

© 2021 Arun Lakshmanan

SAFE PLANNING AND CONTROL VIA  $\mathcal{L}_1$ -ADAPTATION AND CONTRACTION THEORY

BY

ARUN LAKSHMANAN

DISSERTATION

Submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy in Mechanical Engineering  
in the Graduate College of the  
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Naira Hovakimyan, Chair

Professor Srinivasa Salapaka

Professor Dusan Stipanović

Professor Petros Voulgaris, University of Nevada, Reno

Professor Evangelos Theodorou, Georgia Institute of Technology

# Abstract

Autonomous robots that are capable of operating safely in the presence of imperfect model knowledge or external disturbances are vital in safety-critical applications. The research presented in this dissertation aims to enable safe planning and control for nonlinear systems with uncertainties using robust adaptive control theory. To this end we develop methods that (i) certify the collision-risk for the planned trajectories of autonomous robots, (ii) ensure guaranteed tracking performance in the presence of uncertainties, and (iii) learn the uncertainties in the model without sacrificing the transient performance guarantees, and (iv) learn incremental stability certificates parameterized as neural networks.

In motion planning problems for autonomous robots, such as self-driving cars, the robot must ensure that its planned path is not in close proximity to obstacles in the environment. However, the problem of evaluating the proximity is generally non-convex and serves as a significant computational bottleneck for motion planning algorithms. In this work, we present methods for a general class of absolutely continuous parametric curves to compute: the minimum separating distance, tolerance verification, and collision detection with respect to obstacles in the environment.

A planning algorithm is incomplete if the robot is unable to safely track the planned trajectory. We introduce a feedback motion planning approach using contraction theory-based  $\mathcal{L}_1$ -adaptive ( $\mathcal{CL}_1$ ) control to certify that planned trajectories of nonlinear systems with matched uncertainties are tracked with desired performance requirements. We present a planner-agnostic framework to design and certify invariant tubes around planned trajectories that the robot is always guaranteed to remain inside. By leveraging recent results in contraction analysis and  $\mathcal{L}_1$ -adaptive control we present an architecture that induces invariant tubes for nonlinear systems with state and time-varying uncertainties.

Uncertainties caused by large modeling errors will significantly hinder the performance of any autonomous system. We adapt the  $\mathcal{CL}_1$  framework to safely learn the uncertainties while simultaneously providing high-probability bounds on the tracking behavior. Any available data is incorporated into Gaussian process (GP) models of the uncertainties while the error in the learned model is quantified and handled by the  $\mathcal{CL}_1$  controller to ensure that control objectives are met safely. As learning improves, so does the overall tracking performance of the system. This way,

the safe operation of the system is always guaranteed, even during the learning transients.

The tracking performance guarantees for nonlinear systems rely on the existence of incremental stability certificates that are prohibitively difficult to search for. We leverage the function approximation capabilities of deep neural networks for learning the certificates and the associated control policies jointly. The incremental stability properties of the closed-loop system are verified using interval arithmetic. The domain of the system is iteratively refined into a collection of intervals that certify the satisfaction of the stability properties over the interval regions. Thus, we avoid entirely rejecting the learned certificates and control policies just because they violate the stability properties in certain parts of the domain. We provide numerical experimentation on an inverted pendulum to validate our proposed methodology.

*Dedicated to Mridula, for everything.*

# Acknowledgments

This dissertation would not have been completed without help from several people along the way.

Most of all, I would like to thank my adviser Prof. Naira Hovakimyan for her immense support throughout my academic life. When I joined the lab in 2014 as a masters' student, I never thought I would stay on for five more years and complete my PhD. But I am glad that I did and it would not have been possible without her constant guidance and counsel.

I would like to express gratitude to the members of my committee Prof. Srinivasa Salapka, Prof. Dusan Stipanović, Prof. Petros Voulgaris, and Prof. Evangelos Theodorou for taking the time to serve on my committee and their valuable feedback as I prepared this dissertation.

I would also like to thank all the lab members, past and present, for their support and enriching discussions: Aditya, Gabriel, Chris, Zhuohuan, Lin, Mitchell, Alex, Kasey, Javier, Andrew, Thiago, Venanzio, Hamid, Bilal, Hanmin, Donglei, Hyung Jin, Pan, Hunmin, Neng, Yikun, Wenbin, Tao, and Yuliang. In particular, I want to thank Aditya for closely collaborating with me on so many research directions, and to Gabriel, Chris, and Mitchell for being such great roommates and friends during my time at the university.

I am deeply indebted to my parents for their unconditional love and support throughout the years. They have always encouraged me to pursue my own directions and I will forever be grateful for that.

Last but certainly not least, I thank my wife Mridula, for her unwavering support and believing in me even when I didn't want to believe in myself.

# Table of Contents

Notations and Symbols . . . . .	viii
Abbreviations . . . . .	x
Chapter 1 Introduction . . . . .	1
1.1 Motivation . . . . .	1
1.2 General Problem Formulation and Contributions . . . . .	3
1.3 General Framework . . . . .	6
1.4 Thesis Overview . . . . .	8
Chapter 2 Fast Proximity Queries for Safe Motion Planning . . . . .	10
2.1 Introduction . . . . .	10
2.2 Problem Formulation . . . . .	12
2.3 Bounding Methods and Analysis . . . . .	13
2.4 Proximity Queries . . . . .	18
2.5 Numerical Results . . . . .	23
Chapter 3 Safe Feedback Motion Planning for Uncertain Systems . . . . .	28
3.1 Introduction . . . . .	28
3.2 Problem Formulation . . . . .	30
3.3 Preliminaries on Contraction Theory . . . . .	33
3.4 $\mathcal{CL}_1$ -Adaptive Control . . . . .	35
3.5 Performance Analysis . . . . .	39
3.6 Numerical Results . . . . .	41
Chapter 4 Safe Simultaneous Learning and Control . . . . .	45
4.1 Introduction . . . . .	45
4.2 Problem Formulation . . . . .	47
4.3 Preliminaries on Gaussian Process Regression . . . . .	49
4.4 $\mathcal{CL}_1$ Control with Gaussian Process Learning . . . . .	50
4.5 Numerical Results . . . . .	53
Chapter 5 Certifying Feedback Policies using Interval Analysis . . . . .	56
5.1 Introduction . . . . .	56
5.2 Problem Formulation . . . . .	58
5.3 Preliminaries on Interval Arithmetic . . . . .	61

5.4	Certifying Incremental Lyapunov Functions . . . . .	62
5.5	Certifying Control Contraction Metrics . . . . .	66
5.6	Robustness to Disturbances . . . . .	68
5.7	Numerical Examples . . . . .	69
Chapter 6	Conclusion . . . . .	74
6.1	Future Work . . . . .	75
Appendix A	Proofs of Chapter 2 . . . . .	77
Appendix B	Proofs of Chapter 3 . . . . .	80
B.1	Technical Results . . . . .	80
B.2	Main Results . . . . .	91
References	. . . . .	98



# Notations and Symbols

$\mathbb{N}$	Set of natural numbers
$\mathbb{R}^n$	Real coordinate space of dimension $n \in \mathbb{N}$
$\mathbb{C}^n$	Complex coordinate space of dimension $n \in \mathbb{N}$
$\mathbb{R}_{\geq 0}$	Set of non-negative reals
$\mathbb{R}^{n \times m}$	Set of real matrices with $n \in \mathbb{N}$ rows and $m \in \mathbb{N}$ columns
$\mathbb{S}^n$	Set of real symmetric matrices with $n \in \mathbb{N}$ rows and columns
$\mathbb{S}_{\succeq 0}^n$	Set of positive semidefinite symmetric matrices with $n \in \mathbb{N}$ rows and columns
$\mathbb{S}_{\succ 0}^n$	Set of positive definite symmetric matrices with $n \in \mathbb{N}$ rows and columns
$\mathbb{I}^n$	Identity matrix with $n \in \mathbb{N}$ rows and columns
$\mathbf{1}^n$	Column vector of ones with dimension $n \in \mathbb{N}$
$\cdot^\top$	Transpose operation of a vector or a matrix
$[A]_{\mathbb{S}}$	Symmetric component of matrix $A \in \mathbb{R}^{n \times n}$ given by $(A + A^\top)$
$\bar{\lambda}(A)$	Largest eigenvalue of square matrix $A \in \mathbb{R}^{n \times n}$
$\underline{\lambda}(A)$	Smallest eigenvalue of square matrix $A \in \mathbb{R}^{n \times n}$
$\underline{\sigma}_{>0}(A)$	Smallest non-zero singular value of matrix $A \in \mathbb{R}^{n \times m}$
$\mathcal{A} \oplus \mathcal{B}$	Minkowski addition between two compact sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$
$\mathcal{A} \ominus \mathcal{B}$	Minkowski difference between two compact sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$
$ \cdot $	Absolute value of real scalar
$\ \cdot\ $	2-norm of a vectors or a matrix norm induced by vector 2-norm
$\mathcal{L}_\infty(\mathcal{S})$	Space of functions $f : \mathcal{S} \subseteq \mathbb{R} \rightarrow \mathbb{R}^n$ which satisfy $\sup_{y \in \mathcal{S}} \ f(y)\  < \infty$ .
$\ f\ _{\mathcal{L}_\infty}$	$\mathcal{L}_\infty$ norm of a function $\mathcal{L}_\infty(\mathbb{R}) \ni f : \mathbb{R} \rightarrow \mathbb{R}^n$ given by $\sup_{t \geq 0} \ f(t)\ $

$\ f\ _{\mathcal{L}_\infty}^{[0,\tau]}$	Truncated $\mathcal{L}_\infty$ norm of a function $\mathcal{L}_\infty([0, \tau]) \ni f : \mathbb{R} \rightarrow \mathbb{R}^n$ given by $\sup_{t \in [0, \tau]} \ f(t)\ $
$\mathcal{L}_1(\mathcal{S})$	Space of functions $f : \mathcal{S} \subseteq \mathbb{R} \rightarrow \mathbb{R}^n$ which satisfy $\int_{\mathcal{S}} \ f(y)\  dy < \infty$ .
$\ f\ _{\mathcal{L}_1}$	$\mathcal{L}_1$ norm of a function $\mathcal{L}_1(\mathbb{R}) \ni f : \mathbb{R} \rightarrow \mathbb{R}^n$ given by $\int_0^\infty \ f(t)\  dt$
$\ f\ _{\mathcal{L}_1}^{[0,\tau]}$	Truncated $\mathcal{L}_1$ norm of a function $\mathcal{L}_1([0, \tau]) \ni f : \mathbb{R} \rightarrow \mathbb{R}^n$ given by $\int_0^\tau \ f(t)\  dt$
$\mathcal{C}^k$	Space of functions with $k$ continuous derivatives
$f_i(y)$	$i^{\text{th}}$ element of a function $f : \mathbb{R}^k \rightarrow \mathbb{R}^n$
$A_i(y)$	$i^{\text{th}}$ column of a matrix-valued function $A(y) : \mathbb{R}^k \rightarrow \mathbb{R}^{n \times m}$
$\mathcal{L}[f(t)]$	Laplace transform of a function $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$
$\mathcal{L}^{-1}[F(s)]$	Inverse Laplace transform of a function $F : \mathbb{C} \rightarrow \mathbb{C}$
$\frac{df}{dy}, \nabla f$	Total derivative of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with respect to $y \in \mathbb{R}^n$
$\frac{\partial f}{\partial y}, \nabla_y f$	Partial derivative of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with respect to $y \in \mathbb{R}^k$
$\partial_f A(y)$	Directional derivative of a matrix valued function $A : \mathbb{R}^k \rightarrow \mathbb{R}^{n \times m}$ with respect to $y$ along the direction specified by $f : \mathbb{R}^k \rightarrow \mathbb{R}^k$ given by $\sum_{i=1}^k (\partial A(y)/\partial(y)) f_i(y)$ .
$L_f g(y)$	Lie derivative of a scalar valued function $g : \mathbb{R}^n \rightarrow \mathbb{R}$ with respect to the vector field specified by $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ given by $\nabla g(y)^\top f(y)$ .
$[a, b]$	An interval set defined as $\{x \in \mathbb{R} : a \leq x \leq b\}$ . Matrix and vector interval are denoted in the same manner. The distinction should be clear from context.

# Abbreviations

**ALTRO** Augmented Lagrangian trajectory optimizer

**BIT\*** Batch informed trees

**CBF** Control barrier function

**CCM** Control contraction metric

**CLF** Control Lyapunov function

**DNN** Deep neural network

**GP** Gaussian process

**IAS** Incremental asymptotic stability

**IES** Incremental exponential stability

**ILF** Incremental Lyapunov function

**IROA** Incremental region of attraction

**LF** Lyapunov function

**LMI** Linear matrix inequality

**MBRL** Model-based reinforcement learning

**ML** Machine learning

**MPC** Model predictive control

**ODE** Ordinary differential equation

**ReLU** Rectified linear unit

**RKHS** Reproducing kernel Hilbert space

**RL** Reinforcement learning

**ROA** Region of attraction

**RPI** Robust positively invariant set

**SMT** Satisfiability modulo theories

**SOS** Sum-of-squares

**UUB** Uniform ultimate bound

# Chapter 1

## Introduction

### 1.1 Motivation

Planning and control are among the most critical components of an autonomous system’s operational capabilities. Self-driving vehicles [1], aerial drones [2], surgical robots [3], warehouse robots [4], humanoid robots [5], rely on certified planning and control policies to ensure safe behavior in real world environments, Fig. 1.1. Moreover, since many of these robots also operate near or alongside humans, the planning and control modules must adhere to safety-critical design principles in order to prevent injury or loss of lives.

Uncertainties that are either a result of external factors or simply modeling errors are very common sources for malfunctions. Collisions with obstacles due to dynamically changing environments that are inconsistent with the planned trajectory also pose a significant risk to operation of autonomous systems. Other sources for malfunctions may also cause operational failures such as noisy sensor measurements, time-delays between subsystems, failure of actuators or computer hardware, or adversarial human factors. Additionally, the limited on-board computational power on many of these systems restrict the complexity of the safety critical algorithms that are deployed.

Real world environments are often rapidly changing. For instance, a self-driving car operating on busy roads would have to plan safe maneuvers that not only avoid collisions with nearby vehicles but also stay clear of pedestrians and bicyclists who share the road. In these situations, the motion plans must be periodically checked for collisions with obstacles in the environment and re-planned if necessary. Moreover, since sensing information is usually noisy, collisions must be checked with a margin for error that accounts for the sensing inaccuracies. However, collision checking algorithms pose significant computational bottlenecks [6] in motion planning algorithms and require fast operation times to be viable in real world scenarios.

Environmental uncertainties appear in the form of wind disturbances for aerial drones [9], imperfect road conditions for ground robots [10], and uneven surfaces for legged robots [11]. Modeling errors on the other hand appear as a result of imprecise model parameters, incorrect assumptions on the model order, or a poor fit during system identification. The typical approaches to address

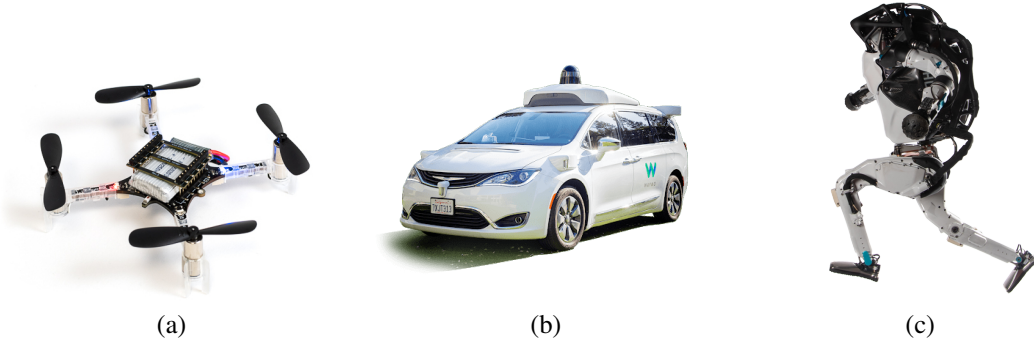


Figure 1.1: Some examples of safety-critical autonomous systems: (a) Crazyflie quadrotor [7], (b) Waymo self-driving car [8], and the (b) Atlas bipedal humanoid robot [5].

these issues rely on modern tools from control theory such as robust [12, 13] and adaptive [14, 15] control. In robust control methods, feedback policies are synthesized by solving an optimization problem that minimizes the degradation (i.e. performance output) due to the disturbances affecting the system. In contrast, adaptive control tools focus on directly learning the model parameters or uncertainties in the model to improve the designed feedback policy in an online-fashion. It is common [16–18] to utilize principles from both areas of research to design feedback policies that both provide quantifiable performance guarantees and improve continually over the lifetime of its operation. However, these approaches require bespoke analysis for every nonlinear robotic system and in general are not interchangeably applicable.

Tools from machine learning (ML) have also proven to be effective in learning system models for planning and control applications [19–21]. This area of research is also referred to as model-based reinforcement learning (MBRL) [22–24]. In MBRL, instead of directly learning the control policy to stabilize the system as in reinforcement learning (RL), a model is learned using ML techniques and model predictive control (MPC) approach is used to stabilize the system. However learning will never be fully accurate due limitations in collected data and assumptions on the structure of the uncertainty. And a control policy designed without careful consideration to the uncertainties arising out of errors in the model learning process may result in malfunctions and failures as previously discussed. Therefore, the safe planning and control policies must also be designed for safety-critical systems with learned components.

A further challenge is the problem of jointly learning certificates and control policies. The certificates that prove stability and safety in the presence of disturbances are obtained by solving prohibitively expensive optimization problems. Moreover, certification procedures [25–27] are typically solved as a feasibility problem and are not guaranteed to provide a solution unless the correct set of hyperparameters are selected prior to the computation. However, it would be more desirable if these procedures report solutions under certain alterations to the hyperparameters of the

problem. For example, a certificate and controller that are valid in a smaller region of the system's state space or converge to the equilibrium at a slower rate than desired will be more useful than reporting a solution that is determined to be infeasible.

It is evident from the above considerations that the following requirements are important to designing planning and control modules for safety-critical systems:

- Design real-time collision checking methods to continuously ensure safety along the planned trajectory and enable faster re-planning in the event that it is necessary;
- Obtain quantifiable tracking error bounds between the planned trajectory and the state of the system in the presence of bounded disturbances;
- Quantify modeling errors that emerge out of ML-based system identification techniques and obtain guaranteed performance bounds for the system during learning transients;
- Develop methods that synthesize certificates and control policies that provide the desired safety guarantees for nonlinear systems.

## 1.2 General Problem Formulation and Contributions

Consider a dynamical system with the state  $x \in \mathbb{R}^n$  and control  $u \in \mathbb{R}^m$  given by the following differential equation

$$\dot{x}(t) = f(t, x, u) + h(t), \quad x(0) = x_0,$$

where  $f(t) \in \mathbb{R}^n$  is known function representing the nominal vector field of the system and  $h(t) \in \mathbb{R}^n$  denotes the uncertainty in the system. The uncertainty may also depend on the states and the control input but without loss of generality is represented here as explicitly depending on time. Additionally, a desired trajectory  $x^*(t) \in \mathbb{R}^n$  is obtained from a motion planner [28–31], based on the nominal model of the system satisfying

$$\dot{x}^*(t) = f(t, x^*, u), \quad x^*(0) = x_0^*.$$

We proceed to define the problems that are critical to certifying safety for planning and control approaches in nonlinear systems. For simplicity, we define the problems in their full generality, however, as we provide solutions to these problems in the forthcoming chapters we bring in further assumptions to make the problems tractable.

**Problem 1.1** (Collision Checking). *Let  $\mathcal{B} \subset \mathbb{R}^n$  represent obstacles in the environments or state constraints that the system remains outside of. In order to certify that the desired trajectory is safe,*

the following inequality must hold

$$\|x^*(t) - b\| < \Delta_{\text{tol}}, \quad \forall b \in \mathcal{B}, \forall t \in [0, \tau],$$

where  $\Delta_{\text{tol}} > 0$  is some tolerance that accounts for tracking error and sensing inaccuracies.

In this dissertation, we introduce a family of algorithms to evaluate the minimum separation distance, tolerance verification, and collision detection queries between trajectories described as absolutely continuous parametric curves and obstacles. The obstacles are defined as convex polytopes, parametric curves, or any other compact set to which minimum distances can be computed. A main feature of the proposed algorithms is their ability to provide proximity queries for a large class of parametric curves, more general than ones that have been considered previously [32–36]. This is in part enabled by the efficient computation of convex hulls for parametric curves based on the arc length for any sub-interval in their domain, and additionally by fast procedures to evaluate the minimum separating distance, tolerance verification, and collision detection queries using interval branch-and-bound methods. Such queries are useful in scenarios when the motion planner’s candidate trajectory incurs a distance-based penalty for approaching close to an obstacle, must maintain a safe distance from an obstacle, or must not intersect with the obstacle’s geometry.

**Problem 1.2** (Guaranteed Tracking Performance). *Given bounded uncertainty, i.e.  $\|h(t)\| \leq \Delta_h$  for all  $t \geq 0$ . In order to certify guaranteed tracking performance, a control input  $u(t)$  must be designed so that*

$$\|x(t) - x^*(t)\| \leq \rho, \quad \forall t \geq 0,$$

where  $\rho > 0$  is the desired maximum tracking error bound.

We present an approach for safe feedback motion planning for control-affine nonlinear systems that relies on contraction theory-based solution for exponential stabilizability around trajectories and  $\mathcal{L}_1$ -adaptive control for handling uncertainties and providing guarantees for transient performance and robustness, which we refer to as the  $\mathcal{CL}_1$  robust adaptive control framework. We present a constructive design of feedback strategy for nonlinear systems using control contraction metrics (CCMs) and  $\mathcal{L}_1$ -adaptive control that provides strong guarantees of transient performance and robustness for a large class of control-affine nonlinear systems. Furthermore, we show how this control architecture induces tubes that can be flexibly changed to ensure safety based on the uncertainty in the system and the environment. In particular, this flexibility is provided by the architecture of the  $\mathcal{L}_1$ -adaptive control by decoupling the control loop from the estimation loop [15]. In this way, the width of the certifiable tubes can be adjusted allowing the safe operation of a robot in tight confines. The proposed control framework is planner-agnostic and is designed to work with any planner capable of generating desired state and control trajectories using the known (learned



or nominal) model. This feature enables the framework to be used in conjunction with many popular planning algorithms such as differential dynamic programming [37], model predictive path integral control [38], and sampling-based planners [39], among many others [31, 40].

**Problem 1.3** (Safe Learning and Control). *Given a learned vector field  $\hat{f}(t, x, u)$ , first quantify the uncertainty in the system*

$$\hat{f}(t, x, u) - f(t, x, u) \leq \Delta_{\hat{h}} \quad \forall x \in \mathcal{X}, u \in \mathcal{U}, t \geq 0,$$

*where  $\mathcal{X}$  and  $\mathcal{U}$  are compact subsets of the real coordinate space. In order to certify guaranteed tracking performance, a control input  $u(t)$  must be designed so that*

$$\|x(t) - \hat{x}^*(t)\| \leq \rho, \quad \forall t \geq 0,$$

*where  $\rho > 0$  is the desired maximum tracking error bound and  $\hat{x}^*$  is the desired trajectory obtained based on the learned model.*

We propose a learning-based control framework using robust adaptive control theory for nonlinear systems that ensures improvement of optimality and performance while simultaneously guaranteeing safety. The safety guarantees are composed of apriori computable transient performance bounds and robustness margins. We rely on Bayesian learning in the form of Gaussian process (GP) regression to learn the state and time-dependent model uncertainties from noisy measurements. We use the predictive distribution provided by GP learning to compute high-probability error bounds for the estimated uncertainties. These estimates are then incorporated within the  $\mathcal{CL}_1$  robust adaptive control framework. A critical feature of the proposed framework is that it enables MBRL algorithms to achieve optimality as learning progresses but the safety is guaranteed at all times regardless of the quality of the learned model. Moreover, the improved model knowledge and the tighter performance bounds are then incorporated into the planner used by the MBRL algorithm to generate more optimal but still safe trajectories.

