid-stay specifications using stochastic lyapunov-barrier functions. In *2022 American Control Conference (ACC)*, pages 2283–2288. IEEE, 2022.

[92] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[93] Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. *arXiv preprint arXiv:1205.4810*, 2012.

[94] Tamas G Molnar, Ryan K Cosner, Andrew W Singletary, Wyatt Ubellacker, and Aaron D Ames. Model-free safety-critical control for robotic systems. *IEEE robotics and automation letters*, 7(2):944–951, 2021.

[95] Manfred Morari, Carlos E Garcia, and David M Prett. Model predictive control: theory and practice. *IFAC Proceedings Volumes*, 21(4):1–12, 1988.

[96] Muhammad Ali Murtaza, Sergio Aguilera, Muhammad Waqas, and Seth Hutchinson. Safety compliant control for robotic manipulator with task and input constraints. *IEEE Robotics and Automation Letters*, 7(4):10659–10664, 2022.

[97] Nami Nakamura, Hisakazu Nakamura, Yuh Yamashita, and Hirokazu Nishitani. Homogeneous stabilization for input affine homogeneous systems. *IEEE Transactions on Automatic Control*, 54(9):2271–2275, 2009.

[98] Ameneh Nejati, Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Data-driven estimation of infinitesimal generators of stochastic systems. *IFAC-PapersOnLine*, 54(5):277–282, 2021.

[99] Ameneh Nejati, Sadegh Soudjani, and Majid Zamani. Compositional construction of control barrier functions for networks of continuous-time stochastic systems. *IFAC-PapersOnLine*, 53(2):1856–1861, 2020.

[100] Ameneh Nejati, Sadegh Soudjani, and Majid Zamani. Compositional construction of control barrier functions for continuous-time stochastic hybrid systems. *Automatica*, 145:110513, 2022.

[101] Quan Nguyen and Koushil Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *Proc. of ACC*, pages 322–328. IEEE, 2016.

[102] Gennaro Notomista, Sebastian F Ruf, and Magnus Egerstedt. Persistification of robotic tasks using control barrier functions. *IEEE Robotics and Automation Letters*, 3(2):758–763, 2018.

[103] Petter Ogren, Magnus Egerstedt, and Xiaoming Hu. A control lyapunov function approach to multi-agent coordination. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, volume 2, pages 1150–1155. IEEE, 2001.

[104] Abhishek Goud Pandala, Yanran Ding, and Hae-Won Park. qpswift: A real-time sparse quadratic program solver for robotic applications. *IEEE Robotics and Automation Letters*, 4(4):3355–3362, 2019.

[105] Marcus Pereira, Ziyi Wang, Ioannis Exarchos, and Evangelos Theodorou. Safe optimal control using stochastic barrier functions and deep forward-backward sdes. In *Conference on Robot Learning*, pages 1783–1801. PMLR, 2021.

[106] Theodore J Perkins and Andrew G Barto. Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3(Dec):803–832, 2002.

[107] Stephen Prajna, Ali Jadbabaie, and George J Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428, 2007.

[108] S Joe Qin and Thomas A Badgwell. An overview of industrial model predictive control technology. In *AIche symposium series*, volume 93, pages 232–256. New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997.

[109] Manuel Rauscher, Melanie Kimmel, and Sandra Hirche. Constrained robot control using control barrier functions. In *Proc. of IROS*, pages 279–285. IEEE, 2016.

[110] Beibei Ren, Qing-Chang Zhong, and Jinhao Chen. Robust control for a class of nonaffine nonlinear systems based on the uncertainty and disturbance estimator. *IEEE Transactions on Industrial electronics*, 62(9):5881–5888, 2015.

[111] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. Uniting control lyapunov and control barrier functions. In *53rd IEEE Conference on Decision and Control*, pages 2293–2298. IEEE, 2014.

[112] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. Stabilization with guaranteed safety using control lyapunov–barrier function. *Automatica*, 66:39–47, 2016.

[113] Ugo Rosolia and Aaron D Ames. Multi-rate control design leveraging control barrier functions and model predictive control policies. *IEEE Control Systems Letters*, 5(3):1007–1012, 2020.

[114] Ugo Rosolia, Stijn De Bruyne, and Andrew G Alleyne. Autonomous vehicle control: A nonconvex approach for obstacle avoidance. *IEEE Transactions on Control Systems Technology*, 25(2):469–484, 2016.

[115] Ugo Rosolia, Andrew Singletary, and Aaron D Ames. Unified multirate control: From low-level actuation to high-level planning. *IEEE Transactions on Automatic Control*, 67(12):6627–6640, 2022.