**Problem 1.4** (Learning Certificates and Control Policies). *Design a certificate  $V(x, x^*)$  and control input  $u(t)$  so that*

$$\|x(t) - x^*(t)\| \leq \kappa(\|x_0 - x_0^*\|, t), \quad \forall x \in \mathcal{X}, x^* \in \mathcal{X}^* \subseteq \mathcal{X}, t \geq 0,$$

*where  $\kappa$  may belong to class  $\mathcal{KL}$  function for asymptotic stability,  $\kappa(x, t) = Cxe^{-\lambda t}$  for exponential stability (with constants  $C, \lambda > 0$ ), or represents convergence to an invariant set.*

We propose a method of learning certified control policies for nonlinear control systems. We provide the certification by jointly learning incremental Lyapunov functions (ILFs) and control

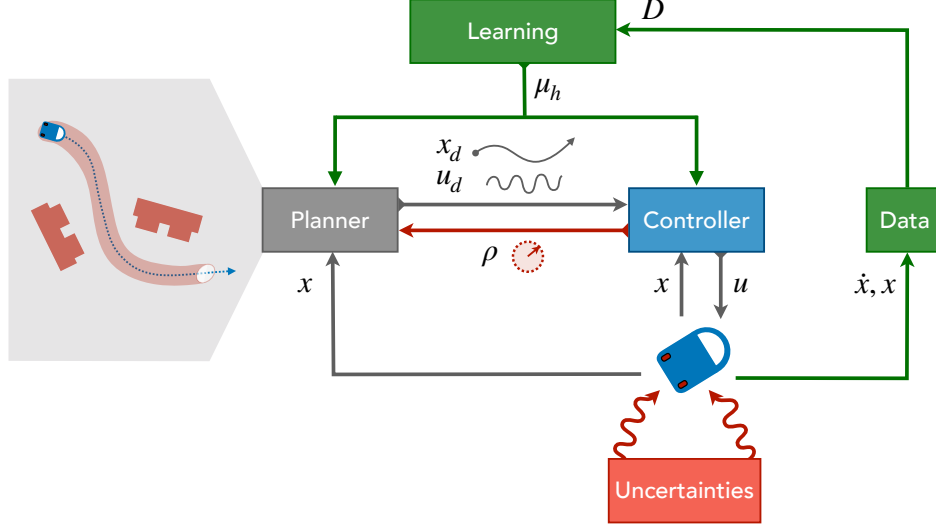


Figure 1.2: An illustration of the safe planning and control pipeline.

policies that prove the incremental exponential stability of the closed-loop systems. The space of candidate feedback laws and ILFs are parameterized using deep neural networks (DNNs) and are trained by minimizing the empirical loss that measures the violations of the Lyapunov conditions at different training points in the state space. Our framework utilizes interval analysis to discover the regions of the state space where the Lyapunov conditions are guaranteed to hold and we provide a region of attraction based on the result. This ensures that a viable solution may be recovered from even poorly trained models of the certificate and controller, albeit with smaller regions of attraction. We provide similar results for CCMs.

### 1.3 General Framework

The main focus of the framework proposed in this dissertation is ensuring that different components in the planning and control pipeline of an autonomous system are certified to give *complete* results and ensure *predictable* behavior. Since these systems form critical components of autonomous system, a failure of one of these systems might result in the failure of the entire system. The *completeness* of an algorithm ensures that if a property, for example being collision-free, is in violation then the algorithm must report a violation. This may also include reporting false negatives, i.e. Type II errors, however from a safety perspective this is preferable to a true negative. The *predictability* of an algorithm, ensures that the true system will behave close to the intended behavior of the nominal system. The collision checking algorithms and the verification of Lyapunov functions (LFs) presented in this work are complete, whereas, the proposed control architectures ensure predictable behavior of the system in the presence of uncertainties. The com-

combination of these properties ensures that the system remains safe during its operational lifetime.

In Fig. 1.2, we provide an illustration of the proposed safe planning and control framework. The two main constituents of the controller are a baseline control for trajectory tracking and an  $\mathcal{L}_1$ -adaptive control augmentation to handle the disturbances [41]. The baseline control ensures incremental exponential stability (IES) of the nominal system, i.e. without considering uncertainties. The stability of the system is certified by searching for an ILF or a CCM. A controller may also be jointly searched along with the certificate, otherwise the ILF can also be treated as a control Lyapunov function (CLF) and a Sontag-like universal formula [42] for the baseline control. We will present both approaches in this work. The other important component of controller is the  $\mathcal{L}_1$ -adaptive augmentation for the partial compensation of the uncertainties and providing guarantees for transient performance and robustness. In the  $\mathcal{L}_1$  control architecture, estimation is decoupled from control, thereby allowing for arbitrarily fast adaptation subject only to hardware limitations, [15]. Both these components together make sure that the system behaves predictably in real world environments, and closely tracks the desired trajectory designed by the planning module.

Based on the level of uncertainty in the system and the properties of the baseline and adaptive controllers, we analytically obtain the tracking performance bounds on the system. The performance bounds correspond to the maximum deviation of the system away from the desired trajectory, which is directly applicable to collision checking in motion planning scenarios. Typically, candidate trajectories are not checked for collision detection exactly with the boundary of obstacles since any inaccuracies in sensing or tracking behavior might cause collisions. In our framework the planner incorporates the tracking performance bounds as a minimum safety clearance to keep away from obstacles [43]. These algorithms are both complete and sound, in that they only report a collision if and only if there is actually a collision along the desired path of the vehicle.

In order to improve the performance of the system during active operation we propose a methodology to collect data and learn the unmodeled dynamics in the system [44]. The uncertainties are learned as non-parametric models using GP regression. However, since the model is trained on finite data samples with possibly incorrect GP hyper-parameters, the learned representation of the uncertainty will not exactly match the true uncertainty in the system. We provide a complete algorithm that quantifies the error incurred during learning for the design of the  $\mathcal{L}_1$ -adaptive control augmentation. Further, the mean dynamics of the uncertainties are used in the planner to design more optimal trajectories and in the baseline controller to ensure IES of the learned model. This naturally leads to less conservative designs of the planning and control modules. As the learning improves, the controller is able to guarantee tighter performance bounds and the planner can utilize a better model and the tighter bounds to produce more optimal trajectories.

## 1.4 Thesis Overview

The dissertation is organized as follows:

- In Chapter 2 we propose computationally efficient techniques for computing distances and checking feasibility of planned trajectories [43]. We present the problem formulation for the different types of proximity queries between parametric curves and obstacles in the environment. The analysis and results for the construction of convex hulls for parametric curves that provide bounds on the minimum separating distance are provided. The algorithms for each of the proximity queries are outlined. Additionally, we present numerical examples and benchmarks to illustrate the efficacy of these methods. The proofs for all the claims in this chapter are provided in Appendix A.
- In Chapter 3 we present the  $\mathcal{CL}_1$ -framework [41] that certifies guaranteed transient performance bounds for nonlinear control-affine systems in the presence of uncertainties. We present the problem formulation as the design of an  $\mathcal{L}_1$ -adaptive control augmentation to compensate for uncertainties in the system. The baseline feedback law, designed using contraction theory, provides exponential stability guarantees to the nominal system (i.e. without disturbances). The performance analysis of the closed-loop system is provided along with simulation examples of nonlinear systems to demonstrate how contraction theory-based  $\mathcal{L}_1$ -adaptive control can be used in conjunction with traditional motion planning algorithms to obtain provably safe trajectories. The proofs for all the claims in this chapter are provided in Appendix B.
- In Chapter 4 we extend the  $\mathcal{CL}_1$ -framework [44] to certify guaranteed transient performance bounds for nonlinear systems with GP learned components. The  $\mathcal{CL}_1$  controller ensures that control objectives are met while providing safety certificates. Furthermore, the controller incorporates any available data into GP models of uncertainties, which improves performance and enables the motion planner to achieve optimality safely. The procedure and analysis behind the quantification of the uniform bounds on the modeling errors incurred during GP regression is also presented. We use the  $\mathcal{CL}_1$ -framework to obtain the tracking performance guarantees for the nonlinear system with the learned uncertainties. We provide illustrative examples that demonstrate the benefits of the performance guarantees provided by our framework at different episodes of the learning process for a variety of nonlinear systems.
- In Chapter 5 we present a framework to jointly learn ILFs and the associated control policies. We leverage the function approximation capabilities of DNNs to parameterize and learn the certificates and the control policies. Our methodology's main feature is the verification of

the validity of the learned certificate and controller using interval arithmetic. The framework allows for violations in certain regions of the state space and provides the region of attraction defined by the learned ILF. We provide some illustrative examples that showcase the benefits of our framework.

- In Chapter 6 we discuss directions for future research and provide concluding remarks.

# Chapter 2

## Fast Proximity Queries for Safe Motion Planning

### 2.1 Introduction

Autonomous robots often operate in rapidly changing environments, and the ability to accurately and quickly predict future collisions is crucial for safely meeting task objectives. As proximity queries serve one of the major bottlenecks in motion planning frameworks [6], algorithms that efficiently assess the proximity of obstacles relative to the future states of the robot will greatly improve the run-time performance of the robot while guaranteeing safe operating procedures.

Fast methods for computing proximity queries between polyhedral objects have been widely studied and developed over the past few decades. The algorithms described in [45, 46] perform several different types of proximity queries between convex polytopes. Proximity between general polyhedral objects using hierarchical representations of convex bounding volumes are investigated in [47] and [48]. In [49], the authors evaluate the proximity queries for polyhedral objects using convex surface decomposition.

It is challenging to assess the proximity between objects in motion. Recent work has focused on developing methods for computing continuous collision detection, a type of query that evaluates the first time of contact between objects in motion. The most practical of these methods is known as conservative advancement, introduced in [50] and [51], and performs several static proximity queries between polyhedral or polygon-soup objects to accurately perform such queries. Conservative advancement has been applied in trajectory refinement algorithms [52] to obtain collision-free cubic B-spline trajectories. However, in the context of motion planning, such methods are computationally expensive and lack the ability to evaluate other proximity queries such as the minimum separating distance or tolerance verification between moving objects. As modern motion planning and trajectory optimization methods generate candidate paths as rectifiable parametric curves (e.g. Lagrange polynomials [29], Legendre polynomials [53], Bézier curves [54–56], B-splines [57], Dubins paths [58], Pythagorean Hodograph curves [59]), the proximity algorithms must be able to handle such representations.

Generally, proximity to parametric curves is handled by algebraic, interval analysis, or curve

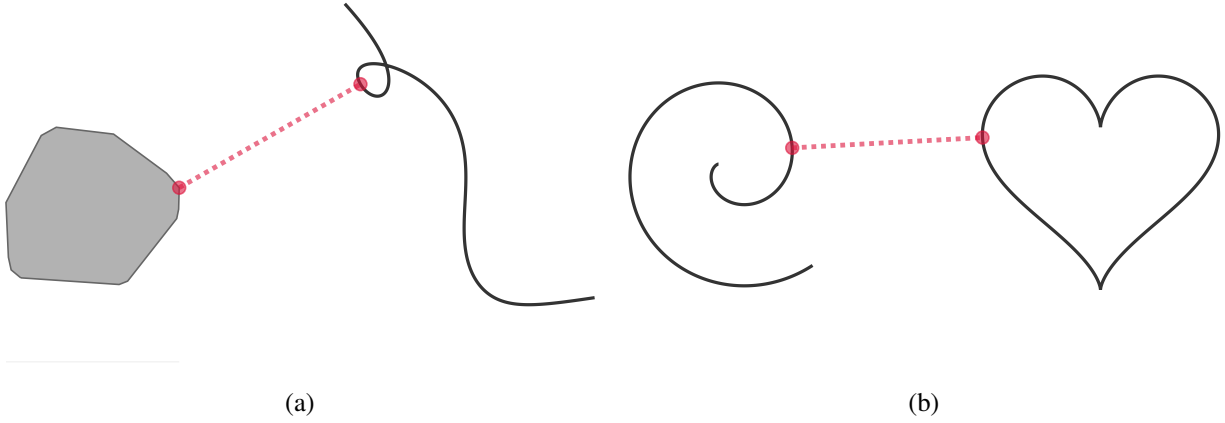


Figure 2.1: Points closest between (a) a convex polygon and a Bézier curve, and (b) an involute of a circle and a heart-shaped curve.

subdivision methods. Finding multiple intersections between parametric curves by implicitization and eigenvalue decomposition is discussed in [32]. In [33], the authors present an approach to compute the exact minimum separating distance between differentiable continuous freeform curves by solving a set of nonlinear equations. Computing the minimum separating distance between Bézier curves by sweeping a sphere along one of the curves and eliminating sections of the curve which lie outside of it through subdivision was introduced in [34]. A similar technique [35] was used to find the closest point on a free form curve to a point in free space. In [36], the authors present an efficient method for computing the minimum separating distances to Bézier curves using subdivision methods. The curves are recursively subdivided until the bounds on the minimum separating distance between the control polygons for each of the curves are within some prescribed accuracy. However, these methods do not hold in general for parametric curves and lack the computational efficiency to be used in motion planning algorithms.

In this work, we introduce a family of algorithms to evaluate the (i) minimum separation distance, (ii) tolerance verification, and (iii) collision detection queries between absolutely continuous parametric curves and obstacles. The obstacles are defined as convex polytopes, parametric curves, or any other compact set to which minimum distances can be computed, Fig. 2.1. A main feature of the proposed algorithms is their ability to provide proximity queries for a large class of parametric curves, more general than ones that have been considered previously [32–36]. Such queries are useful in scenarios when the motion planner’s candidate path (i) incurs a distance-based penalty for approaching close to an obstacle, (ii) must keep a safe distance from an obstacle, or (iii) must not intersect with the obstacle’s geometry.

## 2.2 Problem Formulation

The parametric equation of a curve is given by a function  $\psi : \mathcal{I} \rightarrow \mathbb{R}^d$ , where  $\mathcal{I} \subset \mathbb{R}$  is a compact interval such that  $|\mathcal{I}| > 0$ , and  $d \in \mathbb{N}$ . We define the curve over the closed sub-interval  $\mathcal{Q} \subseteq \mathcal{I}$  (such that  $|\mathcal{Q}| > 0$ ) as the following compact set

$$\Psi_{\mathcal{Q}} = \{\psi(t) \in \mathbb{R}^d : t \in \mathcal{Q}\}. \quad (2.1)$$

We proceed to define the proximity query problems between a curve  $\Psi_{\mathcal{I}}$  and an object  $\mathcal{B}$  represented as a nonempty compact set in  $\mathbb{R}^d$  under the following assumptions.

**Assumption 2.1.** *The function  $\psi$  is absolutely continuous over its entire domain  $\mathcal{I}$ , i.e., for every  $\epsilon > 0$  there exists a  $\delta > 0$ , such that for each  $n \in \mathbb{N}$ , if the collection of mutually disjoint closed sub-intervals  $\{[\alpha_i, \beta_i] \mid i = 1, \dots, n\}$  satisfies  $\sum_i |\alpha_i - \beta_i| < \delta$ , then  $\sum_i \|\psi(\alpha_i) - \psi(\beta_i)\| < \epsilon$ .*

**Assumption 2.2.** *The function  $\psi$  is not a constant map over the entire domain  $\mathcal{I}$ , i.e.,  $\psi(t) \neq y$  for all  $t \in \mathcal{I}$  for some  $y \in \mathbb{R}^d$ .*

**Problem 2.1** (Minimum Separating Distance). *The minimum separating distance between the parametric curve  $\Psi_{\mathcal{I}}$  and the compact set  $\mathcal{B}$  is defined as*

$$d_{\min}(\Psi_{\mathcal{I}}, \mathcal{B}) = \min_{a \in \Psi_{\mathcal{I}}, b \in \mathcal{B}} \|a - b\|. \quad (2.2)$$

*The points  $\psi(t^*) \in \Psi_{\mathcal{I}}$  and  $b^* \in \mathcal{B}$  that verify*

$$d_{\min}(\Psi_{\mathcal{I}}, \mathcal{B}) = \|\psi(t^*) - b^*\|$$

*are the pair of points that lie closest to each other on the respective sets, as shown in Fig. 2.2a. In addition to considering the minimum separating distance over the entire curve,  $\Psi_{\mathcal{I}}$ , the definition can be applied to some continuous compact sub-interval,  $\mathcal{Q} \subseteq \mathcal{I}$ , of the curve as well. This section of the curve is referred to by  $\Psi_{\mathcal{Q}}$ .*

**Problem 2.2** (Tolerance Verification). *Tolerance verification is defined as a predicate function with three arguments; two objects,  $\Psi_{\mathcal{I}}$  and  $\mathcal{B}$ , and a tolerance,  $\Delta > 0$ . The function evaluates the inequality*

$$d_{\min}(\Psi_{\mathcal{I}}, \mathcal{B}) > \Delta \quad (2.3)$$

*and returns either true or false. Fig. 2.2b depicts a scenario, when a curve is separated from a compact set by a distance greater than  $\Delta$ .*



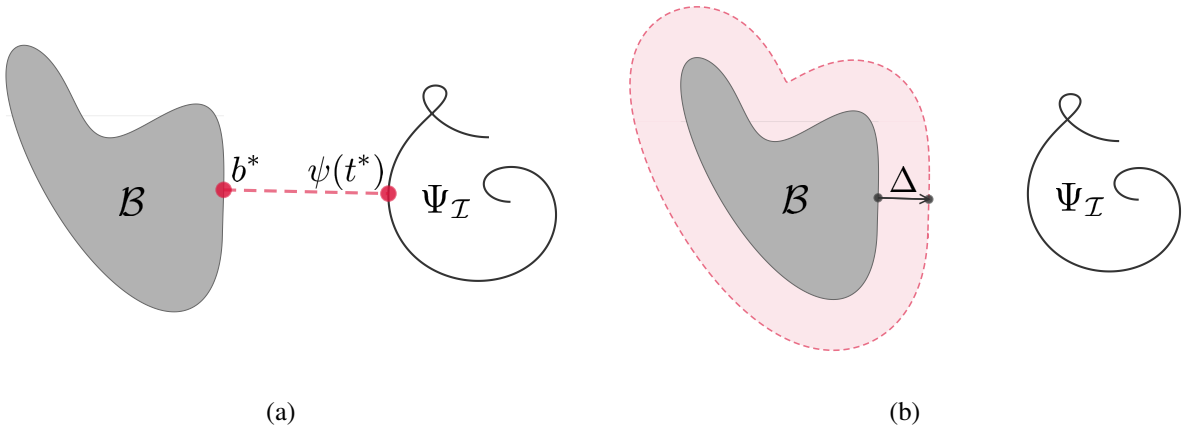


Figure 2.2: (a) The red dashed line shows the segment connecting the pair of closest points between  $\Psi_{\mathcal{I}}$  and  $\mathcal{B}$  respectively. (b) The red shaded region specifies the  $\Delta$ -tolerance afforded to  $\mathcal{B}$  when computing the tolerance verification between the two objects.

**Problem 2.3** (Collision Detection). *The collision detection function is defined as a predicate function with two arguments. These arguments are two objects,  $\Psi_{\mathcal{I}}$  and  $\mathcal{B}$ . The function determines the truth of the following statement:*

$$\Psi_{\mathcal{I}} \cap \mathcal{B} \neq \emptyset. \quad (2.4)$$

**Remark 2.1.** *Notice that the solution of Problem 2.1 implies the solution of Problem 2.2, which in turn implies the solution of Problem 2.3. Nevertheless, defining each problem by itself is useful because, as we will see in Section 2.4, the numerical methods can be specifically tailored for each problem in order to improve their computational efficiency.*

## 2.3 Bounding Methods and Analysis

The problems discussed in Section 2.2 are generally non-convex and difficult to solve<sup>1</sup>. Additionally, when a feasible solution is found it is still difficult to verify that the solution is indeed the global minimum. However, methods that rely on branch-and-bound and interval analysis [60] obtain a solution with a certificate of its optimality within some prescribed accuracy. This is achieved by successively *branching* the problem into smaller and smaller sub-problems over which *bounds* on the optimal solution are computed through relaxation. In this section we present the results and the analysis behind the relaxation of Problem 2.1 and its sub-problems.

<sup>1</sup>Non-convex problems are considered to be at least NP-hard.

### 2.3.1 Upper Bound on the Length of a Curve

As the presented methods require the computation of the arc length of parametric curves, we will only consider parametric curves that are rectifiable, *i.e.* they possess a finite arc length over their domain. The absolute continuity of  $\psi$  over  $\mathcal{I}$  is a sufficient condition [61] for the curve to be rectifiable<sup>2</sup>, and the corresponding arc length function over the interval  $\mathcal{Q} \subseteq \mathcal{I}$  is defined as

$$s_\psi(\mathcal{Q}) = \int_{\mathcal{Q}} \|\psi'(t)\| \, dt, \quad (2.5)$$

where  $\psi$  has a finite derivative  $\psi'$  almost everywhere and is Lebesgue integrable.

As branch-and-bound techniques involve repeated calculations on the curve, for reasons having to do with computational efficiency, it is extremely desirable to have a closed-form expression for the antiderivative of the square root of the inner product of  $\psi'$ . However, the arc length of rectifiable parametric curves cannot be obtained in general. Even for simple curves described with polynomial or sinusoidal basis functions, one cannot express the integral in terms of elementary functions. In the following result, we introduce an upper bound on  $s_\psi(\mathcal{Q})$  in (2.5) that is better suited to provide a closed-form expression.

**Lemma 2.1.** *Let  $\psi$  be an absolutely continuous function and have a derivative  $\psi'$  on the compact interval  $\mathcal{I}$ . For any closed  $\mathcal{Q} \subseteq \mathcal{I}$  define the upper bound*

$$u_\psi(\mathcal{Q}) = \sqrt{|\mathcal{Q}| \int_{\mathcal{Q}} \psi'(t)^\top \psi'(t) \, dt}, \quad (2.6)$$

where  $|\mathcal{Q}|$  is the length of the sub-interval. Then for the arc length function of the curve  $s_\psi$  (from (2.5)) the following inequality holds

$$s_\psi(\mathcal{Q}) \leq u_\psi(\mathcal{Q}).$$

For common parametric curves with polynomial or trigonometric basis functions, such as the ones typically employed in trajectory generation methods [57], the antiderivative for the inner product of  $\psi'$  is readily available, which greatly reduces the computation time. In Table 2.1, notice the improvement in computation time when the antiderivative is known in closed-form.

---

<sup>2</sup>More generally, any function of bounded variation is rectifiable [62], however, the derivatives for such functions may not exist almost everywhere.

Parametric Equations	Median Time (ns)	
	$s_\psi(\mathcal{Q})$	$u_\psi(\mathcal{Q})$
$(2 \cos(t), \sin(t))$	1710.70	58.02
$((t^3 + t), t)$	1673.80	44.69
$((t + 1)^{-1}, t)$	2102.11	38.43

Table 2.1: Comparison between the median evaluation time for computing  $s_\psi(\mathcal{Q})$  and  $u_\psi(\mathcal{Q})$  for several different intervals  $\mathcal{Q} \subseteq [0, 1]$ . For each example presented in the table, the antiderivative of the integrand in (2.5) is not available in closed form and must be numerically integrated, whereas the antiderivative of the integrand in (2.6) is available in closed-form. The numerical integration uses the Gauss-Kronrod quadrature formula procedure over 15 points.

### 2.3.2 Convex Hull on Sub-intervals of a Curve

The following theorem establishes a convex hull for parametric curves on any sub-interval based on their upper bound of the arc length from (2.6).

**Theorem 2.1** (Convex Hull). *Let  $\psi$  be an absolutely continuous function defined on the compact interval  $\mathcal{I}$ . For any closed interval  $[\alpha, \beta] \equiv \mathcal{Q} \subseteq \mathcal{I}$ , define the convex compact set*

$$\mathcal{U}_{\mathcal{Q}} = \{x \in \mathbb{R}^d : \|\psi(\alpha) - x\| + \|x - \psi(\beta)\| \leq u_\psi(\mathcal{Q})\}. \quad (2.7)$$

*Then the curve defined on the interval  $\mathcal{Q}$  satisfies*

$$\Psi_{\mathcal{Q}} \subset \mathcal{U}_{\mathcal{Q}}. \quad (2.8)$$

The convex hull in Theorem 2.1 is a compact interval for curves in  $\mathbb{R}$ , and is also an ellipsoid for curves in higher dimensions. This geometric relationship to an ellipsoid is particularly useful because there are several methods [45, 63, 64] to cheaply compute the minimum separating distance to an ellipsoid. For a convex hull  $\mathcal{U}_{\mathcal{Q}}$  of the function  $\psi$  evaluated over a sub-interval  $[\alpha, \beta] \equiv \mathcal{Q} \subseteq \mathcal{I}$ , the foci of the ellipse are at  $\psi(\alpha)$  and  $\psi(\beta)$ . The major axis has a length of  $u_\psi(\mathcal{Q})$ , and the minor axes are of equal lengths  $\sqrt{u_\psi(\mathcal{Q})^2 - \|\psi(\alpha) - \psi(\beta)\|^2}$ . This construction is illustrated in Fig. 2.3.

**Remark 2.2.** *Fat arcs [65, 66] provide tighter bounds with cubic convergence, however, the non-convexity of the bounding region is problematic when computing proximity queries. Additionally, they are only useful in the context of planar Bézier curves and spirals. The convergence behavior of the ellipsoidal bounding region to the curve will be explored in the future.*

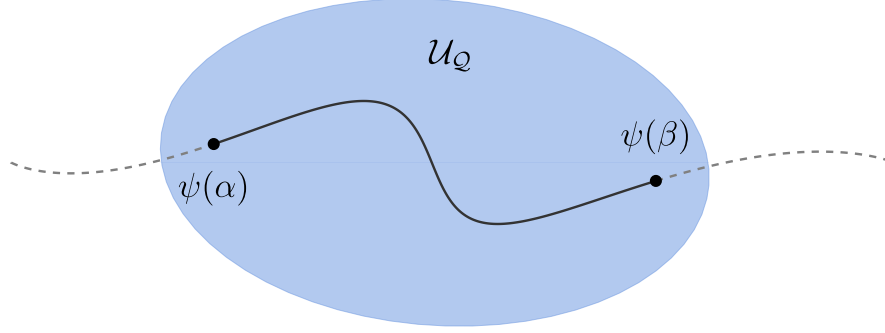


Figure 2.3: The blue shaded region shows the convex set  $\mathcal{U}_Q$ , which contains the function  $\psi$  evaluated over the sub-interval  $[\alpha, \beta]$ . The dashed-line is the evaluation of  $\psi$  over its entire domain.

### 2.3.3 Relaxation of Problem 2.1

Now that since we have enclosed a parametric curve inside a convex hull, Problem 2.1 can be relaxed, and a more tractable problem can be solved by computing the minimum separating distance to the convex hull. The solution to this relaxed problem provides a lower bound to the optimal solution of the original problem. We define this lower bound on the minimum separating distance  $d_{\text{lb}}$  between  $\Psi_Q$  and a compact set  $B$  as

$$d_{\text{lb}}(\Psi_Q, B) = \min_{x \in \mathcal{U}_Q, b \in B} \|x - b\|, \quad (2.9)$$

where  $\mathcal{U}_Q$  is the convex hull corresponding to  $\Psi_Q$  from (2.7). Fig. 2.4a shows the lower bound between a convex polygon and a parametric curve evaluated over a sub-interval. As one might expect, if the object  $B$  is also a parametric curve, then  $d_{\text{lb}}$  is computed between the convex sets constructed from each of the curves (as shown in Fig. 2.4b), as

$$d_{\text{lb}}(\Psi_Q, \Phi_{\mathcal{R}}) = \min_{x \in \mathcal{U}_Q, y \in \mathcal{V}_{\mathcal{R}}} \|x - y\|, \quad (2.10)$$

where  $\Phi_{\mathcal{R}}$  is the parametric curve  $\phi : \mathbb{R} \supset \mathcal{R} \rightarrow \mathbb{R}^d$ , and  $\mathcal{V}_{\mathcal{R}}$  is the convex hull of  $\Phi_{\mathcal{R}}$ .

In addition to the lower bound, an upper bound on the optimal solution must also be defined in order to bound the solution in a compact interval. The upper bound on the minimum separating

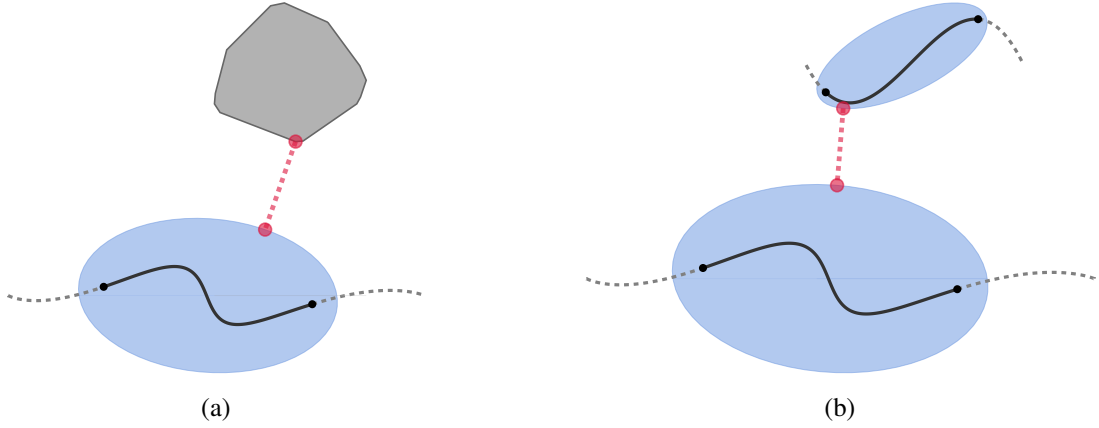


Figure 2.4: Lower bound on minimum separating distance between (a) a parametric curve and a convex polygon, and (b) two parametric curves.

distance  $d_{\text{ub}}$  between  $\Psi_Q$  and a compact set  $B$  is given by

$$d_{\text{ub}}(\Psi_Q, B) = \min_{b \in B} \|x'(\Psi_Q) - b\|, \quad (2.11)$$

such that  $x'(\Psi_Q) \in \Psi_Q$ . Similarly,  $d_{\text{ub}}$  between the two parametric curves  $\Psi_Q$  and  $\Phi_{\mathcal{R}}$  is

$$d_{\text{ub}}(\Psi_Q, \Phi_{\mathcal{R}}) = \|x'(\Psi_Q) - x''(\Phi_{\mathcal{R}})\|, \quad (2.12)$$

such that  $x'(\Psi_Q) \in \Psi_Q$  and  $x''(\Phi_{\mathcal{R}}) \in \Phi_{\mathcal{R}}$ . In the numerical implementation as seen in Fig. 2.5, the functions  $x'$  and  $x''$  select the middle points of  $\Psi_Q$  and  $\Phi_{\mathcal{R}}$  respectively. Heuristics to select points in the curve set based on the relative configuration of the objects rather than simply the midpoints may find tighter bounds, however, this is beyond the scope of the dissertation.

We present analysis on the functions  $d_{\text{lb}}$  and  $d_{\text{ub}}$ . We show for each interval  $Q \subseteq \mathcal{I}$  that indeed these functions bound the optimal solution on that interval, and that the difference in upper and lower bounds converges uniformly to zero as  $|Q| \rightarrow 0$ . The boundedness and uniform convergence results are necessary in order to guarantee the convergence of interval branch-and-bound algorithms in finite time.

**Theorem 2.2** (Boundedness). *Let  $\mathcal{B}$  be a compact set in  $\mathbb{R}^d$ , and  $\Psi_{\mathcal{I}}$  be the curve  $\psi : \mathcal{I} \rightarrow \mathbb{R}^d$ . Then, for every closed  $Q \subseteq \mathcal{I}$ , we have*

$$d_{\text{lb}}(\Psi_Q, \mathcal{B}) \leq d_{\text{min}}(\Psi_Q, \mathcal{B}) \leq d_{\text{ub}}(\Psi_Q, \mathcal{B}). \quad (2.13)$$

**Theorem 2.3** (Uniform Convergence). *Let  $\mathcal{B}$  be a compact set in  $\mathbb{R}^d$ , and  $\Psi_{\mathcal{I}}$  be the curve  $\psi : \mathcal{I} \rightarrow$*

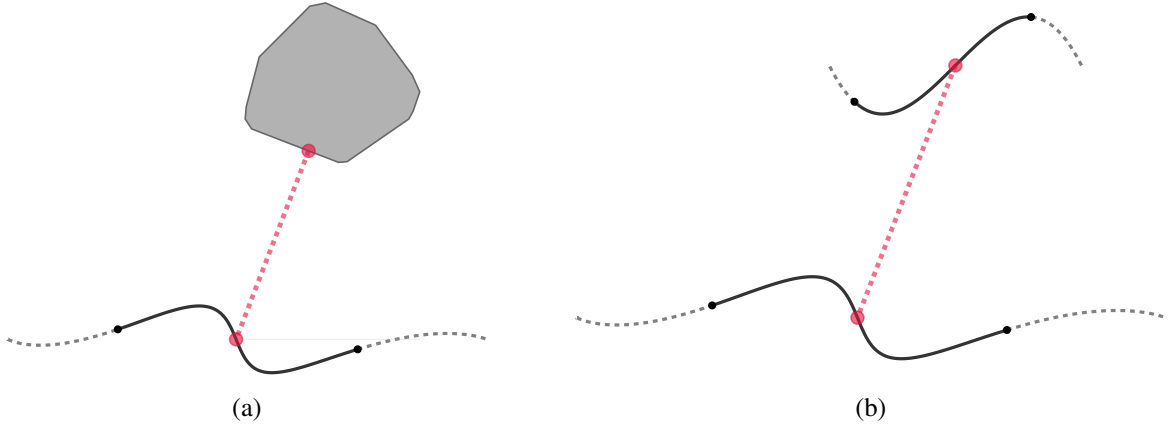


Figure 2.5: Upper bound on minimum separating distance between (a) a parametric curve and a convex polygon, and (b) two parametric curves.

$\mathbb{R}^d$ . For every  $\epsilon > 0$ , there exists a  $\delta > 0$  such that for all  $\{\mathcal{Q} : \mathcal{Q} \subseteq \mathcal{I}, |\mathcal{Q}| < \delta\}$ , we satisfy

$$d_{\text{ub}}(\Psi_{\mathcal{Q}}, \mathcal{B}) - d_{\text{lb}}(\Psi_{\mathcal{Q}}, \mathcal{B}) < \epsilon. \quad (2.14)$$

**Remark 2.3.** The minimization problems of (2.9) and (2.11) may also be nonconvex depending on the geometry of  $\mathcal{B}$ . We refer readers to [45, 46] for convex polytopes and [47, 48] for more complex models.

## 2.4 Proximity Queries

Given the boundedness results of Theorem 2.2 and the convergence results of Theorem 2.3, we present algorithms that evaluate the solutions to Problems 2.1 to 2.3 within some tolerance. Central to each of these algorithms is a process of interval subdivision and bounding. First, we present the methods to compute the minimum separating distance between a parametric curve and a compact set (or another parametric curve). Later, we highlight the variations of this algorithm for evaluating the tolerance verification and collision detection queries without computing the minimum separating distance between the objects.

### 2.4.1 Minimum Separating Distance

Given a parametric function  $\psi : \mathcal{I} \rightarrow \mathbb{R}^d$ , Algorithm 1 computes the minimum separating distance between the curve  $\Psi_{\mathcal{I}}$  and a compact set  $\mathcal{B}$ .

---

**Algorithm 1** Minimum Separating Distance ( $\Psi_{\mathcal{I}}, B$ )

---

```
1:  $\mathcal{L} \leftarrow \{\mathcal{I}\}$ 
2:  $\bar{d} \leftarrow d_{\text{ub}}(\Psi_{\mathcal{I}}, \mathcal{B})$ 
3:  $\underline{d} \leftarrow d_{\text{lb}}(\Psi_{\mathcal{I}}, \mathcal{B})$ 
4: while  $\bar{d} - \underline{d} > \epsilon$  do
5:    $\mathcal{X} \leftarrow \arg \min_{Q \in \mathcal{L}} d_{\text{lb}}(\Psi_Q, \mathcal{B})$ 
6:    $\mathcal{X}_L, \mathcal{X}_R \leftarrow \text{split}(\mathcal{X})$ 
7:    $\mathcal{L} \leftarrow \mathcal{L} \setminus \{\mathcal{X}\}$ 
8:    $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathcal{X}_L, \mathcal{X}_R\}$ 
9:    $\bar{d} \leftarrow \min_{Q \in \mathcal{L}} d_{\text{ub}}(\Psi_Q, \mathcal{B})$ 
10:   $\underline{d} \leftarrow \min_{Q \in \mathcal{L}} d_{\text{lb}}(\Psi_Q, \mathcal{B})$ 
11: end while
12: return  $\underline{d}$ 
```

---

We proceed to describe the algorithm. The general structure of the algorithm closely matches that of a vanilla branch-and-bound method [67]. Consider computing the minimum separating distance between a curve and a convex polygon as shown in Fig. 2.6a. A collection  $\mathcal{L}$ , which stores the intervals over which the bounds on the optimal solution are computed, is initialized with a single element: the entire domain  $\mathcal{I}$ . The relaxed problem is solved over the domain  $\mathcal{I}$  (Fig. 2.6b), and the upper and lower bound values are stored in the states  $\bar{d}$  and  $\underline{d}$  respectively. If the difference between the bounds is above a prescribed tolerance, the algorithm proceeds to the iterative phase.  $\mathcal{I}$  is split into sub-intervals  $\mathcal{X}_R$  and  $\mathcal{X}_L$ , over which the relaxations of the problems on the new intervals are solved (Fig. 2.6c). These sub-intervals are added to the collection  $\mathcal{L}$ , and the original domain  $\mathcal{I}$  is removed from the collection. The least upper and lower bounds on the solution are kept track of by  $\bar{d}$  and  $\underline{d}$ , respectively, on all the sub-intervals present in the collection  $\mathcal{L}$ . Each successive iteration sees the sub-interval with the least lower bound chosen for subdivision. Figures 2.6d and 2.6e show snapshots of the algorithm into the third and seventh iteration respectively. Notice that the algorithm preferentially selects sub-intervals closer to the polygon to subdivide because of the best-first search strategy.

Since the minimization problem of (2.2) is over a continuous space, the branch-and-bound method will repeatedly subdivide infinitely many times (whenever the objects  $\Psi_{\mathcal{I}}$  and  $\mathcal{B}$  do not collide). However, by setting an absolute tolerance  $\epsilon > 0$  on the solution, we obtain an  $\epsilon$ -suboptimal solution in a finite number of iterations, *i.e.*

$$d_{\min}(\Psi_{\mathcal{I}}, \mathcal{B}) \in [\underline{d}, \underline{d} + \epsilon], \quad (2.15)$$

where  $\underline{d}$  is the lower bound on the solution obtained from the minimum separating distance algorithm. The certificate proving that the global minimum lies in the interval shown in (2.15) can

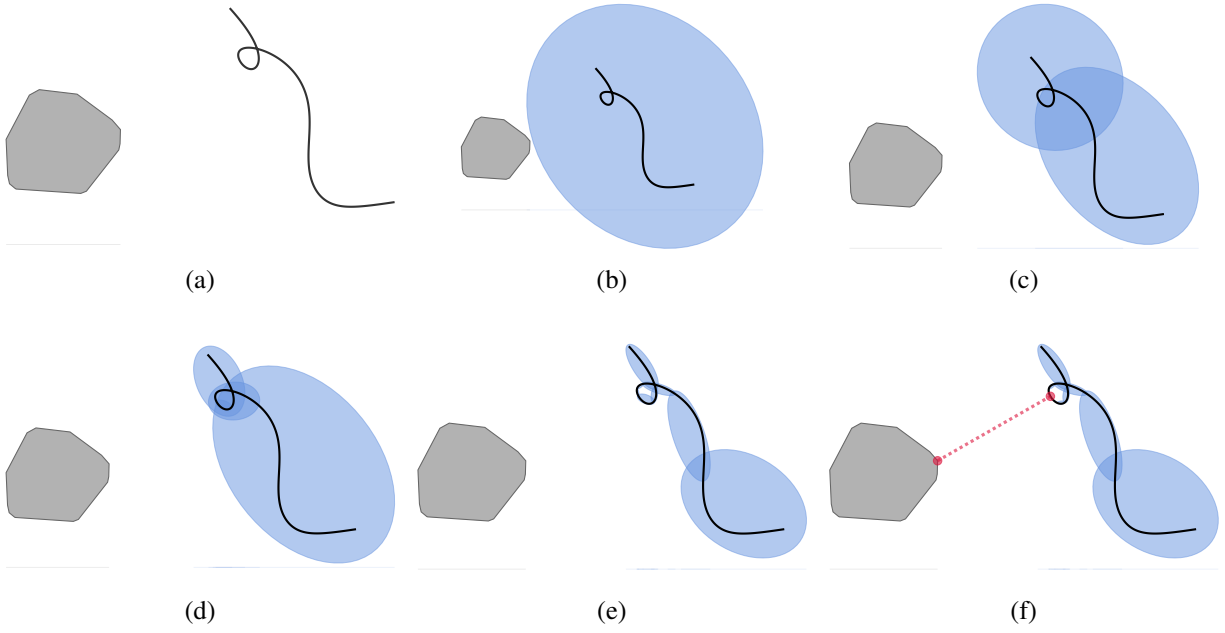


Figure 2.6: The different stages of Algorithm 1 when computing the minimum distance between a 13<sup>th</sup> order Bézier curve and a convex polygon.

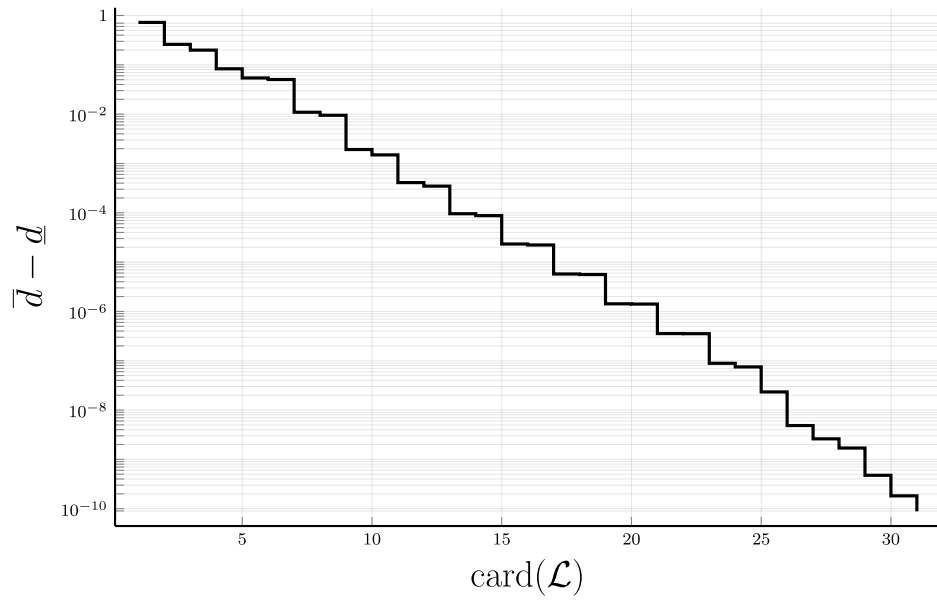


Figure 2.7: Convergence of  $\bar{d} - \underline{d}$  for the scenario presented in Fig. 2.6 to  $\epsilon = 10^{-10}$ .



be obtained by examining the collection  $\mathcal{L}$ . Figure 2.6f highlights the solution, when the bounds are separated by a magnitude of no more than  $\epsilon$ . The convergence of  $\bar{d} - \underline{d}$  to  $\epsilon$  for the example presented in Fig. 2.6 is shown in Fig. 2.7.

Algorithm 2 describes the procedure to evaluate the minimum distance from  $\Psi_{\mathcal{I}}$  to  $\Phi_{\mathcal{J}}$ , which is the trace of  $\phi : \mathcal{J} \rightarrow \mathbb{R}^d$ . As one might expect, structurally very little differs between Algorithms 1 and 2. The collection  $\mathcal{L}$  stores pairs of sub-intervals over which the bounds on the solution will be evaluated. The lower and upper bounds are computed using (2.10) and (2.12) respectively. Figure 2.8 shows the pair of closest points between Bézier curves computed using Algorithm 2.

---

**Algorithm 2** Minimum Separating Distance ( $\Psi_{\mathcal{I}}, \Phi_{\mathcal{J}}$ )

---

```

1:  $\mathcal{L} \leftarrow \{\{\mathcal{I}, \mathcal{J}\}\}$ 
2:  $\bar{d} \leftarrow d_{\text{ub}}(\Psi_{\mathcal{I}}, \Phi_{\mathcal{J}})$ 
3:  $\underline{d} \leftarrow d_{\text{lb}}(\Psi_{\mathcal{I}}, \Phi_{\mathcal{J}})$ 
4: while  $\bar{d} - \underline{d} > \epsilon$  do
5:    $\mathcal{X} \leftarrow \arg \min_{\{\mathcal{Q}, \mathcal{R}\} \in \mathcal{L}} d_{\text{lb}}(\Psi_{\mathcal{Q}}, \Phi_{\mathcal{R}})$ 
6:    $\mathcal{X}_L, \mathcal{X}_R \leftarrow \text{split}(\mathcal{X})$ 
7:    $\mathcal{L} \leftarrow \mathcal{L} \setminus \{\mathcal{X}\}$ 
8:    $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathcal{X}_L, \mathcal{X}_R\}$ 
9:    $\bar{d} \leftarrow \min_{\{\mathcal{Q}, \mathcal{R}\} \in \mathcal{L}} d_{\text{ub}}(\Psi_{\mathcal{Q}}, \Phi_{\mathcal{R}})$ 
10:   $\underline{d} \leftarrow \min_{\{\mathcal{Q}, \mathcal{R}\} \in \mathcal{L}} d_{\text{lb}}(\Psi_{\mathcal{Q}}, \Phi_{\mathcal{R}})$ 
11: end while
12: return  $\underline{d}$ 

```

---

**Remark 2.4.** The  $\mathcal{X}_L, \mathcal{X}_R \leftarrow \text{split}(\mathcal{X})$  operation from (line 6) of Algorithms 1 and 2 returns two mutually disjoint sets such that  $\mathcal{X}_L \cup \mathcal{X}_R = \mathcal{X}$ . When  $\mathcal{X}$  is an interval, the operation must ensure that both  $\mathcal{X}_L$  and  $\mathcal{X}_R$  are closed, and when  $\mathcal{X}$  is a collection of two intervals, the operation must only split the larger interval so as to preserve the uniform convergence property from Theorem 2.3. In the implementation<sup>3</sup>, the  $\text{split}(\mathcal{X})$  operation bisects the interval  $\mathcal{X}$ ; however, uneven splitting techniques may lead to better performance as seen in [36].