[116] Ali Salamati, Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Data-driven verification and synthesis of stochastic systems through barrier certificates. *arXiv preprint arXiv:2111.10330*, 2021.

[117] Ali Salamati and Majid Zamani. Safety verification of stochastic systems: A repetitive scenario approach. *IEEE Control Systems Letters*, 7:448–453, 2022.

[118] Cesar Santoyo, Maxence Dutreix, and Samuel Coogan. A barrier function approach to finite-time stochastic system verification and control. *Automatica*, 125:109439, 2021.

[119] Meenakshi Sarkar, Debasish Ghose, and Evangelos A Theodorou. High-relative degree stochastic control lyapunov and barrier functions. *arXiv preprint arXiv:2004.03856*, 2020.

[120] Simo Särkkä et al. *Recursive Bayesian inference on stochastic differential equations.* Helsinki University of Technology, 2006.

[121] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proc. of ICML*, pages 1889–1897, 2015.

[122] Katrine Seel, Esten I Grøtli, Signe Moe, Jan T Gravdahl, and Kristin Y Pettersen. Neural network-based model predictive control with input-to-state stability. In *2021 American Control Conference (ACC)*, pages 3556–3563. IEEE, 2021.

[123] Joohwan Seo, Joonho Lee, Eunkyu Baek, Roberto Horowitz, and Jongeun Choi. Safety-critical control with nonaffine control inputs via a relaxed control barrier function for an autonomous vehicle. *IEEE Robotics and Automation Letters*, 7(2):1944–1951, 2022.

[124] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[125] Andrew Singletary, Karl Klingebiel, Joseph Bourne, Andrew Browning, Phil Tokumaru, and Aaron Ames. Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8129–8136. IEEE, 2021.

[126] Tong Duy Son and Quan Nguyen. Safety-critical control for non-affine nonlinear systems with application on autonomous vehicle. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7623–7628. IEEE, 2019.

[127] Thomas M Stout and Theodore J Williams. Pioneering work in the field of computer process control. *IEEE Annals of the History of Computing*, 17(1):6–18, 1995.

[128] Rin Takano and Masaki Yamakita. Robust constrained stabilization control using control lyapunov and control barrier function in the presence of measurement noises. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pages 300–305. IEEE, 2018.

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

[130] Andrew J Taylor and Aaron D Ames. Adaptive safety with control barrier functions. In *Proc. of ACC*, pages 1399–1405. IEEE, 2020.

[131] Ben Tearle, Kim P Wabersich, Andrea Carron, and Melanie N Zeilinger. A predictive safety filter for learning-based racing control. *IEEE Robotics and Automation Letters*, 6(4):7635–7642, 2021.

[132] Keng Peng Tee, Shuzhi Sam Ge, and Eng Hock Tay. Barrier lyapunov functions for the control of output-constrained nonlinear systems. *Automatica*, 45(4):918–927, 2009.

[133] Valerio Turri, Ashwin Carvalho, Hongtei Eric Tseng, Karl Henrik Johansson, and Francesco Borrelli. Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In *16th international IEEE conference on intelligent transportation systems (ITSC 2013)*, pages 378–383. IEEE, 2013.

[134] Kim P Wabersich and Melanie N Zeilinger. Safe exploration of nonlinear dynamical systems: A predictive safety filter for reinforcement learning. *arXiv preprint arXiv:1812.05506*, 2018.

[135] Rameez Wajid, Asad Ullah Awan, and Majid Zamani. Formal synthesis of safety controllers for unknown stochastic control systems using gaussian process learning. In *Learning for Dynamics and Control Conference*, pages 624–636. PMLR, 2022.

[136] Chenfei Wang, Mahroo Bahreinian, and Roberto Tron. Chance constraint robust control with control barrier functions. In *2021 American Control Conference (ACC)*, pages 2315–2322. IEEE, 2021.

[137] 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 *2021 European Control Conference (ECC)*, pages 1459–1464. IEEE, 2021.

[138] Chuanzheng Wang, Yiming Meng, Stephen L Smith, and Jun Liu. Safety-critical control of stochastic systems using stochastic control barrier functions. In *2021 IEEE Conference on Decision and Control (CDC)*, pages 5924–5931, 2021.

[139] Chuanzheng Wang, Yiming Meng, Stephen L Smith, and Jun Liu. Data-driven learning of safety-critical control with stochastic control barrier functions. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 5309–5315. IEEE, 2022.