## 2.4.2 Tolerance Verification

In many applications an exact measurement on the minimum separating distance between objects is not required. A weaker method that verifies if two objects are separated by a distance greater than  $\Delta$  may be preferred for computational reasons. Since the states  $\bar{d}$  and  $\underline{d}$  in Algorithms 1 and 2 keep track of the the bounds on the global optimum, the algorithm will terminate early if either  $\bar{d}$

<sup>3</sup><https://github.com/arlk/CurveProximityQueries.jl>

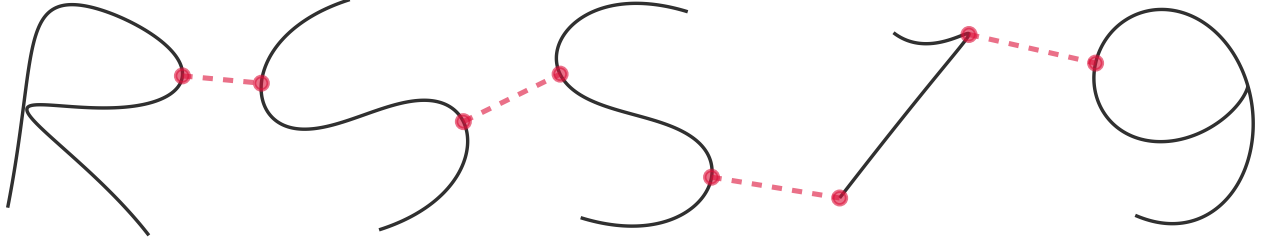


Figure 2.8: Points closest between pairs of characters 'R', 'S', 'S', '1', and '9' respectively, that are represented as Bézier curves. The evaluation time for the four minimum separating distance queries was 1.14 ms.

or  $\underline{d}$  violate the  $\Delta$ -tolerance, *i.e.* all the convex hulls formed from the curve evaluated at the sub-intervals in collection  $\mathcal{L}$  are separated from the object  $\mathcal{B}$  by a distance of at least  $\Delta$ . Algorithm 3 highlights the differences from Algorithm 1.

---

**Algorithm 3** Tolerance Verification ( $\Psi_{\mathcal{I}}, B, \Delta$ )

---

```

    ⋮
4: while  $\bar{d} - \Delta > \epsilon$  do
    ⋮
11:   if  $\underline{d} > \Delta$  then return true
12: end while
13: return false

```

---

### 2.4.3 Collision Detection

Sometimes only the detection of intersection between two objects is required. This is a special case of Algorithm 3, where  $\Delta = 0$ . Algorithm 3 highlights the differences from Algorithm 1.

---

**Algorithm 4** Collision Detection ( $\Psi_{\mathcal{I}}, B$ )

---

```

    ⋮
4: while  $\bar{d} > \epsilon$  do
    ⋮
11:   if  $\underline{d} > 0$  then return false
12: end while
13: return true

```

---

## 2.5 Numerical Results

We present numerical simulations of our approach for several different proximity query problems. In general, the classes of examples that we consider are categorized as the proximity query between a parametric curve and a point, a convex polygon, and another parametric curve. In these numerical problems we only consider proximity queries in  $\mathbb{R}^2$  for ease of viewing, however, the algorithm will hold in general as long as there exist methods to compute  $d_{lb}$  and  $d_{ub}$  in the given dimension. Examples of proximity queries for parametric curves in  $\mathbb{R}^3$  can be found in the online repository<sup>3</sup>. In every problem setting, the execution times for computing the minimum separating distance, tolerance verification and collision detection between two objects are benchmarked. Each benchmark result is computed as the median over 10,000 trials of the program. The implementation<sup>4</sup> uses double-precision arithmetic, and we chose the tolerance of  $\epsilon = 10^{-10}$  to be used in the optimization procedure. For the tolerance verification queries presented in this section, we choose the value of  $\Delta$  as half the minimum separating distance between the objects which they are querying.

### 2.5.1 Curve - Point Proximity

We show the performance of computing proximity queries between Bézier curves and a point in  $\mathbb{R}^2$  using our approaches and the approach found in [36]. The control points of the curve are uniformly randomly placed in  $[0, 1] \times [0, 1]$ , and the point to which the proximity is computed is also randomly placed in the same unit square. Figure 2.9 shows a few examples of the problems that were considered.

The numerical simulations compute the mean execution time over a range of different order Bézier curves as shown in Fig. 2.10. Notice that the computational efficiency (Fig. 2.10a) of Algorithm 1 is improved compared to that of the curve subdivision algorithm in [36]. This is in part because the construction of the bounding region in [36] is the control polygon of the subdivided curve that is obtained through the expensive De Casteljau’s algorithm [69]. Furthermore, the parameterizations of the subdivided curves are stored in a queue, which is very memory expensive for higher order curves as seen in Fig. 2.10b. On the other hand, our methods are very memory efficient as the priority queue (expressed as the collection  $\mathcal{L}$  in Algorithm 1 only contains information of the intervals. Additionally, computing proximity queries for Bézier curves using our approach is numerically robust as (2.6) is also a Bézier curve and does not require any change of basis. Approaches to compute the closed-form expression of (2.6) can be found in [70].

---

<sup>4</sup>The algorithms are implemented in Julia [68], and the benchmarks were conducted on a 2.3 GHz Intel Core i5 machine with 8 Gigabytes of RAM.

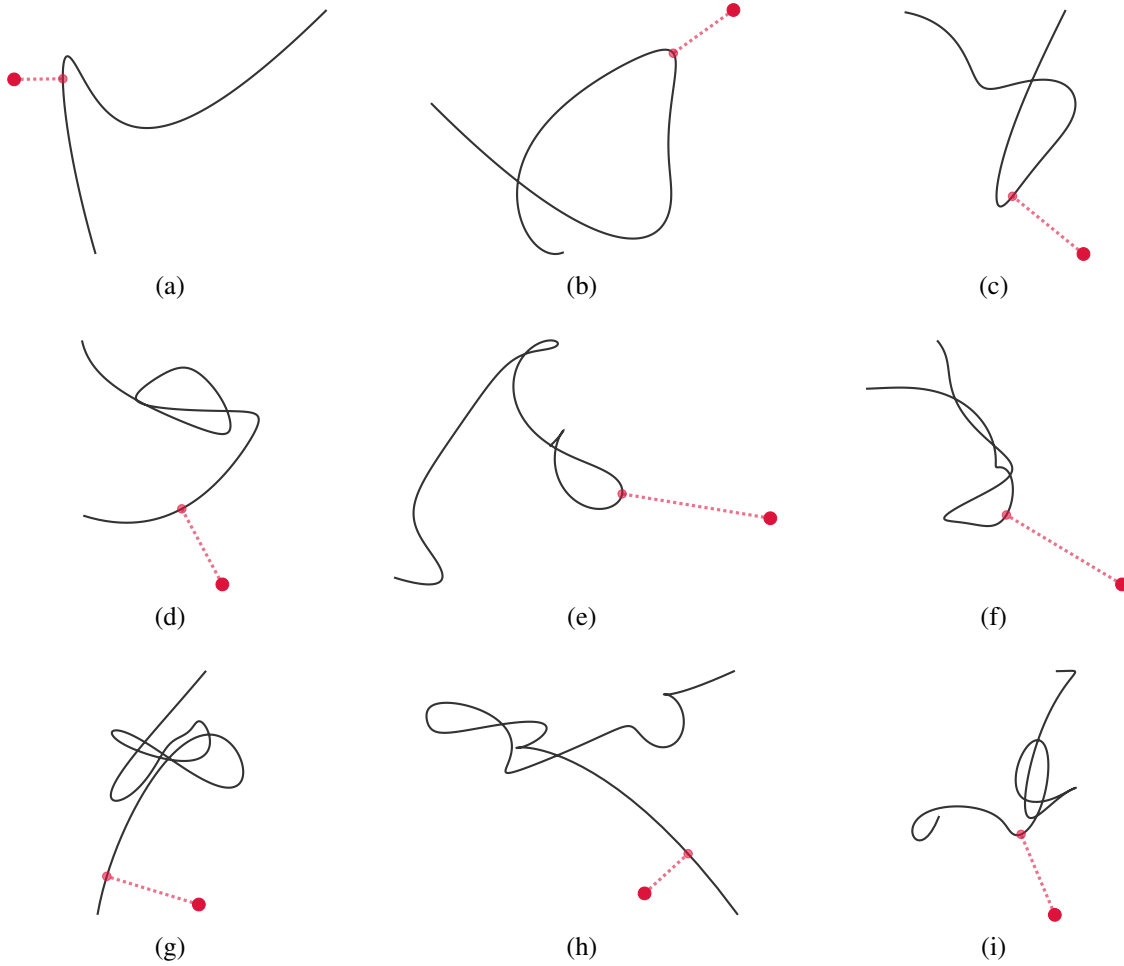


Figure 2.9: The point on a (a) 5<sup>th</sup>, (b) 10<sup>th</sup>, (c) 15<sup>th</sup>, (d) 20<sup>th</sup>, (e) 25<sup>th</sup>, (f) 30<sup>th</sup>, (g) 35<sup>th</sup>, (h) 40<sup>th</sup>, and (i) 45<sup>th</sup> Bézier curve that is closest to another randomly chosen point.

Both Algorithms 3 and 4 significantly outperform the other methods. This is a result of the fact that the algorithms are terminated as soon as the bounds are met on the  $\Delta$ -tolerance, rather than proceeding to obtain an  $\epsilon$ -suboptimal solution.

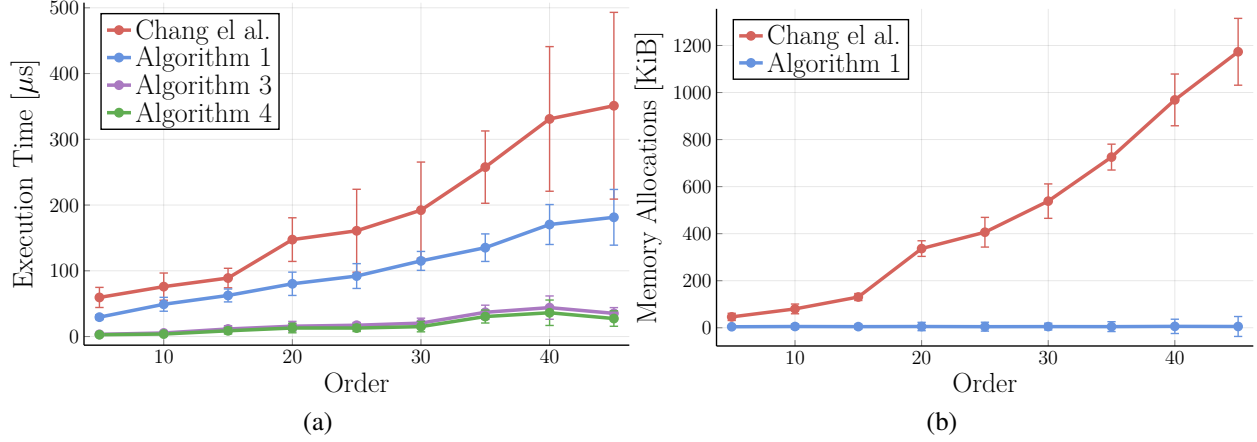


Figure 2.10: The average computational time (a) and allocated memory (b) to evaluate a proximity query between ten randomly generated Bézier curves and a point. The algorithm presented in [36] and Algorithm 1 both use a tolerance  $\epsilon = 10^{-10}$ . The error bars represent the standard deviation of the trial runs.

Query	Median Time ( $\mu$ s)					
	(a)	(b)	(c)	(d)	(e)	(f)
Alg. 1/2	208.22	234.52	$150.03 \times 10^3$	425.66	$1.21 \times 10^3$	$3.74 \times 10^3$
Alg. 3	10.52	80.90	$29.70 \times 10^3$	49.20	204.68	471.99
Alg. 4	10.38	59.19	$17.08 \times 10^3$	37.66	116.22	229.25

Table 2.2: The median computational time to evaluate a proximity query between each of the objects in Fig. 2.11.

### 2.5.2 Curve - Convex Polygon Proximity

We proceed to discuss the performance of proximity queries between parametric curves and polygons through some examples in Fig. 2.11. We consider the general class of absolutely continuous parametric curves, and, since not all the curves can be transformed<sup>5</sup> to a Bernstein basis, we do not provide numerical comparisons to [36] here. For the underlying routines that compute the bounds on the solution, *i.e.*  $d_{lb}$  and  $d_{ub}$ , we use the GJK-algorithm [45] to compute the distances between

<sup>5</sup>Even transformations between polynomial bases are numerically unstable for higher order polynomials [74].

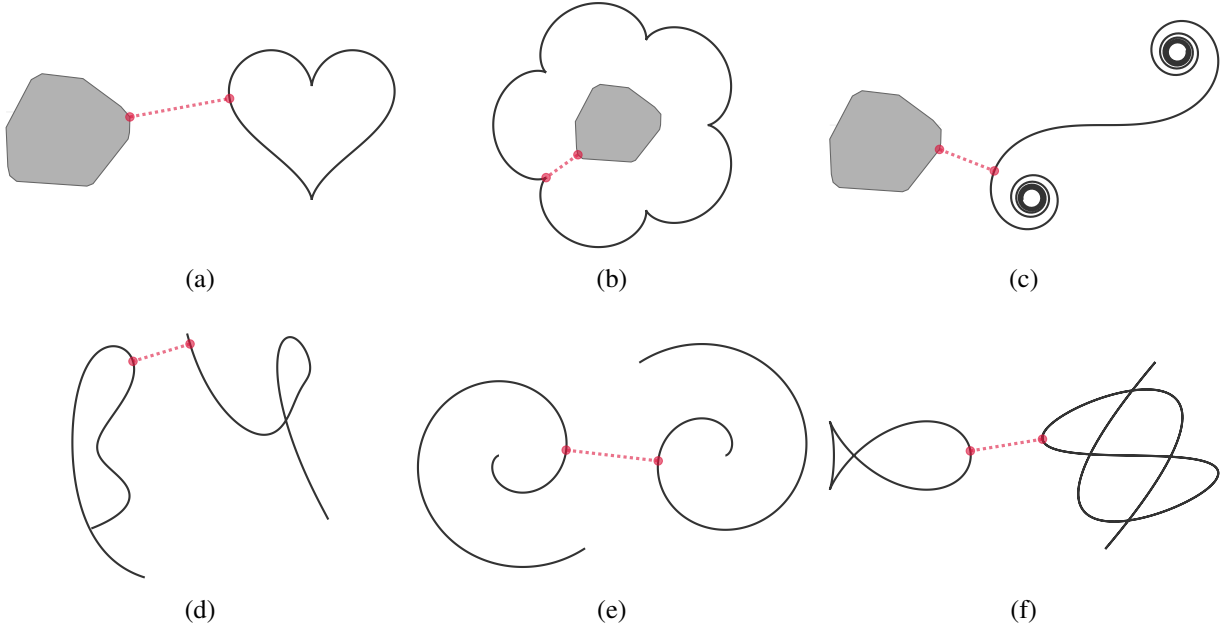


Figure 2.11: Points closest between (a) a convex polygon and a heart-shaped curve [71], (b) a convex polygon and an epicycloid [72], (c) a convex polygon and an Euler spiral, (d) two 10<sup>th</sup> order Bézier curves, (e) two involutes of a circle, and (f) a fish curve [73] and a Lissajous curve.

the polygons and the ellipsoidal convex hulls. Figures 2.11a to 2.11c show proximity of curves with a sinusoidal basis with convex polygons. It should be noted that although the Euler spiral is defined over  $\mathbb{R}$ , we consider its trace Fig. 2.11c in the compact subset:  $[-2\pi, 2\pi]$ . In Table 2.2, the median execution times for the different problems are enumerated. It should be highlighted that the reason for the longer run times of the Euler spiral is due to the evaluation of the parametric function that requires numerical integration.

### 2.5.3 Curve - Curve Proximity

Similar to above, Table 2.2 also shows the performance for computing the proximity queries for the problems in Figs. 2.11d to 2.11f. As these problems require simultaneously searching across dual parameter spaces, it is natural that the proximity queries will have longer run times. This will be offset if a lower  $\epsilon$ -tolerance is chosen in the procedures.

### 2.5.4 Trajectory Replanning

To demonstrate the benefit of fast collision detection, Figure 2.12 shows a common situation in path replanning problems. A vehicle is confronted with an obstacle and randomly samples a large

number of possible trajectories from a distribution to plan a path around the obstacle [75]. For each of these trajectories, a certificate of collision avoidance and minimum safety distance are necessary. In this situation, 1000 quintic Bézier curves are generated and, using Algorithm 3, a tolerance verification query is performed against both the obstacles with a total run time of 11.61 milliseconds. In the simulation, 818 trajectories of the 1000 samples are in collision with at least one of the two obstacles, 116 trajectories are collision free but violate the minimum safety distance constraint, and only the remaining 66 trajectories are feasible. This ability to rapidly compute the feasibility of trajectories allow for a large sample size of trajectories to be validated in a very short amount of time. In addition, these methods prove beneficial for predicting collisions in dynamic environments wherein only probabilistic information of the obstacle behavior is available [76] by computing proximity queries with the boundary of the confidence region of an obstacle's trajectory.

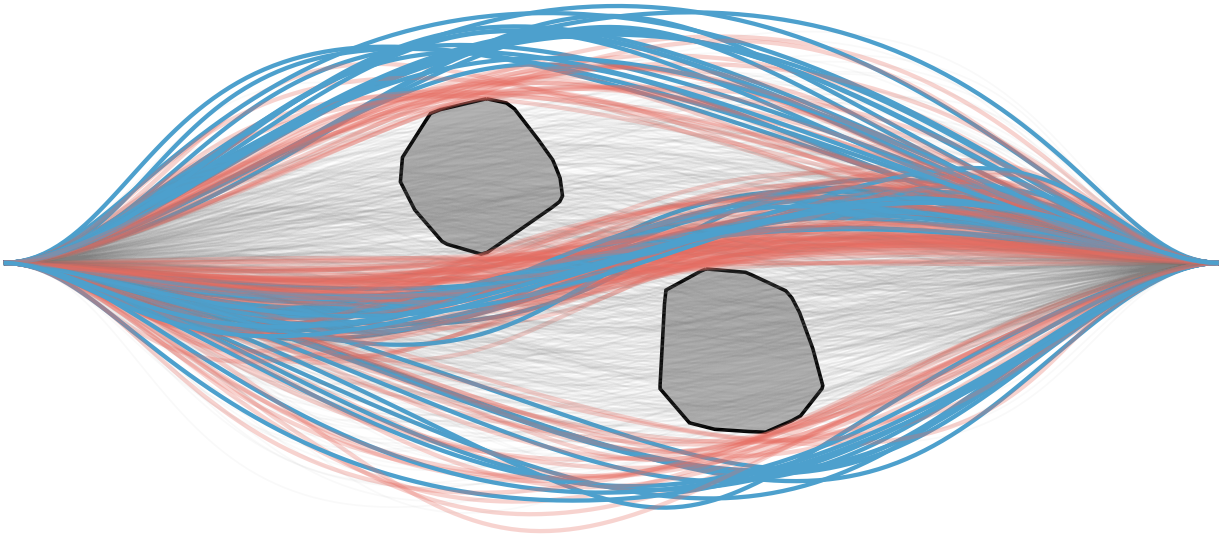


Figure 2.12: An example scenario of a robot (starting on the left) attempting to replan its trajectory around obstacles: The blue trajectories represent feasible solutions, the red trajectories are collision-free, but infeasible because of their close proximity to the obstacles (*i.e.*  $\leq \Delta$ ), and the grey trajectories are infeasible, because they collide with the obstacles in the environment.

# Chapter 3

## Safe Feedback Motion Planning for Uncertain Systems

### 3.1 Introduction

Motion planning algorithms generate optimal open-loop trajectories for robots to follow; however, any uncertainty in the system can potentially drive the robot far away from the desired path. For instance, quadrotors experience blade-flapping and induced drag forces that are dependent on the velocity, ground effects that are dependent on the altitude, and external wind effects that are often unaccounted for by the motion planner, [2]. Accurate modeling of these uncertainty effects on system dynamics can be very expensive and time-consuming. A widely accepted approach to account for uncertainty in motion planning is through feedback [28, Chapter 8]. In practice, ancillary tracking controllers or model predictive control (MPC) schemes are employed to alleviate this problem. However, the presence of the uncertainties is not explicitly considered in the control design process, and instead the performance is achieved with hand-tuned controller parameters and experimental validation. Without valid safety certificates, the uncertainty might drive the system unstable and far enough away from the desired trajectory, resulting in collisions with obstacles, Fig. 3.1a.

Robust trajectory tracking controllers using classical Lyapunov stability theory have been designed for helicopters [77], hovercraft [78], marine vehicles [79], and several other autonomous robots, which exhibit nonlinear behavior. These approaches rely on backstepping techniques, sliding-mode control, passivity-based control, or other robust nonlinear control design tools [80, Chapter 14]. However, the classical methods do not provide a ‘one size fits all’ procedure for the constructive design of tracking controllers for a large class of nonlinear systems. Unless the problem has a very specific structure that can be exploited, a control Lyapunov function (CLF) has to be found which can be prohibitively difficult for general nonlinear systems because the feasibility conditions do not appear as linear matrix inequalities (LMIs), unlike in case of linear systems.

Advances in computational resources and optimization toolboxes available to autonomous robots have led to active developments in the field of robust MPC. The two large classes of methods of interest are min-max MPC [81–83] and tube-based MPC [10, 84–87]. Min-max MPC approaches



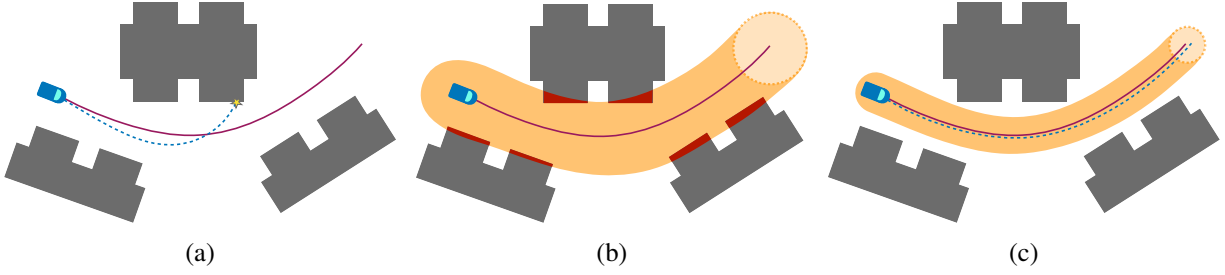


Figure 3.1: Although the planned path is collision free (purple), the robot’s actual trajectory (dashed-blue) might lead to a collision with the obstacles (gray) in the environment due to model discrepancies or external disturbances. (b) A feedback policy ensures that the robot stays inside of the (orange) tube which is too wide to pass between the obstacles without colliding (c) The safe feedback controller proposed in this dissertation guarantees that the robot’s trajectory never escapes the tube, which itself is also collision-free.

consider the worst-case disturbance that can affect the system making them overly conservative. If the uncertainty is too large or the robot is planning over a long horizon, a min-max MPC based approach may even render the optimization infeasible. Tube-based MPC methods address these issues by employing an ancillary controller to attenuate disturbances and ensure that the robot stays inside of a ‘tube’ around the desired trajectory. However, with the exception of [87], these methods assume the existence of a stabilizing ancillary controller and its region of attraction along the desired trajectory. Moreover, the resulting tubes are of fixed width, which may be overly conservative depending on the operating conditions (see Fig. 3.1b). This issue is partly addressed for feedback linearizable systems in [87] by using sliding-mode boundary layer control to construct tubes of any desired size during the MPC optimization procedure. Furthermore, unlike classical methods, the MPC-based approaches while applicable to larger class of systems incur a heavy computational load and are not always amenable to real-time applications. Another class of corrective methods that ensure safety is based on control barrier functions (CBFs) [88–91] rely on specialized functions that ensure set invariance which prevent the system states from reaching unsafe regions. However, CBFs do not provide tracking error bounds with respect to a desired trajectory which is critical in evaluating the robot’s performance. Moreover, these approaches require discovering a CBF and an ancillary controller which are non-trivial to synthesize. In our approach, we provide an explicit design for the feedback controller with stability and performance guarantees.

Contraction theory-based approaches [92, 93] bridge the gap between classical and optimization-based methods, and provide a constructive control design procedure for nonlinear systems. In [94], the authors introduce contraction analysis as tool for studying stability of nonlinear systems using differential geometry. In particular, the authors show that the ‘contracting’ or convergent nature of solutions to nonlinear systems can be derived from the differential dynamics of the system.

Since the differential dynamics for nonlinear systems are of linear-time varying form, all the results from linear systems theory can be leveraged for nonlinear systems through the contraction analysis framework. In [92], constructive control design techniques from linear systems theory can be used to find a control contraction metric (CCM), which is analogous to CLFs in the differential framework. This is significantly easier than directly finding the CLFs for nonlinear systems, because the feasibility conditions for CCMs are represented as LMIs. In [95], a design procedure for synthesizing CCM-based controllers is given, which induces fixed-width tubes in the presence of bounded external disturbances, excluding modeling uncertainties. However, as discussed before, fixed-width tubes might result in infeasibility of the problem and result in more work for the planner to find a more conservative path that produces feasible tubes. Recently, our work in [96] alleviates this issue by synthesizing CCMs that minimize the differential  $\mathcal{L}_\infty$  gain which results in tighter tubes around the desired trajectory. In, [97] a model reference control architecture in conjunction with CCM-based feedback is proposed for handling uncertainties in the system.

In this work, we present an approach for safe feedback motion planning for control-affine nonlinear systems that relies on contraction theory-based solution for exponential stabilizability around trajectories and  $\mathcal{L}_1$ -adaptive control for handling uncertainties and providing guarantees for transient performance and robustness. In  $\mathcal{L}_1$  control architecture, estimation is decoupled from control, thereby allowing for arbitrarily fast adaptation subject only to hardware limitations, [15]. The  $\mathcal{L}_1$  control has been successfully implemented on NASA's AirStar 5.5% subscale generic transport aircraft model [98], Calspan's Learjet [99], and unmanned aerial vehicles [100–104]. In this dissertation, we present a constructive design of feedback strategy for nonlinear systems using CCMs and  $\mathcal{L}_1$ -adaptive control that provides strong guarantees of transient performance and robustness for control-affine nonlinear systems. Furthermore, we show how this control architecture induces tubes that can be flexibly changed to ensure safety based on the uncertainty in the system and the environment. In particular, this flexibility is provided by the architecture of the  $\mathcal{L}_1$ -adaptive control by decoupling the control loop from the estimation loop [15]. In this way, the width of the certifiable tubes can be adjusted allowing the safe operation of a robot in tight confines.

## 3.2 Problem Formulation

We consider systems for which the evolution of dynamics can be represented as

$$\dot{x}(t) = F(x(t), u(t)) \tag{3.1a}$$

$$= f(x(t)) + B(x(t))(u(t) + h(t, x(t))), \tag{3.1b}$$

with initial condition  $x(0) = x_0$ , where  $x(t) \in \mathbb{R}^n$  is the system state and  $u(t) \in \mathbb{R}^m$  is the control input. The functions  $f(x) \in \mathbb{R}^n$  and  $B(x) \in \mathbb{R}^{n \times m}$  are known, and  $h(t, x) \in \mathbb{R}^m$  represents the uncertainties. The unperturbed/nominal dynamics ( $h \equiv 0$ ) are therefore represented as

$$\dot{x}(t) = \bar{F}(x(t), u(t)) \quad (3.2a)$$

$$= f(x(t)) + B(x(t))u(t), \quad x(0) = x_0. \quad (3.2b)$$

Consider a desired control trajectory  $u^*(t) \in \mathbb{R}^m$  and the induced desired state trajectory  $x^*(t) \in \mathbb{R}^n$  from any planner based on unperturbed/nominal dynamics

$$\dot{x}^*(t) = \bar{F}(x^*(t), u^*(t)), \quad x^*(0) = x_0^*. \quad (3.3)$$

Together,  $(x^*(t), u^*(t))$  is referred to as the desired state-input trajectory pair. The planner ensures that the desired state-trajectory  $x^*(t)$  remains in a compact safe set  $\mathcal{X} \subset \mathbb{R}^n$ , for all  $t \geq 0$ .

The goal is to design a control input  $u(t)$  so that the state  $x(t)$  of the uncertain system in (3.1) remains ‘close’ to the desired trajectory  $x^*(t)$  while also ensuring  $x(t) \in \mathcal{X}$ , for all  $t \geq 0$ . In order to rigorously define the notion of ‘closeness’, we need the following definition:

**Definition 3.1.** *Given a positive scalar  $\rho$  and the desired state trajectory  $x^*(t)$ ,  $\Omega(\rho, x^*(t))$  denotes the  $\rho$ -norm ball around  $x^*(t)$ , i.e.*

$$\Omega(\rho, x^*(t)) := \{y \in \mathbb{R}^n \mid \|y - x^*(t)\| \leq \rho\}. \quad (3.4)$$

Clearly  $\Omega(\rho, x^*(t))$  induces a tube centered around  $x^*(t)$ , where the tube is given by

$$\mathcal{O}(\rho) := \bigcup_{t \geq 0} \Omega(\rho, x^*(t)), \quad (3.5)$$

with  $\rho > 0$  as the radius.

The problem under consideration can now be stated as follows: Given the desired trajectory  $x^*(t) \in \mathcal{X}$  and a positive scalar  $\rho$ , design a control input  $u(t)$  such that the state of the uncertain system (3.1) satisfies:

$$x(t) \in \Omega(\rho, x^*(t)) \subset \mathcal{X}, \quad \forall t \geq 0.$$

Note the condition that  $\Omega(\rho, x^*(t)) \subset \mathcal{X}$  is dependent on the desired trajectory  $x^*(t)$  (given by the planner) and the tube width  $\rho$  (chosen by the user). To ensure that this control-independent condition is satisfied, we place the following assumption.

**Assumption 3.1.** Given the positive scalar  $\rho$ , the desired state trajectory satisfies  $x^*(t) \in \mathcal{X}_\rho$ , for all  $t \geq 0$ , where

$$\mathcal{X}_\rho = \mathcal{X} \ominus \mathcal{B}(\rho), \quad \mathcal{B}(\rho) := \{y \in \mathbb{R}^n \mid \|y\| \leq \rho\}. \quad (3.6)$$

**Remark 3.1.** The implication of Assumption 3.1 is that if the state trajectory satisfies  $x(t) - x^*(t) \in \mathcal{B}(\rho)$  and  $x^*(t) \in \mathcal{X}_\rho$ , for all  $t \geq 0$ , then the definition of the Pontryagin set difference implies that  $x(t) \in \Omega(\rho, x^*(t)) \subset \mathcal{X}$ , for all  $t \geq 0$ .

**Assumption 3.2.** The desired control/input trajectory satisfies

$$\|u^*(t)\| \leq \Delta_{u^*}, \quad \forall t \geq 0,$$

with the upper bound  $\Delta_{u^*}$  known.

Note that the bound  $\Delta_{u^*}$  is obtained from the planner, which provides the desired state-input trajectory in (3.3). Next, we place assumptions on the boundedness and continuity properties of the system functions and uncertainties.

**Assumption 3.3.** The known functions  $f(x) \in \mathbb{R}^n$  and  $B(x) \in \mathbb{R}^{n \times m}$  are bounded and continuously differentiable with bounded derivatives, satisfying

$$\|f(x)\| \leq \Delta_f, \quad \left\| \frac{\partial f(x)}{\partial x} \right\| \leq \Delta_{f_x}, \quad \|B(x)\| \leq \Delta_B, \quad \sum_{i=1}^n \left\| \frac{\partial B(x)}{\partial x_i} \right\| \leq \Delta_{B_x}, \quad \sum_{j=1}^m \left\| \frac{\partial b_j(x)}{\partial x} \right\| \leq \Delta_{b_x}$$

for all  $x \in \mathcal{O}(\rho)$ , where  $b_j(x)$  is the  $j^{\text{th}}$  column of  $B(x)$  and the bounds are assumed to be known.

**Assumption 3.4.** The uncertainty  $h(t, x)$  is bounded and continuously differentiable in both  $x$  and  $t$  with bounded derivatives, satisfying

$$\|h(t, x)\| \leq \Delta_h, \quad \left\| \frac{\partial h(t, x)}{\partial x} \right\| \leq \Delta_{h_x}, \quad \left\| \frac{\partial h(t, x)}{\partial t} \right\| \leq \Delta_{h_t},$$

for all  $x \in \mathcal{O}(\rho)$  and  $t \geq 0$ , where the bounds are assumed to be known.

**Assumption 3.5.** The input gain matrix  $B(x)$  has full column rank. Furthermore, the Moore-Penrose inverse of  $B(x)$  defined as  $B^\dagger(x) = (B^\top(x)B(x))^{-1} B^\top(x)$  satisfies the following bounds

$$\|B^\dagger(x)\| \leq \Delta_{B^\dagger}, \quad \sum_{i=1}^n \left\| \frac{\partial B^\dagger(x)}{\partial x_i} \right\| \leq \Delta_{B_x^\dagger}, \quad \forall x \in \mathcal{O}(\rho).$$

### 3.3 Preliminaries on Contraction Theory

Contraction theory allows to synthesize feedback laws so that, in the absence of uncertainties, the state of the unperturbed/nominal dynamics in (3.2) tracks a feasible desired trajectory  $x^*(t)$ . We begin with the notion of incremental exponential stability (IES).

**Definition 3.2** ([95]). *Consider a desired state-input trajectory pair  $(x^*(t), u^*(t))$  satisfying (3.3). Suppose there exist scalars  $\lambda, R > 0$  and a feedback operator  $k_c : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^m$  can be constructed such that the trajectory  $x(t)$  of the unperturbed dynamics  $\dot{x}(t) = \bar{F}(x(t), u_c(t))$  with control  $u_c(t) = u^*(t) + k_c(x^*(t), x(t))$  satisfies*

$$\|x^*(t) - x(t)\| \leq R e^{-\lambda t} \|x^*(0) - x(0)\|, \quad \forall t \geq 0.$$

*Then, the system with the unperturbed dynamics is said to be IES with rate  $\lambda$  and overshoot  $R$ .*

With the notion of IES defined, we now proceed to examine how IES may be established for a given system. For the compact safe set  $\mathcal{X} \subset \mathcal{R}^n$  defined in Section 3.2, let  $T_x \mathcal{X}$  be the tangent space of  $\mathcal{X}$  at  $x \in \mathcal{X}$ . Consequently, we denote by  $T\mathcal{X} = \dot{\bigcup}_{x \in \mathcal{X}} T_x \mathcal{X}$  the tangent bundle of  $\mathcal{X}$ , where  $\dot{\bigcup}$  denotes the disjoint union. Details on differential geometric notions used in the manuscript may be found in [105]. The variational dynamics of the unperturbed/nominal system in (3.2) may be written as [106, Chapter 3]

$$\dot{\delta}_x = \left( \frac{\partial f(x)}{\partial x} + \sum_{j=1}^m u[j] \frac{\partial b_j(x)}{\partial x} \right) \delta_x + B(x) \delta_u, \quad (3.7)$$

with  $\delta_x(0) = x_0$ , where we have dropped the temporal dependencies for brevity. Here,  $\delta_x(t) \in T_{x(t)} \mathcal{X}$ ,  $\delta_u(t) \in T_{u(t)} \mathbb{R}^m$ ,  $u[j](t)$  is the  $j^{\text{th}}$  element of the control vector and  $b_j(x) \in \mathbb{R}^n$  is the  $j^{\text{th}}$  column of  $B(x)$ .

**Definition 3.3.** *Consider the differential dynamics in (3.7). Suppose there exist positive scalars  $\lambda, \underline{\alpha}, \bar{\alpha}$ ,  $0 < \underline{\alpha} < \bar{\alpha} < \infty$ , and a smooth<sup>1</sup> function  $M : \mathbb{R}^n \rightarrow \mathbb{S}^n$  such that for all  $(x, \delta_x) \in T\mathcal{X}$  one has*

$$\underline{\alpha} \mathbb{I}_n \preceq M(x) \preceq \bar{\alpha} \mathbb{I}_n, \quad (3.8a)$$

$$\delta_x^\top M(x) B(x) = 0 \Rightarrow \delta_x^\top \left( \partial_f M(x) + \left[ M(x) \frac{\partial f(x)}{\partial x} \right]_{\mathbb{S}} + 2\lambda M(x) \right) \delta_x \leq 0, \quad (3.8b)$$

$$\partial_{b_j} M(x) + \left[ M(x) \frac{\partial b_j(x)}{\partial x} \right]_{\mathbb{S}} = 0, \quad j \in \{1, \dots, m\}. \quad (3.8c)$$

---

<sup>1</sup>Throughout the manuscript, by smooth we mean the class  $\mathcal{C}^\infty$  of functions defined on appropriate domains.

Then, the function  $M(x)$  is defined to be the CCM for the nominal/unperturbed dynamics (3.2).

**Theorem 3.1** ([92, 95]). *Given positive scalars  $\lambda$ , and  $\underline{\alpha} \leq \bar{\alpha} < \infty$ , suppose there exists a CCM  $M(x)$  for the nominal/unperturbed dynamics in (3.2). Then, given any desired state-input trajectory  $(x^*(t), u^*(t))$  as in (3.3), there exists a feedback operator  $k_c : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^m$  such that the trajectory  $x(t)$  of the unperturbed dynamics  $\dot{x}(t) = \bar{F}(x(t), u_c(t))$  with control  $u(t) = u^*(t) + k_c(x^*(t), x(t))$  is IES with respect to  $x^*(t)$  with the overshoot of  $R = \bar{\alpha}/\underline{\alpha}$  in the sense of Definition 3.2.*

The central idea to this result is that the function  $V(x, \delta_x) := \delta_x^\top M(x) \delta_x$  can be interpreted as a differential Lyapunov function (LF) and the conditions in (3.8) ensure that  $\dot{V}(x, \delta_x) \leq -2\lambda V(x, \delta_x)$  for all  $(x, \delta_x) \in T\mathcal{X}$ . We place the following assumption on the known/unperturbed dynamics.

**Assumption 3.6.** *The nominal/unperturbed dynamics in (3.2) admit a CCM  $M(x)$  for all  $x \in \mathcal{X}$  with positive scalars  $\lambda$ ,  $\underline{\alpha}$ , and  $\bar{\alpha}$ , as in Definition 3.3.*

Using Theorem 3.1 it is straightforward to conclude that the consequence of this assumption is that any desired state-input trajectory can be tracked by the nominal/unperturbed dynamics in the sense of Definition 3.2 with rate  $\lambda$  and overshoot  $R = \bar{\alpha}/\underline{\alpha}$ . Let  $\Xi(p, q)$  be the set of smooth curves connecting any two points  $p, q \in \mathcal{X}$ . Then using the Riemannian metric  $M$ , the length of any curve  $\gamma \in \Xi(p, q)$  is given by the following expression

$$l(\gamma) := \int_0^1 \sqrt{\gamma_s(s) M(\gamma(s)) \gamma_s(s)} ds, \quad (3.9)$$

where  $\gamma_s(s) = \partial\gamma(s)/\partial s$ . By definition, the minimizing geodesic  $\bar{\gamma} : [0, 1] \rightarrow \mathcal{X}$  satisfies the following relationship

$$d(p, q) := l(\bar{\gamma}) = \inf_{\gamma \in \Xi(p, q)} l(\gamma), \quad (3.10)$$

where  $d(p, q)$  refers to the Riemannian distance between the two points  $p$  and  $q$ . Existence of the minimizing geodesic is guaranteed by the Hopf-Rinow theorem. The Riemannian energy between the two points is defined using the Riemannian distance as the following quantity

$$\mathcal{E}(p, q) := d(p, q)^2. \quad (3.11)$$

Further details on Riemannian geometry may be found in [107]. A direct and straightforward consequence of Assumption 3.6 is that

$$\underline{\alpha} \|p - q\|^2 \leq \mathcal{E}(p, q) \leq \bar{\alpha} \|p - q\|^2, \quad \forall p, q \in \mathcal{X}. \quad (3.12)$$

The proof for this relationship can be found in Lemma B.3. We will rely on the Riemannian energy's interpretation as a CLF for the presented methodology. This interpretation was initially presented in [92].

### 3.4 $\mathcal{CL}_1$ -Adaptive Control

In this section we introduce the structure of the proposed contraction theory-based  $\mathcal{L}_1$ -adaptive controller, which we refer to as  $\mathcal{CL}_1$ -adaptive control, for the uncertain nonlinear system in (3.1). Consider the following feedback decomposition

$$u(t) = u_c(t) + u_a(t), \quad (3.13)$$

where  $u_c : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$  is the contraction theory based control designed to guarantee IES (Definition 3.2) of the nominal dynamics in (3.2), and  $u_a : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$  is the  $\mathcal{L}_1$  control signal. The overall architecture of the proposed feedback is illustrated in Fig. 3.2. We refer to the uncertain system in (3.1) with the feedback law (3.13) as the  $\mathcal{L}_1$  closed-loop system. Before we proceed with the description of the individual components of the controller, we introduce the following list of constants that are of importance for the results and analysis presented in this chapter:

$$\Delta_{M_x} := \sup_{x \in \mathcal{O}(\rho)} \sum_{i=1}^n \left\| \frac{\partial M}{\partial x_i}(x) \right\|, \quad (3.14)$$

$$\Delta_{\Psi_x} := 2\Delta_{B_x} + \frac{\Delta_B \Delta_{M_x}}{\underline{\alpha}}, \quad (3.15)$$

$$\Delta_{\delta_u} := \frac{1}{2} \sup_{x \in \mathcal{O}(\rho)} \left( \frac{\bar{\lambda}(L^{-\top}(x)F(x)L^{-1}(x))}{\underline{\sigma}_{>0}(B^\top(x)L^{-1}(x))} \right), \quad (3.16)$$

$$\Delta_{\dot{x}_r} := \Delta_f + \Delta_B(\|\mathbb{I}_m - C(s)\|_{\mathcal{L}_1} \Delta_h + \Delta_{u^*} + \rho \Delta_{\delta_u}), \quad (3.17)$$

$$\Delta_{\dot{x}} := \Delta_f + \Delta_B(2\Delta_h + \Delta_{u^*} + \rho \Delta_{\delta_u}), \quad (3.18)$$

$$\Delta_{\tilde{x}} := \sqrt{\frac{4\bar{\lambda}(P)\Delta_h(\Delta_{h_t} + \Delta_{h_x}\Delta_{\dot{x}})}{\underline{\lambda}(P)\underline{\lambda}(Q)} + \frac{4\Delta_h^2}{\underline{\lambda}(P)}}, \quad (3.19)$$

$$\Delta_{\tilde{\eta}} := \left( \Delta_{B_x^\dagger} \Delta_{\dot{x}} + (\|sC(s)\|_{\mathcal{L}_1} + \|A_m\|) \Delta_{B^\dagger} \right) \Delta_{\tilde{x}}, \quad (3.20)$$

$$\Delta_{\theta} := \frac{\Delta_B \bar{\alpha} \Delta_{\tilde{\eta}}}{\lambda}, \quad (3.21)$$

$$\Delta_{\dot{\Psi}} := \bar{\alpha} \left( \Delta_B \Delta_{\dot{\gamma}_s} + \frac{\Delta_B \Delta_{M_x} \Delta_{\dot{x}}}{\sqrt{\bar{\alpha} \underline{\alpha}}} + \Delta_{B_x} \Delta_{\dot{x}} \right), \quad (3.22)$$

$$\Delta_{\dot{\gamma}_s} := \sqrt{\frac{\bar{\alpha}}{\underline{\alpha}}} \left( \Delta_{f_x} + (\Delta_h + \Delta_{u^*} + \rho \Delta_{\delta_u}) \Delta_{b_x} + \left( \Delta_{h_x} + \frac{\sqrt{\bar{\alpha}} \Delta_{\delta_u}}{\sqrt{\underline{\alpha}}} \right) \Delta_B \right), \quad (3.23)$$

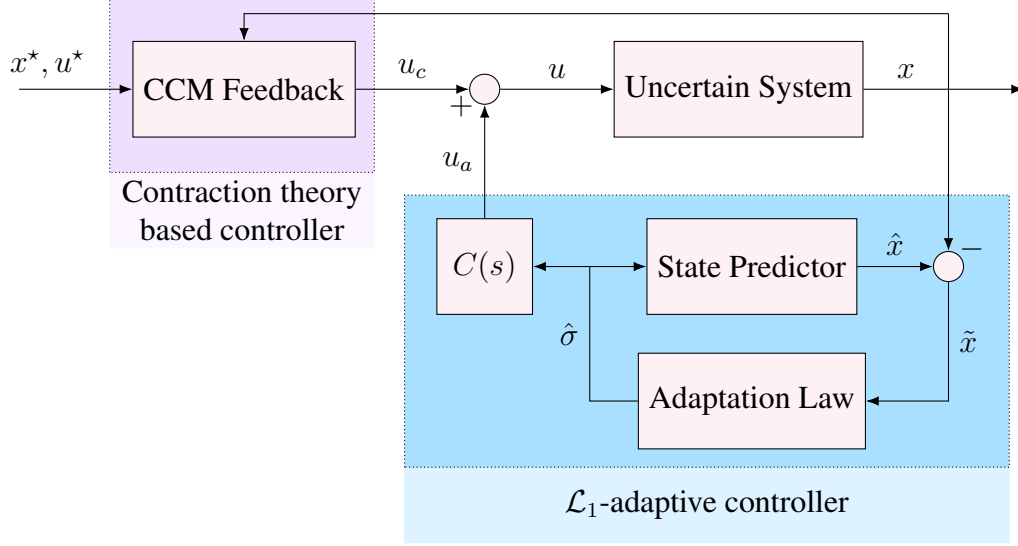


Figure 3.2: Architecture of  $\mathcal{CL}_1$ -adaptive control

where  $\mathcal{O}(\rho)$  is defined in (3.5);  $\Delta_{u^*}$  is defined in Assumption 3.2;  $\Delta_f, \Delta_{f_x}, \Delta_B, \Delta_{B_x}, \Delta_{b_x}$ , are defined in Assumption 3.3;  $\Delta_h, \Delta_{h_t}, \Delta_{h_x}$  are defined in Assumption 3.4;  $\Delta_{B^\dagger}$  and  $\Delta_{B_x^\dagger}$  are defined in Assumption 3.5;  $\bar{\alpha}$  and  $\underline{\alpha}$  are defined in Assumption 3.6; and  $F(x)$  is defined as

$$F(x) := -\partial_f W(x) + 2 \left[ \frac{\partial f}{\partial x}(x) W(x) \right]_{\mathbb{S}} + 2\lambda W(x),$$

where  $W(x) = M(x)^{-1}$  is referred to as the dual metric and  $L(x)^\top L(x) = W(x)$ .