[140] Haijing Wang, Jinzhu Peng, Fangfang Zhang, Hui Zhang, and Yaonan Wang. High-order control barrier functions-based impedance control of a robotic manipulator with time-varying output constraints. *ISA transactions*, 129:361–369, 2022.

[141] Li Wang, Aaron D Ames, and Magnus Egerstedt. Safe certificate-based maneuvers for teams of quadrotors using differential flatness. In *Proc. of ICRA*, pages 3293–3298. IEEE, 2017.

[142] Li Wang, Evangelos A Theodorou, and Magnus Egerstedt. Safe learning of quadrotor dynamics using barrier certificates. In *Proc. of ICRA*, pages 2460–2465. IEEE, 2018.

[143] Qiang Wang, Dachuan Li, and Joseph Sifakis. Safe and efficient collision avoidance control for autonomous vehicles. In *2020 18th ACM-IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE)*, pages 1–6. IEEE, 2020.

[144] Tyler Westenbroek, David Fridovich-Keil, Eric Mazumdar, Shreyas Arora, Valmik Prabhu, S Shankar Sastry, and Claire J Tomlin. Feedback linearization for unknown systems via reinforcement learning. *arXiv preprint arXiv:1910.13272*, 2019.

[145] Guofan Wu and Koushil Sreenath. Safety-critical and constrained geometric control synthesis using control lyapunov and control barrier functions for systems evolving on manifolds. In *Proc. of ACC*, pages 2038–2044. IEEE, 2015.

[146] Guofan Wu and Koushil Sreenath. Safety-critical control of a 3d quadrotor with range-limited sensing. In *Dynamic Systems and Control Conference*, volume 50695, page V001T05A006. American Society of Mechanical Engineers, 2016.

[147] Wei Xiao and Calin Belta. Control barrier functions for systems with high relative degree. In *Proc. of CDC*, pages 474–479. IEEE, 2019.

[148] Wei Xiao, Calin Belta, and Christos G Cassandras. Event-triggered safety-critical control for systems with unknown dynamics. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 540–545. IEEE, 2021.

[149] Yuhan Xiong, Di-Hua Zhai, Mahdi Tavakoli, and Yuanqing Xia. Discrete-time control barrier function: High-order case and adaptive case. *IEEE Transactions on Cybernetics*, 2022.

[150] Xiangru Xu. Constrained control of input–output linearizable systems using control sharing barrier functions. *Automatica*, 87:195–201, 2018.

[151] Bai Xue, Naijun Zhan, and Martin Fränzle. Reach-avoid analysis for stochastic differential equations. *arXiv preprint arXiv:2208.10752*, 2022.

[152] Shakiba Yaghoubi, Georgios Fainekos, and Sriram Sankaranarayanan. Training neural network controllers using control barrier functions in the presence of disturbances. *arXiv preprint arXiv:2001.08088*, 2020.

[153] Shakiba Yaghoubi, Keyvan Majd, Georgios Fainekos, Tomoya Yamaguchi, Danil Prokhorov, and Bardh Hoxha. Risk-bounded control using stochastic barrier functions. *IEEE Control Systems Letters*, 5(5):1831–1836, 2020.

[154] Fan Yang, Jiangping Hu, Likun Wang, and Bijoy K Ghosh. Safety-critical control of nonlinear systems via new exponential control barrier functions. In *2022 41st Chinese Control Conference (CCC)*, pages 504–509. IEEE, 2022.

[155] Yongsoon Yoon, Jongho Shin, H Jin Kim, Yongwoon Park, and Shankar Sastry. Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. *Control Engineering Practice*, 17(7):741–750, 2009.

[156] Mario Zanon and Sébastien Gros. Safe reinforcement learning using robust mpc. *IEEE Transactions on Automatic Control*, 66(8):3638–3652, 2020.

[157] Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29(3):972–983, 2020.

[158] Weiye Zhao, Tairan He, and Changliu Liu. Model-free safe control for zero-violation reinforcement learning. In *5th Annual Conference on Robot Learning*, 2021.

[159] Wenxiao Zhao, George Yin, and Er-Wei Bai. Sparse system identification for stochastic systems with general observation sequences. *Automatica*, 121:109162, 2020.

[160] Dongliang Zheng, Hesheng Wang, Jingchuan Wang, Xiufeng Zhang, and Weidong Chen. Toward visibility guaranteed visual servoing control of quadrotor uavs. *IEEE/ASME Transactions on Mechatronics*, 24(3):1087–1095, 2019.

[161] Hongpeng Zhou, Chahine Ibrahim, Wei Xing Zheng, and Wei Pan. Sparse bayesian deep learning for dynamic system identification. *Automatica*, 144:110489, 2022.