### 3.4.1 Contraction Theory-Based Control: $u_c(t)$

As mentioned in Section 3.3, under Assumption 3.6, Theorem 3.1 guarantees the existence of a feedback law which renders the nominal dynamics in (3.2) IES. In particular, we propose the following law

$$u_c(t) = u^*(t) + k_c(x^*(t), x(t)), \quad (3.24)$$

where, for the feedback term, we use the law constructed in [95, Sec. 5.1], which is the solution to the following quadratic program:

$$k_c(x^*(t), x(t)) = \arg \min_{k \in \mathbb{R}^m} \|k\|^2, \quad (3.25a)$$

$$\text{s.t. } 2\bar{\gamma}_s^\top(1, t)M(x(t))\dot{x}_k(t) - 2\bar{\gamma}_s^\top(0, t)M(x^*(t))\dot{x}^*(t) \leq -2\lambda\mathcal{E}(x^*(t), x(t)), \quad (3.25b)$$



in which  $M(\cdot)$  is the CCM (Definition 3.3),  $\bar{\gamma}(s, t)$ ,  $s \in [0, 1]$ , is the minimizing geodesic with  $\bar{\gamma}(1, t) = x(t)$  and  $\bar{\gamma}(0, t) = x^*(t)$ . As previously defined, the desired state-input pair satisfies  $\dot{x}^*(t) = \bar{F}(x^*(t), u^*(t))$  with the nominal dynamics defined in (3.2). Additionally,  $\dot{x}_k(t) = \bar{F}(x(t), u^*(t) + k)$ .

**Remark 3.2.** *As explained by the authors in [95, Sec. 5.1], the solution to the quadratic program in (3.25) has an analytic solution given the minimizing geodesic  $\bar{\gamma}(\cdot, t)$ . Alternatively, one may use the differential controller proposed in [92].*

### 3.4.2 $\mathcal{L}_1$ -Adaptive Control: $u_a(t)$

The computation of the signal  $u_a(t)$  depends on three components illustrated in Fig. 3.2, namely, the state-predictor, the adaptation law, and a low-pass filter. Similar to [83], we define the state-predictor as

$$\dot{\hat{x}}(t) = \bar{F}(x(t), u_c(t) + u_a(t) + \hat{\sigma}(t)) + A_m \tilde{x}(t), \quad (3.26)$$

with  $\hat{x}(0) = x_0$ , and where  $\hat{x}(t) \in \mathbb{R}^n$  is the state of the predictor,  $\tilde{x}(t) = \hat{x}(t) - x(t)$  is the state prediction error, and  $A_m \in \mathbb{R}^{n \times n}$  is an arbitrary Hurwitz matrix.

The uncertainty estimate  $\hat{\sigma}(t)$  in (3.26) is governed by the following adaptation law

$$\dot{\hat{\sigma}}(t) = \Gamma \text{Proj}_{\mathcal{H}}(\hat{\sigma}(t), -B(x)^\top P \tilde{x}(t)), \quad \hat{\sigma}(0) \in \mathcal{H}, \quad (3.27)$$

where  $\Gamma > 0$  is the adaptation rate,  $\mathcal{H} = \{y \in \mathbb{R}^m \mid \|y\| \leq \Delta_h\}$  is the set to which the uncertainty estimate is restricted to remain in with  $\Delta_h$  defined in Assumption 3.4. Furthermore,  $\mathbb{S}^n \ni P \succ 0$ , is the solution to the Lyapunov equation  $A_m^\top P + P A_m = -Q$ , for some  $\mathbb{S}^n \ni Q \succ 0$ . Moreover,  $\text{Proj}_{\mathcal{H}}(\cdot, \cdot)$  is the projection operator standard in adaptive control literature [108], [109].

Finally, the control law  $u_a(t)$  is defined as the following Laplace transform

$$u_a(s) = -C(s)\hat{\sigma}(s), \quad (3.28)$$

where  $C(s)$  is a low-pass filter with bandwidth  $\omega$  and satisfies  $C(0) = \mathbb{I}_m$ . Note that there is an abuse of notation when we denote both the geodesic interval parameter and the Laplace variable by  $s$ . The delineation between the two is clear from the context.

### 3.4.3 Filter Bandwidth and Adaptation Rate

The design of the  $\mathcal{L}_1$ -adaptive controller involves the design of a strictly proper and stable low-pass filter  $C(s)$  with  $C(0) = \mathbb{I}_m$ . Let the bandwidth of this filter be  $\omega$ . In the thesis, for the sake of

simplicity, we choose  $C(s) = \frac{\omega}{s+\omega} \mathbb{I}_m$ . As we will see in Section 3.5, the bandwidth  $\omega$  of the low-pass filter  $C(s)$  in (3.28) and the adaptation rate  $\Gamma$  in (3.27) are design parameters which can be thought of as ‘tuning-knobs’. However, these entities need to satisfy a few conditions mentioned below. The reasoning behind these conditions will be made clear in the subsequent section.

Suppose that Assumption 3.6 holds. Then, for arbitrarily chosen positive scalars  $\epsilon$  and  $\rho_a$ , define

$$\rho_r = \sqrt{\frac{\bar{\alpha}}{\underline{\alpha}}} \|x_0^* - x_0\| + \epsilon, \quad (3.29)$$

$$\rho = \rho_r + \rho_a. \quad (3.30)$$

Furthermore, suppose that Assumptions 3.1 and 3.5 hold. Define

$$\zeta_1(\omega) = 2\rho\Delta_B \frac{\bar{\alpha}}{\underline{\alpha}} \left( \frac{\Delta_h}{|2\lambda - \omega|} + \frac{\Delta_{h_t} + \Delta_{h_x}\Delta_{\dot{x}_r}}{2\lambda\omega} \right), \quad (3.31a)$$

$$\zeta_2(\omega) = \bar{\alpha}\Delta_{\Psi_x} \left( \frac{\Delta_h}{|2\lambda - \omega|} + \frac{\Delta_{h_t} + \Delta_{h_x}\Delta_{\dot{x}_r}}{2\lambda\omega} \right), \quad (3.31b)$$

$$\zeta_3(\omega) = \bar{\alpha}\Delta_{h_x} \left( \frac{4\lambda\Delta_B + \Delta_{\Psi}}{\lambda\omega} \right), \quad (3.31c)$$

where  $\Delta_{\dot{x}_r}$ ,  $\Delta_{\Psi_x}$ , and  $\Delta_{\Psi}$ , are known positive scalars defined in (3.15), (3.17) and (3.22) respectively. Then, the bandwidth  $\omega$  of the low-pass filter  $C(s)$  and the adaptation rate need to verify the following conditions

$$\rho_r^2 \geq \frac{\mathcal{E}(x_0^*, x_0)}{\underline{\alpha}} + \zeta_1(\omega), \quad (3.32a)$$

$$\underline{\alpha} > \zeta_2(\omega) + \zeta_3(\omega), \quad (3.32b)$$

$$\sqrt{\Gamma} > \frac{\Delta_\theta}{\rho_a(\underline{\alpha} - \zeta_2(\omega) - \zeta_3(\omega))}, \quad (3.32c)$$

where  $\Delta_\theta$  is another known positive scalar defined in (3.21).

**Remark 3.3.** Based on the definition of  $\rho_r$  in (3.30) and the bounds on the Riemannian energy  $\mathcal{E}(x^*(t), x(t))$  in (3.12), the inequality  $\rho_r^2 > \mathcal{E}(x_0^*, x_0)/\underline{\alpha}$  holds. Furthermore, since  $\zeta_1(\omega)$ ,  $\zeta_2(\omega)$ , and  $\zeta_3(\omega)$ , all converge to zero as  $\omega$  increases, the bandwidth conditions in (3.32a)-(3.32b) can always be satisfied by choosing a large enough  $\omega$ . However, it is important to note that an increased bandwidth may result in the loss of robustness to model inaccuracies such as time-delays in the control architecture. Such inaccuracies are commonplace in real world scenarios and may justify an upper limit on the maximum bandwidth based on the hardware architecture and sensor design, limiting the ‘flexibility’ of the tube.

### 3.5 Performance Analysis

In this section we analyze the performance of the uncertain system in (3.1) with the  $\mathcal{L}_1$  control feedback  $u(t)$  defined in (3.13). As in [83], to derive the bounds between the desired trajectory  $x^*(t)$  and the state  $x(t)$  of the uncertain system, we first introduce the following intermediate system, which we refer to as the reference system:

$$\dot{x}_r(t) = F(x_r(t), -\eta_r(t)) = f(x_r(t)) + B(x_r(t))(u_{c,r}(t) - \eta_r(t) + h(t, x_r(t))), \quad (3.33a)$$

$$u_{c,r}(t) = u^*(t) + k_c(x^*(t), x_r(t)), \quad (3.33b)$$

$$\eta_r(s) = C(s)\mathcal{L}[h(t, x_r(t))], \quad x_r(0) = x_0, \quad (3.33c)$$

where  $k_c$  is defined in (3.25) using  $x_r$  in place of  $x$ . The main feature of the reference system is that it defines the *the best achievable performance*, given the perfect knowledge of uncertainty, i.e. it reflects that the cancellation of the uncertainty  $h(t, x_r(t))$  can happen only within the bandwidth of the low-pass filter.

The analysis consists of two parts: we first derive bounds between the desired trajectory and the reference system  $\|x^*(t) - x_r(t)\|$ . Then we derive the bounds between the states of the reference system and the actual system  $\|x_r(t) - x(t)\|$ . Recall that we refer to the actual system as the  $\mathcal{L}_1$  closed loop system, which is given by (3.1) with the control law in (3.13). Finally, the triangle inequality produces the desired bound on  $\|x^*(t) - x(t)\|$ . In this way, the reference system behaves as an ‘anchor system’ for the analysis. These bounds are illustrated in Fig. 3.3. Furthermore, we provide the justification of treating the bandwidth  $\omega$  of  $C(s)$  and the adaptation rate  $\Gamma$  as tuning-knobs. Indeed, the upcoming analysis will show that we can ensure that  $x(t) \in \Omega(\rho, x^*(t))$  (see (3.4)) for all  $t \geq 0$ .

We begin with the bound between the reference system state and desired state trajectory. This corresponds to the green tube in Fig. 3.3. The proofs for all the claims in this section are provided in Appendix B.2.

**Lemma 3.1.** *Let all the assumptions hold and let  $\rho_r$  be given in (3.30). If the conditions in (3.32a)-(3.32b) hold, then for any desired state trajectory  $x^*(t)$ , the state  $x_r(t)$  of the reference system in (3.33) satisfies*

$$x_r(t) \in \Omega(\rho_r, x^*(t)), \quad \forall t \geq 0, \quad (3.34)$$

*and is uniformly ultimately bounded as*

$$x_r(t) \in \Omega(\mu(\omega, T), x^*(t)) \subset \Omega(\rho_r, x^*(t)), \quad \forall t \geq T > 0, \quad (3.35)$$

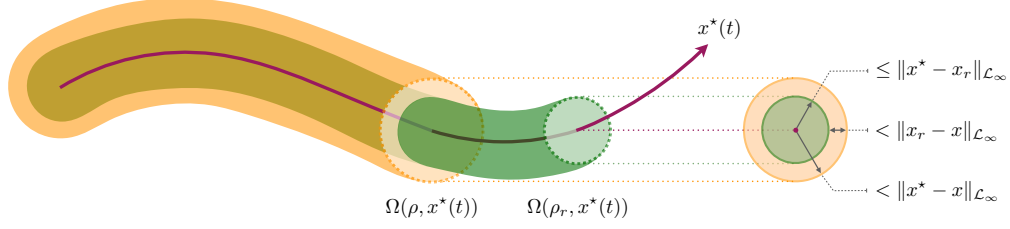


Figure 3.3: The bounds/tubes for the analysis of the CCM based  $\mathcal{L}_1$ -adaptive controller.

where the ultimate bound is defined as

$$\mu(\omega, T) := \sqrt{\frac{e^{-2\lambda T} \mathcal{E}(x_0^*, x_0)}{\underline{\alpha}} + \zeta_1(\omega)}. \quad (3.36)$$

Next, we compute the bounds between the reference system in (3.33) and the  $\mathcal{L}_1$  closed-loop system ((3.1) with (3.13)).

**Lemma 3.2.** *Suppose that the stated assumptions and the conditions in (3.32) hold. Additionally, assume that the trajectory of the  $\mathcal{L}_1$  closed-loop system satisfies  $x(t) \in \Omega(\rho, x^*(t))$ , for all  $t \in [0, \tau]$ , for some  $\tau > 0$ , with  $\Omega(\rho, x^*(t))$  and  $\rho$  defined in (3.4) and (3.30), respectively. Then,*

$$\|x_r - x\|_{\mathcal{L}_\infty}^{[0, \tau]} < \rho_a,$$

where  $\rho_a$  is given in (3.30).

We now use Lemmas 3.1 and 3.2 to state the main result of the chapter.

**Theorem 3.2.** *Suppose that the stated assumptions and conditions in (3.32) hold. Consider a desired state trajectory  $x^*(t)$  as in (3.3) and the state of the  $\mathcal{L}_1$  closed-loop system defined via (3.1) and (3.13). Then, the system state satisfies*

$$x(t) \in \Omega(\rho, x^*(t)), \quad \forall t \geq 0, \quad (3.37)$$

and is uniformly ultimately bounded as

$$x(t) \in \Omega(\delta(\omega, T), x^*(t)) \subset \Omega(\rho, x^*(t)), \quad \forall t \geq T > 0. \quad (3.38)$$

Here, the ultimate bound is defined as

$$\delta(\omega, T) := \mu(\omega, T) + \rho_a, \quad (3.39)$$

where the positive scalars  $\rho$  and  $\rho_a$  are defined in (3.30), and  $\mu(\omega, T)$  is defined in Lemma 3.1.

## Discussion

A few critical comments are in order for the performance analysis. The main result in Theorem 3.2 provides uniform ultimate bounds. Let us first discuss the implication of the uniform bound  $\rho$  in (3.37). As per the definition in (3.30),  $\rho = \rho_r + \rho_a$ . It is evident from the definition that  $\rho$  is lower bounded by the initial condition difference  $\|x_0^* - x_0\|$  and the positive scalars  $\underline{\alpha}$  and  $\bar{\alpha}$  which are associated with the CCM  $M(x)$  of the nominal dynamics. Furthermore, as per the proof of Lemma 3.2, since  $\rho_a \propto 1/\sqrt{\Gamma}$ , the adaptation rate  $\Gamma$  can be increased to the maximum value allowable by the computation hardware to guarantee the smallest  $\rho_a$ , and thus, the smallest uniform bound  $\rho$ . However, the fact remains that the uniform bound  $\rho$  guaranteed by the  $\mathcal{L}_1$ -controller for the tracking remains lower bounded by  $\|x_0^* - x_0\|$ . The only way this bound can be further reduced is if the underlying planner which provides the desired state-input pair  $(x^*(t), u^*(t))$  can minimize  $\|x_0^* - x_0\|$ .

Theorem 3.2 also provides the (uniform) ultimate bound via  $\delta(\omega, T)$  defined in (3.39). As already mentioned,  $\rho_a \propto 1/\sqrt{\Gamma}$ . Furthermore, from the definition of  $\zeta_1(\omega)$  in (3.31a), it is evident that by choosing a large enough  $\omega$ , there will always exist a known  $0 < T < \infty$  such that  $\delta(\omega, t) \leq \bar{\rho}$ , for all  $t \geq T$ , for any chosen  $\bar{\delta} > 0$ . Therefore, we can always arbitrarily shrink the tube  $\mathcal{O}(\bar{\delta})$  by choosing appropriate bandwidth  $\omega$  and rate of adaptation  $\Gamma$ . This feature of the  $\mathcal{CL}_1$ -controller is very advantageous, since, for example, this capability will allow the safe navigation of a robot through tight and cluttered environments. This improved performance, however, comes at the cost of reduced robustness. There exists a trade-off between performance and robustness that should be taken into consideration. As aforementioned, performance (radius of tubes around  $x^*(t)$ ) depends on  $\Gamma$  and  $\omega$ . The rate of adaptation  $\Gamma$  is obviously limited by the available computational hardware. More importantly, the role of the low-pass filter  $C(s)$  in the  $\mathcal{L}_1$ -control architecture (Fig. 3.2) is to decouple the control loop from the estimation loop [15]. Thus, increasing the bandwidth  $\omega$  of  $C(s)$  in order to get a tighter tube will lead to the  $u_a(t)$  component of the  $\mathcal{L}_1$ -input to behave as a high-gain signal, thus possibly sacrificing desired robustness levels [110]. Therefore, this trade-off must always be taken into account during the planning phase.

## 3.6 Numerical Results

We provide two illustrative examples. In the first example, we consider the non-feedback linearizable system from [92] and synthesize the controller to ensure safe regulation around the equilibrium point. We also show the effect of uniform ultimate bounds discussed in the previous section, if the system were to start far away from the equilibrium. In the second example, we consider the system from [111] and ensure safety in a motion planning context during trajectory tracking. In particular

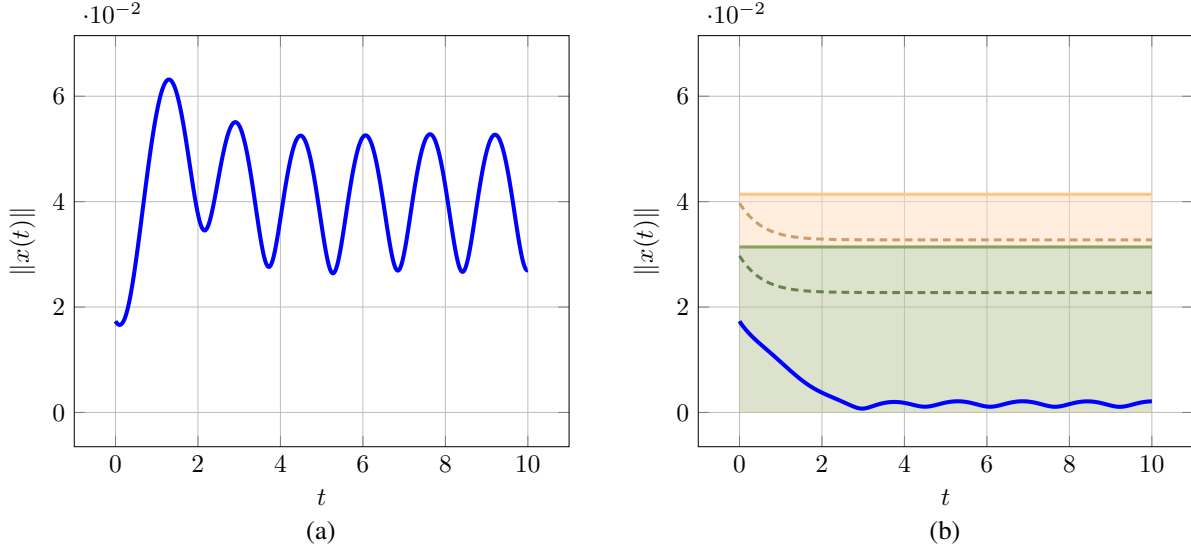


Figure 3.4: Comparison of controller performance between (a) pure CCM-based feedback, and a (b)  $\mathcal{CL}_1$  architecture. The green and orange shaded regions signify the induced  $\Omega(\rho_r, x^*)$  and  $\Omega(\rho, x^*)$  tubes respectively. The dashed green and orange lines signify the uniform ultimate bounds  $\mu(\omega, T)$  and  $\delta(\omega, T)$  evaluated at every timestep.

we show how altering the tube parameters affects the choice in the filter bandwidth and adaptation rate.

### 3.6.1 Non-feedback Linearizable Systems

Consider the system with the structure defined in (3.1) and the system functions given by

$$f(x) = \begin{bmatrix} -x_1(t) + x_3(t) \\ x_1^2(t) - 2x_1(t)x_3(t) - x_2(t) + x_3(t) \\ -x_2(t) \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

where the state  $x(t) = [x_1(t) \ x_2(t) \ x_3(t)]^\top$ . The dual metric  $W(x) = M(x)^{-1}$  satisfying the conditions in (3.8a) to (3.8c) was found using the sum-of-squares programming toolbox SumOf-Squares.jl [112], optimization software JuMP [113], and the optimization solver [114], as

$$W(x) = \begin{bmatrix} 0.2 & -0.41x_1(t) & -0.01 \\ -0.41x_1(t) & 0.81x_1(t)^2 + 0.22 & 0.01x_1(t) - 0.01 \\ -0.01 & 0.01x_1(t) - 0.01 & 0.07x_1(t) + 0.22 \end{bmatrix}.$$

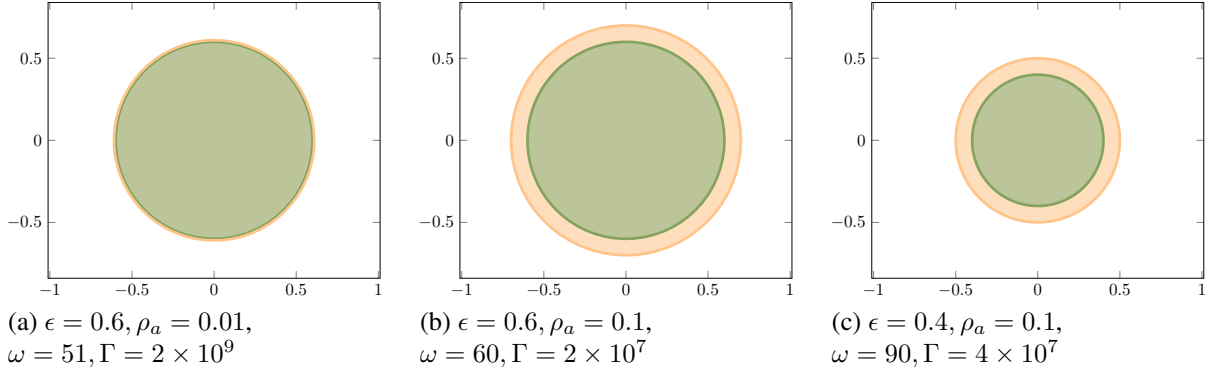


Figure 3.5: Relationship between the choice of tube parameters  $\epsilon$  and  $\rho_a$  and the controller parameters  $\omega$  and  $\Gamma$  through the conditions defined in (3.32). For clarity the initial conditions for the desired trajectory and the actual system in this illustration are assumed to be the same.

The metric satisfies a convergence rate  $\lambda = 1.0$  and is uniformly bounded in the set  $\mathcal{X} = \{y \in \mathbb{R}^3 \mid \|y\|_\infty \leq 0.1\}$  with  $\bar{\alpha} = 5.88$  and  $\underline{\alpha} = 3.85$ . Now, suppose that the system is experiencing sinusoidal disturbances of the form:  $h(t) = 0.1 \sin(2t)$ . We chose the initial condition of the system as  $x_0 = [1 \ -1 \ 1]^\top \times 10^{-2}$  and the desired state as  $x^* = [0 \ 0 \ 0]^\top$ . Incidentally, the desired state is also the equilibrium point of the system which means that the desired control is  $u^*(t) \equiv 0$ .

A pure CCM-based feedback strategy produces the oscillatory behavior, seen in Fig. 3.4a. A  $\mathcal{CL}_1$ -adaptive controller is designed in Fig. 3.4b for tube widths  $\epsilon = 0.01$  and  $\rho_a = 0.01$ . The filter bandwidth and adaptation rate required to achieve this level of performance were chosen as  $\omega = 50$  and  $\Gamma = 5 \times 10^6$  respectively by satisfying the conditions in (3.32). Notice that the bounds are far more conservative than the actual behavior of the system. In fact, the error in tracking is uniformly bounded as  $\|x\|_{\mathcal{L}_\infty} < 0.02$ . Additionally, notice that the uniform ultimate bounds of the reference system tube from (3.36) and the actual system tube from (3.39) shrink with time and are essentially ‘forgetting’ the initial conditions of the system.

### 3.6.2 Safe Tubes for Motion Planning

Consider the system with the structure defined in (3.1) and the system functions given by

$$f(x) = \begin{bmatrix} -x_1(t) + 2x_2(t) \\ -0.25x_2^3(t) - 3x_1(t) + 4x_2(t) \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 \\ -2 \end{bmatrix},$$

where the state  $x(t) = [x_1(t) \ x_2(t)]^\top$ . Since this particular system is feedback linearizable, it admits a constant (or flat) dual metric for all  $x \in \mathbb{R}^2$  [93]. The value of the dual metric and the

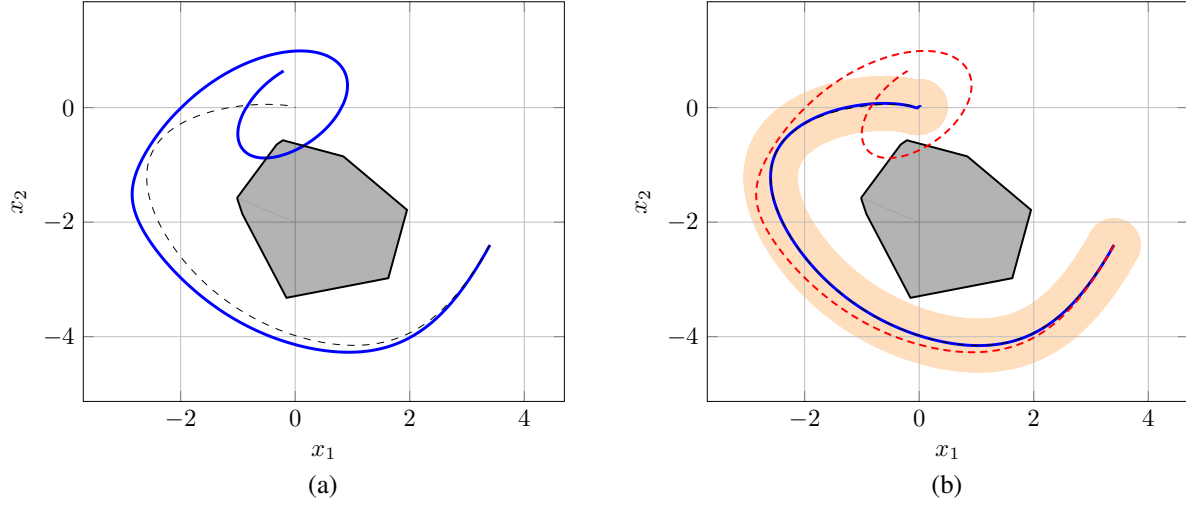


Figure 3.6: Comparison of performance and robustness between (a) pure CCM-based feedback, and (a)  $\mathcal{CL}_1$  architecture with tube parameters  $\epsilon = 0.4$  and  $\rho_a = 0.1$ . The dashed black line shows the desired trajectory designed by a planner; the gray polygon is an obstacle, and the orange shaded region is the safe tube given by  $\Omega(\rho, x^*(t))$ . The behavior of the system under pure CCM-based feedback has been overlaid as a dashed red line in (b) for clarity.

associated convergence parameter is computed in [111] and provided here for completeness:

$$W = \begin{bmatrix} 4.26 & -0.93 \\ -0.93 & 3.77 \end{bmatrix}, \quad \lambda = 1.74.$$

Similar to [111], we chose the initial condition of the system as  $x_0 = [3.4 \ -2.4]^\top$  and the target state as  $x^* = [0 \ 0]^\top$ . The desired state and control trajectory pair was computed using the iterative LQR solver provided by [115] with the parameters  $Q = 0.5\mathbb{I}_2$  and  $R = 1.0$ . Suppose the system is affected by uncertainties of the form:  $h(t, x) = -2\sin(2t) - 0.1\|x(t)\|$ , consisting of both time and state dependent terms. Depending on the desired level of tracking performance or closeness to obstacles in the environment, the user will pick the tube parameters  $\epsilon$  and  $\rho_a$  as defined in (3.30). In Fig. 3.5, we illustrate the trade-offs between choosing a tighter  $\rho_a$  (Fig. 3.5a) versus a tighter  $\epsilon$  (Fig. 3.5c) for this system.

In Fig. 3.6b, we observe the performance and robustness benefits of using  $\mathcal{CL}_1$ -adaptive control. Not only does the system track the desired trajectory closely, but also avoids colliding with obstacles (unlike in Fig. 3.6a) through an appropriate choice of tube parameters.



# Chapter 4

## Safe Simultaneous Learning and Control

### 4.1 Introduction

Machine learning (ML) algorithms are potent tools for producing complex and accurate models of uncertain systems. The accurate representations help model-based reinforcement learning (MBRL) algorithms achieve performance and optimality [116]. However, model uncertainties can make the system unstable during learning transients, which can have serious consequences, especially for safety-critical systems [117]. Control-theoretic approaches based on Lyapunov functions (LFs) and control invariant sets can offer safety certificates [20, 118, 119]. For instance, control-theoretic notions like asymptotic stability [120, Chapter 3] are useful to guarantee system behavior in the limit. However, it is equally important to quantify the system’s behavior during the complete operation and not just in the limit. More importantly, for learning-based control, the question of how to quantify and guarantee the system’s safety during learning-transients is crucial.

Learning-based tracking control [121–124] reduces conservatism by using measured data to improve models. However, several frameworks [125–129] in this domain also require restrictions on the structure of the dynamics (e.g. strict feedback form, differential flatness, etc.) to ensure safety in the presence of uncertainties. The authors in [20] use the regularity of the uncertainty and the sufficient statistics of the learned Gaussian process (GP) models to safely expand the region of attraction, and improve control performance using LFs. In contrast, our control architecture actively compensates for the model uncertainties allowing the system states to reach any part of the operating region safely even when the quality of the learned model is poor. Probabilistic chance constraint methods, which use uncertainty propagation, have been shown to provide both asymptotic and transient bounds on the tracking performance [130]. The implementations that rely on approximate uncertainty propagation offer excellent empirical performance without theoretical guarantees, shown in [21, 131]. However, uncertainty propagation methods sacrifice long-term accuracy for computational efficiency, for example by linearization, in order to be more tractable for real-time applications. Our proposed method avoids uncertainty propagation completely when considering nonlinear dynamics. Instead, we rely on uniform error bounds for GP predictions to

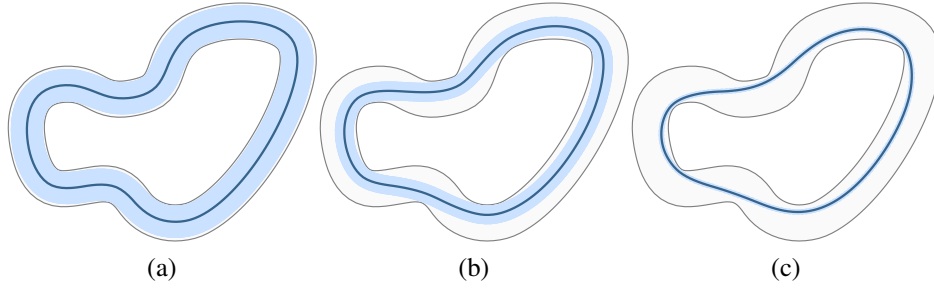


Figure 4.1: Consider a vehicle traversing a race track with some nominal model knowledge. Depending on the uncertainty and robustness requirements, the control framework guarantees the safety bounds (blue tubes). As the learning improves, so do the performance and optimality.

apriori guarantee tracking performance with respect to the desired trajectory. This controller is capable of incorporating the learned dynamics while ensuring safety. This incorporation is based on both contraction theory [92, 111, 132] and the  $\mathcal{L}_1$ -adaptive control theory [15]. Safe planning and control using  $\mathcal{L}_1$ -adaptive control theory can be found in [41, 103, 133].

We provide a systematic design of the feedback law  $u = \pi(x, x_d, u_d)$ , such that we can apriori compute a positive scalar  $\rho$  to ensure  $\|x_d(t) - x(t)\| \leq \rho$ , for all  $t \in [0, T_f]$ , with high-probability, where  $T_f \leq \infty$  is the planning horizon. This guarantee implies the existence of a tube  $\mathcal{O}_{x_d}(\rho)$  of radius  $\rho$  centered around the trajectory  $x_d$ , in which the actual state of the system  $x$  is guaranteed to lie in. The control design is based on our recent work in [41]. We then define the notion of safety as the existence of the apriori quantifiable tubes  $\mathcal{O}_{x_d}(\rho)$ . Any planning algorithm which produces  $(x_d, u_d)$  w.r.t.  $\bar{F}$  by incorporating the additional constraint  $\mathcal{O}_{x_d}(\rho) \notin \mathcal{X}_{\text{obs}}$  will thus ensure that the actual state  $x \notin \mathcal{X}_{\text{obs}}$  (obstacle set). This minimal requirement enables the framework to be used in conjunction with many popular planning algorithms like [31, 37–40, 134].

We define performance as the radius  $\rho$  of the tube since a smaller radius  $\rho$  implies better tracking performance and vice-versa. To improve performance, we rely on Bayesian learning (GP regression) to learn the model uncertainties. We use GP learning’s predictive distribution to compute high-probability error bounds for the estimated uncertainties. These estimates are then incorporated within the feedback law  $u = \pi(x, x_d, u_d)$  to handle the uncertainties as represented by the variance. The planner can thus operate by only incorporating the mean dynamics. The notion of performance also includes the desired robustness margins of the closed-loop system. Our recently developed control methodology [41], introduced in the previous section, provides a sensible approach to balance the trade-off between performance and robustness requirements for safe navigation. However, this trade-off implies a limit on how tight the performance bounds can get for a specification of robustness margins. This framework addresses this problem by using model learning to reduce the uncertainty resulting in tighter performance bounds than would be possible with [41] alone.

The improved performance, as defined above, implies an improvement in optimality. Using improved performance given guaranteed safety (in the form of reduced radii tubes), a planner can produce the desired trajectory  $x_d$  that is optimal in the sense of the total path's length and the time taken to traverse the path. We refer to this as performance-dependent optimality. However, there is an additional notion of optimality w.r.t. to the learned models. MBRL algorithms rely on the known/learned model  $\bar{F}$  to generate pairs  $(x_d, u_d)$  optimal for  $\bar{F}$ . Therefore, as learning improves, and thus  $\bar{F} \rightarrow F$ , the underlying MBRL should produce desired trajectories approaching optimality w.r.t. the actual dynamics. We refer to this as model-based optimality. Both performance-based and model-based optimality constitute the overall optimality. It is important to highlight that, as aforementioned, our control framework is planner agnostic: it enables the improvement of optimality via any planner capable of doing so, rather than guaranteeing it. The proposed framework provides the planner with improved performance guarantees and learned models; it is up to the planner to use these to improve optimality. The proposed framework enables the behavior in Fig. 4.1. Note that this figure does not show the distinction of the aforementioned optimality types but rather an improvement in the overall cost of the planned trajectory.

## 4.2 Problem Formulation

We consider the following uncertain control-affine nonlinear dynamics of the form

$$\dot{x}(t) = f(x(t)) + B(x(t))(u(t) + h(x(t))), \quad x(0) = x_0, \quad (4.1)$$

where  $x(t) \in \mathbb{R}^n$ ,  $u(t) \in \mathbb{R}^m$ , for  $t \in \mathbb{R}_{\geq 0}$ , represent the system state and control input, respectively. The functions  $f(x) \in \mathbb{R}^n$  and  $B(x) \in \mathbb{R}^{n \times m}$  are the known components of the dynamics, whereas,  $h(x) \in \mathbb{R}^m$  denotes the model uncertainties. The control-affine systems presented in (4.1) cover a wide range of physical control systems including, for e.g., nonlinear aircraft models [135], and quadrotor models [95, 136]. Note that in (4.1), for the clarity of exposition, we place the assumption that the uncertainties are matched, i.e.,  $g(x) = B(x)h(x) \in \text{span}\{B\}$ . The proposed method can be extended for uncertainties  $\notin \text{span}\{B\}$  following the work laid out in [95] and [137], with the expectation that the unmatched uncertainties cannot be fully compensated but instead only attenuated.

**Definition 4.1.** *We define the known and uncertain model parameter sets as  $\bar{\mathcal{M}} = \{f, B\}$  and  $\hat{\mathcal{M}} = \{h\}$ , respectively. These sets induce the vector fields  $F(\bar{\mathcal{M}}; x, u) = f(x) + B(x)u$  and  $F(\mathcal{M}; x, u) = f(x) + B(x)(u + h)$ , where  $\mathcal{M} = \bar{\mathcal{M}} \cup \hat{\mathcal{M}}$ , which define the known and actual*

(uncertain) dynamics respectively as<sup>1</sup>

$$\dot{x} = F(\bar{\mathcal{M}}; x, u) = f(x) + B(x)u, \quad \dot{x} = F(\mathcal{M}; x, u) = f(x) + B(x)(u + h(x)). \quad (4.2)$$

The following definition is placed for planning.

**Definition 4.2.** Over a planning horizon  $[0, T_f]$ ,  $0 < T_f \leq \infty$ ,  $(x^*(t), u^*(t))$  is a desired state-input pair if  $\dot{x}_d(t) = F(\bar{\mathcal{M}}; x^*(t), u^*(t))$  and  $x^*(t) \in \mathcal{X}$ , for all  $t \in [0, T_f]$ , where  $\mathcal{X} \subset \mathbb{R}^n$  is any compact convex set. Given any  $\rho > 0$ , we define

$$\Omega(\rho, x^*(t)) := \{y \in \mathbb{R}^n \mid \|y - x^*(t)\| \leq \rho\}, \quad \mathcal{O}_{x^*}(\rho) = \cup_{t \in [0, T_f]} \Omega(\rho, x^*(t)). \quad (4.3)$$

We refer to  $\mathcal{O}_{x^*}(\rho)$  as the tube. Here  $\|\cdot\|$  denotes the Euclidean norm.

Since  $F(\bar{\mathcal{M}}; x, u)$  is known, any model-based planner can generate the desired pair  $(x^*(t), u^*(t))$  satisfying the state-constraints. The following ensures the generation of safe desired trajectories.

**Assumption 4.1.** Given any tube width  $\rho > 0$  and planning horizon  $[0, T_f]$ ,  $0 < T_f \leq \infty$ , the planner produces a state-input pair  $(x^*(t), u^*(t))$  (as in Definition 4.2) such that the induced tube  $\mathcal{O}_{x^*}(\rho)$  satisfies  $\mathcal{O}_{x^*}(\rho) \in \mathcal{X} / \mathcal{X}_{obs}$ , for all  $t \in [0, T_f]$ , where  $\mathcal{X}_{obs}$  represents the obstacles. The desired control input  $u^*(t)$  satisfies  $\|u^*(t)\| \leq \Delta_{u^*}$ , for all  $t \in [0, T_f]$ , with the upper bound known.

**Problem Statement:** Given the learned probabilistic estimates of the uncertainty  $h(x)$ , any desired state-input pair  $(x^*(t), u^*(t))$ ,  $t \in [0, T_f]$ , designed by a planner using the known dynamics  $F(\bar{\mathcal{M}}; x, u)$  (Definitions 4.1 and 4.2), and the desired robustness margins, the goal is to design the control input  $u(t)$  that guarantees the existence of an apriori computable tube-width  $\rho$  so that the state of the uncertain dynamics in (4.1) ( $F(\mathcal{M}; x, u)$  in Definition 4.1) satisfies  $x(t) \in \Omega(\rho, x^*(t)) \subset \mathcal{O}_{x^*}(\rho)$  with high probability, for all  $t \geq 0$ , from all initial conditions  $x_0 \in \mathcal{X}$ , while satisfying the robustness requirements. Importantly, the existence of the pre-computable tubes should not depend on the quality of the learned estimates, thus ensuring that safety remains decoupled from learning. The learning should only affect the performance bounds and the optimality of the planned trajectory.

We now place the following assumptions on the known and uncertain model parameters.

**Assumption 4.2.** The functions  $f(x)$ ,  $B(x)$ ,  $h(x)$  are continuous, bounded, and Lipschitz, for all  $x \in D \subset \mathbb{R}^n$ , where  $D$  is any compact set which can be arbitrarily large. Moreover, the matrix  $B(x)$  has full column rank for all  $x \in D$ , thus guaranteeing the existence of the Moore-Penrose inverse  $B^\dagger(x) = (B^\top(x)B(x))^{-1} B^\top(x)$ .

---

<sup>1</sup>We suppress the temporal dependencies for brevity.

The probabilistic estimates of the uncertainty  $h(x)$  in (4.1) are learned using GP regression. For completeness, we now provide a brief preliminary on GP regression.

### 4.3 Preliminaries on Gaussian Process Regression

**Assumption 4.3.** *We assume that each of the elements  $[h]_i(x)$ ,  $i \in \{1, \dots, m\}$ , are independent. Moreover, we assume that each element is a sample from a GP  $[h]_i(x) \sim \mathcal{GP}(0, K_i(x, x'))$ , where the kernel functions  $K_i : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$  are known. Moreover, the kernels are twice-continuously differentiable with known constants  $L_{K_i}$ ,  $\nabla_x L_{K_i}$ , such that  $L_{K_i} = \max_{x, x' \in \mathcal{X}} \|\nabla_x K_i(x, x')\|$ , and  $\nabla_x L_{K_i} = \max_{x, x' \in \mathcal{X}} \|\nabla_x^2 K_i(x, x')\|$ , for  $i \in \{1, \dots, m\}$ .*

The assumption is less conservative than requiring the uncertainty to be a member of the reproducing kernel Hilbert space (RKHS) associated with the kernel. For example, sample functions of GPs with squared-exponential kernels correspond to continuous functions, whereas the associated RKHS space contains only analytic functions [138]. Moreover, the constants assumed to exist in Assumption 4.3 are easily computable, for example, for the squared-exponential kernel. However, it is important to note that the element-wise independence assumption on the uncertainty might be restrictive in certain scenarios, and we are investigating relaxing this condition in future work.

Assume that we have  $N$  measurements of the form  $y_k = h(x_k) + \kappa = B^\dagger(x_k)(\dot{x}_k - f(x_k)) - u_k + \kappa \in \mathbb{R}^m$ ,  $k \in \{1, \dots, N\}$ , where  $\kappa \sim \mathcal{N}(0_m, \sigma^2 \mathbb{I}_m)$  is the measurement noise and  $0_m \in \mathbb{R}^m$  is a vector of zeros. We set up the data as  $\mathbf{D} = \{\mathbf{Y}, \mathbf{X}\}$ ,  $\mathbf{Y} = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix} \in \mathbb{R}^{m \times N}$ ,  $\mathbf{X} = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix} \in \mathbb{R}^{n \times N}$ . Thus, for each of the constituent functions  $[h]_i$ ,  $i \in \{1, \dots, m\}$ , we have the data as  $\mathbf{D}_i = \{[\mathbf{Y}]_{i,\cdot}, \mathbf{X}\}$ , where  $[\mathbf{Y}]_{i,\cdot}$  denotes the  $i^{\text{th}}$  row of  $\mathbf{Y}$ . Conditioning the prior in Assumption 4.3 on the measured data  $\mathbf{D}$ , we obtain the posterior distribution at any test point  $x_t \in \mathcal{X}$  as [139]

$$\mathbb{R} \ni [h]_i(x_t) \sim \mathcal{N}(\nu_{i,N}(x_t), \sigma_{i,N}^2(x_t)), \quad i \in \{1, \dots, m\}, \quad (4.4)$$

with mean  $\nu_{i,N}(x_t) = K_i(x_t, \mathbf{X})^\top [K_i(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbb{I}_N]^{-1} ([\mathbf{Y}]_{i,\cdot})^\top$ , and variance  $\sigma_{i,N}^2(x_t) = K_i(x_t, x_t) - K_i(x_t, \mathbf{X})^\top [K_i(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbb{I}_N]^{-1} K_i(x_t, \mathbf{X})$ . Using the linearity of the differential operator, we also compute the posterior distributions of the partial derivatives of  $h(x)$  as

$$(\nabla_x [h]_i(x_t))^\top \sim \mathcal{N}(\nabla_x \nu_{i,N}(x_t)^\top, \nabla_x \sigma_{i,N}^2(x_t)), \quad (4.5)$$

with mean  $\nabla_x \nu_{i,N}(x_t)^\top = (\nabla_x K_i(x_t, \mathbf{X}))^\top [K_i(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbb{I}_N]^{-1} ([\mathbf{Y}]_{i,\cdot})^\top \in \mathbb{R}^n$  and variance  $\nabla_x \sigma_{i,N}^2(x_t) = \nabla_{x,x'}^2 K_i(x_t, x_t) - (\nabla_x K_i(x_t, \mathbf{X}))^\top [K_i(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbb{I}_N]^{-1} \nabla_x K_i(x_t, \mathbf{X}) \in \mathbb{S}^n$ .

## 4.4 $\mathcal{CL}_1$ Control with Gaussian Process Learning

We now present our control framework which brings together contraction theory-based  $\mathcal{L}_1$ -adaptive ( $\mathcal{CL}_1$ ) control presented in Chapter 3 with GP regression. The detailed description of the framework is illustrated in Fig. 4.2. Given the posterior distribution of the uncertainty in (4.4), we may update the known and uncertain model parameter sets in (4.2), as  $\bar{\mathcal{M}} = \{f + B\nu_N, B\}$ ,  $\hat{\mathcal{M}} = \{h - \nu_N\}$ ,  $\mathcal{M} = \bar{\mathcal{M}} \cup \hat{\mathcal{M}}$ , which induce the learned representations of the known and actual dynamics as

$$\dot{x} = F(\bar{\mathcal{M}}; x, u) = f(x) + B(x)\nu_N(x) + B(x)u, \quad (4.6a)$$

$$\dot{x} = F(\mathcal{M}; x, u) = f(x) + B(x)\nu_N(x) + B(x)(u + h(x) - \nu_N(x)). \quad (4.6b)$$

Note that this step simply entails adding and subtracting the mean  $\nu_N$  in the control channel. The control design is driven by the philosophy that the input  $u$  compensates for the uncertainty  $h(x) - \nu_N(x)$  as quantified by the variance of the posterior distribution in (4.4). The uncertainty is compensated so that the actual system  $\dot{x} = F(\mathcal{M}; x, u)$  behaves like the known/learned  $\dot{x} = F(\bar{\mathcal{M}}; x, u)$  within guaranteed tube bounds presented in Definition 4.2. Then, any underlying planner can generate the desired pair  $(x^*, u^*)$  satisfying the deterministic and uncertainty-free dynamics  $\dot{x}^* = F(\bar{\mathcal{M}}; x^*, u^*)$  safe in the knowledge that the state  $x$  of the actual uncertain system  $\dot{x} = F(\mathcal{M}; x, u)$  will remain in the tube  $\mathcal{O}_{x^*}(\rho)$  centered on  $x^*$ . Therefore the following sequential tasks need to be performed: i) quantification of the uncertainty  $h - \nu_N$ , and ii) design of the input  $u$  to compensate for the quantified uncertainty.

To quantify the uncertainty, we use the posterior distributions in (4.4) and (4.5) to show that there exist bounds  $\Delta_{\Xi_u}$ , such that for all  $x \in \mathcal{X}$ , with high probability,

$$\|\Xi_u(x)\| \leq \Delta_{\Xi_u}, \quad (4.7)$$

where  $\Xi_u = \{h - \nu_N, \frac{\partial h - \nu_N}{\partial x}\}$ . The bounds are presented in the following theorem, which is presented in a highly condensed form due to space considerations. The expanded version of the theorem and its proof can be found in [140, Theorem. 3.1].

**Theorem 4.1.** *Let Assumptions 4.2 and 4.3 hold. Consider the posterior distributions in (4.4) and (4.5) and any scalars  $\delta \in (0, 1)$ ,  $\tau > 0$ . Then there exist computable functions  $\Delta h(x, \tau)$  and  $\nabla_x \Delta h(x, \tau)$ , so that with  $\Delta_{h-\nu_N} = \sup_{x \in \mathcal{X}} \Delta h(x, \tau)$  and  $\Delta_{\frac{\partial h - \nu_N}{\partial x}} = \sup_{x \in \mathcal{X}} \nabla_x \Delta h(x, \tau)$ , and the bounds in (4.7) hold with probability at least  $1 - \delta$ .*

With the bounds on the learned representation of the uncertainty  $h(x) - \nu_N(x)$  established in Theorem 4.1, we proceed with the control design. To reiterate, for any  $(x^*, u^*)$  satisfying the

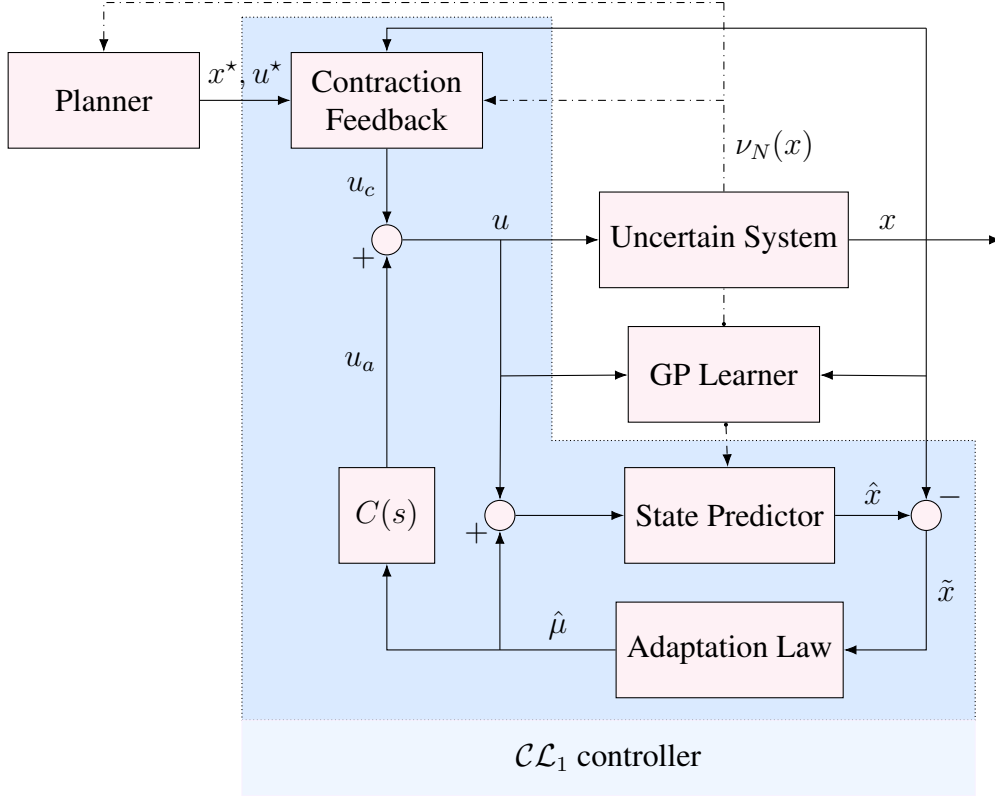


Figure 4.2: The proposed safe learning and control architecture.

deterministic learned dynamics  $\dot{x}_d = F(\bar{\mathcal{M}}; x^*, u^*)$  in (4.6a), we design the input  $u$  so that the state of the actual dynamics  $\dot{x} = F(\mathcal{M}; x, u)$  in (4.6b) satisfies  $\|x - x^*\| \leq \rho \Rightarrow x \in \mathcal{O}_{x^*}(\rho)$  uniformly in time, for some tube-width  $\rho > 0$ . The control input, as in Fig. 4.2, is computed as

$$u(t) = u_c(\bar{\mathcal{M}}; t) + u_a(\bar{\mathcal{M}}; t), \quad (4.8)$$

where  $u_c(\bar{\mathcal{M}}; t)$  is the contraction-theory based input and  $u_a(\bar{\mathcal{M}}; t)$  is the  $\mathcal{L}_1$ -adaptive input. By Assumption 3.6 in Section 3.3, there exists a control contraction metric (CCM)  $M(x)$  for the original known dynamics  $\dot{x} = F(\bar{\mathcal{M}}; x, u)$  ((4.2)). More importantly, by [132, Lemma 1],  $M(x)$  is a CCM for both the learned representations of the known and actual (uncertain) dynamics in (4.6). This allows us to seamlessly incorporate the learned mean  $\nu_N$  and straightforwardly design the input  $u_c(\bar{\mathcal{M}}, t)$  for the learned representation in (4.6a) using the CCM  $M(x)$  as in [95, Sec. 5.1]. The design of the  $\mathcal{L}_1$ -adaptive controller consists of a state-predictor, adaptation-law, and a low-pass filter  $C(s)$  as illustrated in Fig. 4.2. Jointly, the  $\mathcal{L}_1$  input  $u_a(\bar{\mathcal{M}}; t)$  can be represented as

$$\dot{\hat{x}} = F(\bar{\mathcal{M}}; x, u + \hat{\mu}) + A_m \tilde{x}, \quad \dot{\hat{\mu}} = \Gamma \text{Proj}_{\hat{\mathcal{H}}}(\hat{\mu}, -B^\top(x)P\tilde{x}), \quad u_a(\bar{\mathcal{M}}; s) = -C(s)\hat{\mu}(s), \quad (4.9)$$

with  $\hat{x}(0) = x_0$ ,  $\hat{\mu}(0) \in \hat{\mathcal{H}}$ , and  $s$  represents the Laplace variable. Here,  $\hat{x}$  is the state of the

predictor,  $\tilde{x} = \hat{x} - x$ , and  $A_m \in \mathbb{R}^{n \times n}$  is an arbitrary Hurwitz matrix. The uncertainty estimate  $\hat{\mu}$  is generated by the adaptation law for adaptation rate  $\Gamma > 0$ , with the projection operator  $\text{Proj}_{\hat{\mathcal{H}}}(\cdot, \cdot)$ , which ensures that  $\hat{\mu} \in \hat{\mathcal{H}}$ ,  $\hat{\mathcal{H}} = \{y \in \mathbb{R}^m \mid \|y\| \leq \Delta_{h-\nu_N}\}$ , and  $\mathbb{S}^n \ni P \succ 0$ , which is the solution to the Lyapunov equation  $A_m^\top P + P A_m = -Q$ , for any  $\mathbb{S}^n \ni Q \succ 0$ . Finally, the low-pass filter  $C(s)$  has a bandwidth  $\omega$  and satisfies  $C(0) = \mathbb{I}_m$ . The high-level design idea is that the input  $u_a$  compensates for the uncertainty  $h - \nu_N$  via the estimate  $\hat{\mu}$  and within the bandwidth of  $C(s)$ .

Next, we analyze the uncertain system  $\dot{x} = F(\mathcal{M}; x, u)$  in (4.6b) driven by the input (4.8). The complete details of the analysis can be found in [140]. Given a desired trajectory  $x^*$  and arbitrarily chosen positive scalars  $\rho_a$  and  $\epsilon$ , define

$$\rho_r = \sqrt{\frac{\bar{\alpha}}{\underline{\alpha}}} \|x^*(0) - x_0\| + \epsilon, \quad \rho = \rho_r + \rho_a, \quad (4.10)$$

where  $\bar{\alpha}, \underline{\alpha}$  are defined in Assumption 3.6. Under Assumptions 3.6 and 4.1, we obtain conditions on the magnitude of the rate of adaptation  $\Gamma$  and the bandwidth  $\omega$  of the low-pass filter  $C(s)$  in (4.9) so that we are guaranteed stability and can quantify the performance. Once again, for clarity we choose not to present the complete definitions of the conditions and the reader is directed to [140] for details. It is important to note that there always exists an adaptation rate  $\Gamma$  and a bandwidth  $\omega$  that satisfy these conditions; see [41] for further discussions. The following theorem establishes the performance of the closed-loop system and its proof can be found in [140, Theorem. 3.2].

**Theorem 4.2.** *Let Assumptions 4.1 and 4.3 hold and let the bounds in Theorem 4.1 be computed for some  $\delta \in (0, 1)$  and  $\tau > 0$ . Suppose the control input in (4.8) is designed so that the conditions on the rate of adaptation and filter bandwidth as given in [140] are satisfied. Then, given any desired pair  $(x^*, u^*)$  satisfying the deterministic known dynamics  $\dot{x}_d = F(\bar{\mathcal{M}}; x^*, u^*)$  in (4.6a), the state  $x$  of the actual (uncertain) dynamics  $\dot{x} = F(\mathcal{M}; x, u)$  in (4.6b) driven by the input  $u$  from (4.8) satisfies with probability at least  $1 - \delta$*

$$x(t) \in \Omega(\rho, x^*(t)) \subset \mathcal{O}_{x^*}(\rho), \quad \forall t \geq 0, \quad (4.11)$$

where  $\rho$  is defined in (4.10). Furthermore, the actual state  $x$  is uniformly ultimately bounded, with probability at least  $1 - \delta$ , as

$$x(t) \subset \Omega(\hat{\delta}(\omega, T), x^*(t)) \subset \mathcal{O}_{x^*}(\rho), \quad \forall t \geq T > 0, \quad (4.12)$$

where the uniform ultimate bound (UUB) is defined as  $\hat{\delta}(\omega, T) = \mu(\omega, T) + \rho_a$  with  $\mu(\omega, T) = \sqrt{e^{-2\lambda T} \mathcal{E}(x_{d,0}, x_0) / \underline{\alpha}} + \zeta_1(\Xi_{\{u,k,c\}}, \omega)$ .

**Discussion:** As the learning improves, the variance of the predictive Gaussian distribution col-



lapses, and thus the constants in (4.7) decrease. Therefore, without changing the filter bandwidth  $\omega$  and adaptation rate  $\Gamma$ , the UUB in Theorem 4.2 decreases. The decrease in the UUB, and the lack of a requirement for the re-tuning of the control parameters, is due to the monotonic dependence of the constants  $\zeta_i$  on  $\Delta_{\Xi_u}$  in (4.7). Furthermore, as aforementioned, the CCM  $M(x)$  does not need to be re-synthesized as the model is updated using learning. Thus, without re-tuning the parameters of the control input ( $M$ ,  $\Gamma$ , and  $\omega$ ), with the control designed using only Assumption 4.2, the performance improves as a function of learning. Of course the learning is not guaranteed to improve always, in which case, it will be reflected in the bounds  $\Delta_{\Xi_u}$ . In this scenario, we are in no compulsion to incorporate the learned estimates, since the controller guarantees safety with the previously learned, or no, estimates. This is the exact reason that the proposed method does not require a high-rate, or any fixed rate, of model updates. Whenever it is provided with an improved model, it will be incorporated. The uniform bound in (4.11) is lower-bounded by the initialization error in (4.10). The size of the terms  $\epsilon$  and  $\rho_a$  depends on the value of the adaptation rate  $\Gamma$  and filter-bandwidth  $\omega$ . Thus, while in theory we can achieve the lowest-possible tube width, the size of  $\Gamma$  is limited by the available computation, and  $\omega$  is limited by the desired robustness margins. Alternatively, instead of the uniform tube, a planner can use the UUB in (4.12), which induces tubes that exponentially collapse to a fixed radius dependent on  $\zeta_1$ , a term that decreases as the learning improves. Compared to our initial work in [141], the presented work is much more applicable to real-world problems. In particular, in [141] we could only consider linear known systems and did not provide any theoretical guarantees. In the presented work, we are able to explicitly consider nonlinear systems because of bringing contraction theory within  $\mathcal{L}_1$  control with theoretical guarantees. This further enables the use of learning for performance and optimality improvement with persistent safety as presented. Also note that since the adaptive control directly compensates for the uncertainty as quantified by the variance of the posterior distribution, any underlying planner need only incorporate the deterministic mean and not perform any uncertainty propagation, which is both approximate and computationally expensive. Finally, note the semi-global nature of the  $\mathcal{L}_1$  augmentation. For a given CCM controller  $u_c$  that renders the known dynamics incrementally exponentially stable (IES), the  $\mathcal{L}_1$  augmentation can make any tube, no matter how large, forward invariant for the actual (uncertain) dynamics. The semi-global nature comes from the fact that, as is evident in this section, the control design explicitly depends on the size of the set/tube.

## 4.5 Numerical Results

We demonstrate our approach using two illustrative simulations. In the first example, we consider a modified Dubin's car system from [8] and show how our control framework is used to ensure safety

guarantees during the learning process, while outperforming a purely CCM-based approach. In the second example, we consider a planar quadrotor model from [95] and show the usefulness of our control framework in feedback motion planning applications. The CCMs were discovered using DNNs from [8] in the first example, and using the sum-of-square programming approach from [92] in the second example. In both scenarios, the dataset is generated by randomly sampling the state space, but one could also use more sophisticated exploration techniques to safely gather data based on our framework.

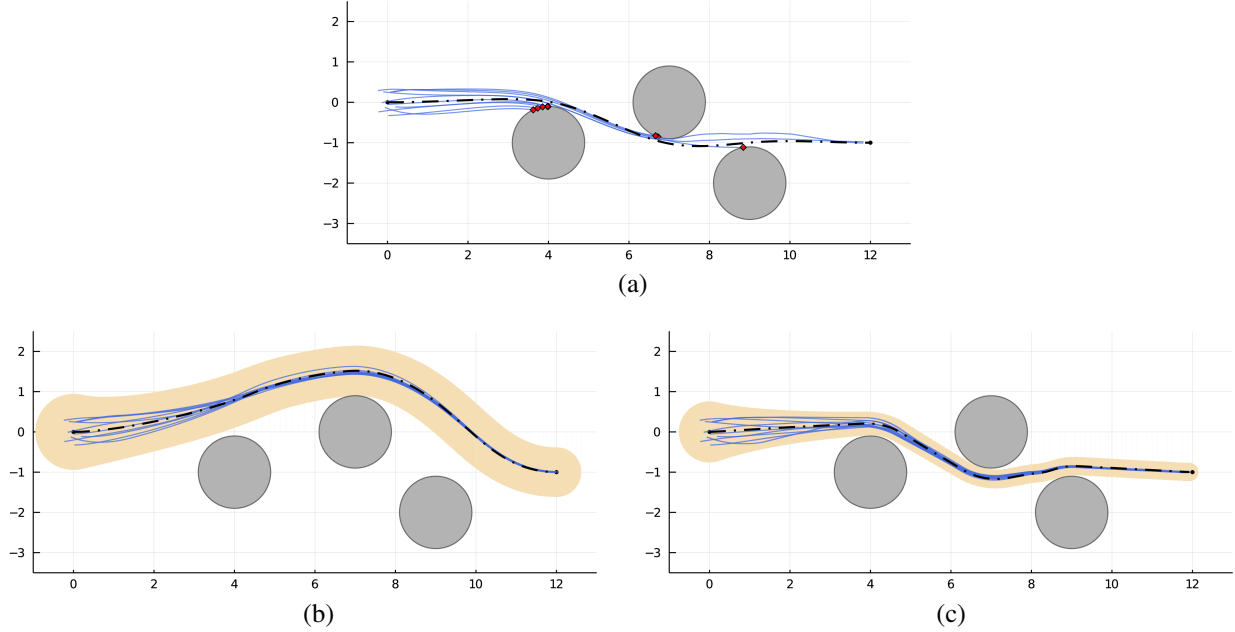


Figure 4.3: A Dubin’s car traversing an obstacle forest using (a) CCM-based feedback (b)  $\mathcal{CL}_1$  control, and (c) our framework. The system trajectories are denoted in blue, the obstacles in gray, the planned trajectory in dashed black, and the tubes are shown as shaded orange regions. Any collisions with an obstacle terminate the trajectory immediately and are indicated as red diamonds.

**Dubin’s Car:** The vehicle dynamics are described by  $(p_x, p_y, \theta, v)$ , where  $p_x$  and  $p_y$  are positions,  $\theta$  is the heading angle, and  $v$  is the velocity. The system has two control inputs that act on  $(\dot{\theta}, \dot{v})$ . The vehicle is tasked with traversing an obstacle forest from positions  $(0, 0)$  to  $(12, 0)$ , and a desired trajectory is planned using the augmented Lagrangian trajectory optimizer (ALTRO) presented in [40] while minimizing a quadratic cost on the states and controls. Let the system be randomly initialized around the origin and experience an unknown parasitic drag force given by  $0.1v^2$ . In Fig. 4a, a CCM-based feedback strategy is applied without concern for the uncertainty affecting the system. Out of the ten random initial conditions only two trajectories successfully reach the goal position, whereas in the majority of the simulations the vehicle collides with one of the obstacles before completing the task. With a conservative knowledge of the bounds on the uncertainty and its growth, a  $\mathcal{CL}_1$  control is designed so that the system trajectories can be guaranteed to remain inside

of a tube computed using the UUB in (4.12), as shown in Fig. 4b. As the uncertainty is learned following our approach, the bounds on the remainder uncertainty collapse with high probability as given in Theorem 4.1. Figure 4c shows improved performance certificates in the form of the tightened tubes. Furthermore, the learned estimates are incorporated into the planner and  $\mathcal{CL}_1$  architecture through the learned dynamics as (4.6). This example shows the clear improvement in the performance-dependent optimality and enables model-based optimality by incorporating  $\bar{F}$  into the planner, while ensuring safety.

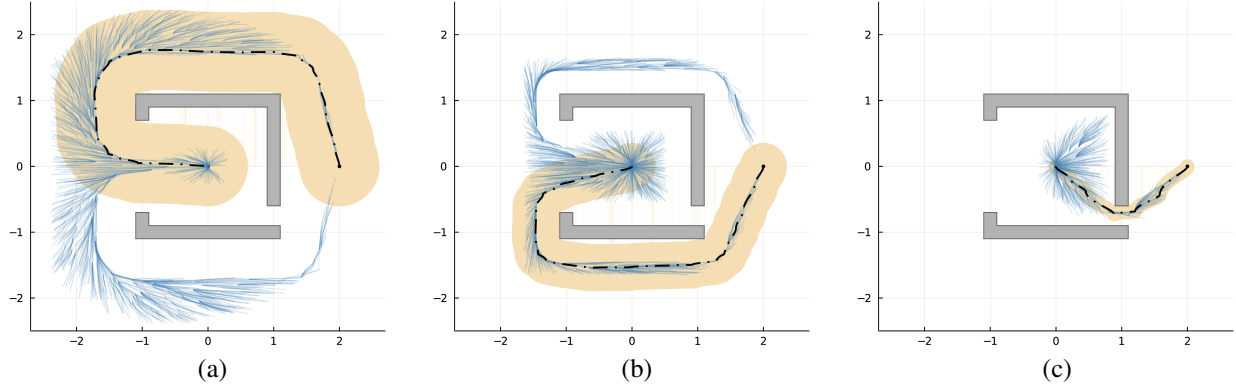


Figure 4.4: A planar quadrotor escaping a bug trap using (a) only a deterministic knowledge of the uncertainty, (b) model learned with  $N = 25$  dataset, (c) model learned with  $N = 100$  dataset. The blue lines indicate the edges of the random geometric graph constructed by BIT\*. The dashed-black line indicates the lowest cost trajectory found by batch informed trees (BIT\*).

**Planar Quadrotor:** The vehicle dynamics are described by  $(p_x, p_y, v_x, v_y, \theta, \dot{\theta})$ , where  $p_x$  and  $p_y$  are positions,  $v_x$  and  $v_y$  are velocities, and  $\theta$  and  $\dot{\theta}$  are the pitch angle and rate respectively. The system has two control inputs: the thrust and pitch moment commands that act on  $(\dot{v}_y, \dot{\theta})$ . The vehicle starts in a room at position  $(0, 0)$  and is tasked with planning a trajectory that takes it to  $(2, 0)$  safely. For such problems, complete or probabilistically complete planners are the algorithms of choice, since other methods typically get stuck at a local minimum and never reach the goal. We use the popular sampling-based planner BIT\* [142], where the two-point boundary value problem is solved using ALTRO. Similar to the previous example, consider that the system experiences an unknown parasitic drag force given by  $-0.1(v_x^2 + v_y^2)$  and a constant unknown offset in the thrust command. Planning without taking into account these uncertainties might generate trajectories that drive the system into regions that are unsafe. However in our framework, the uniform performance guarantees (4.11), provided by  $\mathcal{CL}_1$  control, ensure that BIT\* only samples states that lead to provably safe trajectories, Fig. 5a. Figures 5b and 5c show the tightening of the tubes as the uncertainty is learned batch-wise following our approach. This allows BIT\* to construct a graph of safe trajectories with improved performance guarantees.

## Chapter 5

# Certifying Feedback Policies using Interval Analysis

### 5.1 Introduction

Machine learning (ML) methods have proven effective at learning nonlinear dynamic models and controllers [143, 144]. Despite the impressive empirical performance of ML-based control, obtaining formal certificates for stability and convergence remains challenging. A further challenge is the problem of jointly learning certificates and control policies. The lack of quantified safety margins has relegated the operation of robots with ML-based dynamics and controllers, for the time being, to controlled environments so that the consequences of failure remain trivial.

Classical control techniques provide means of synthesizing control policies and certificates using Lyapunov theory [80]. Lyapunov theory establishes the stability and convergence properties of dynamical systems to an equilibrium point by introducing a Lyapunov function (LF) that acts as a surrogate for the ‘energy’ of a system. Studying the LF’s dissipation along the descriptive vector field allows one to conclude the system’s stability properties. A similar approach is used for the control policy synthesis problem so that the controlled system admits an LF, thus proving its stability. One can search for LFs and control policies for finite-dimensional linear systems by solving linear matrix inequalities (LMIs) [13]. However, most robotic platforms evolve as per dynamics described by nonlinear ordinary differential equations (ODEs). The search for LFs and control policies for nonlinear systems is a challenging problem, and unlike linear systems, there is a lack of general methods to achieve the task. When ODEs with polynomial vector fields can describe the nonlinear systems, sum-of-squares (SOS) optimization has proven to be effective [145–147]. However, in addition to the constraint of requiring polynomial vector fields, SOS methods scale poorly to high-dimensional systems since they rely on solving underlying semi-definite programming problems [148]. Recent advances in deep learning techniques have enabled more general approaches wherein one searches for LFs and control policies parameterized by deep neural networks (DNNs). In [149], the authors use DNN-parameterized LFs to learn certified safe regions, and in [27] both LFs and control policies are parameterized by DNNs and used to certify a safe control policy. However, such DNN-based approaches require the verification of the candidate

solutions, which can prove problematic due to the ‘deep’ nature of the DNNs used. For example, in [149] the Lipschitz constants of the DNN-parameterized LF and the dynamics are estimated and used to verify the solution. Such verification process can prove to be overly conservative and would not scale very well to high-dimensional problems. Moreover, the estimation of the Lipschitz constants may not be rigorous leading to insufficient verification procedures. On the other hand, in [27] the authors use formal verification techniques in the form of satisfiability modulo theories (SMT) algorithms to verify the LF or report a counter-example if the condition is violated. However, dedicated SMT solvers have only been able to handle very small networks ([150]), with a few exceptions like the Reluplex SMT solver, which is applicable only to non-smooth DNNs that use the rectified linear unit (ReLU) activation functions [151]. Despite the challenges of verifying, DNN-based approaches remain promising for learning LFs and control policies. A more thorough review of DNN verification algorithms can be found in [152].

The notion of incremental stability is highly pertinent for robotics applications. As opposed to the stability of equilibria, incremental stability refers to the stability of trajectories that are realizable by a system [153] with respect to each other. The utility of incremental stability for robotics is obvious. For example, mobile robotic platforms should be capable of converging to any desired trajectory/path computed by a planner. One can search for control policies that render the system incrementally stable via either a Lyapunov approach [153] or by using contraction theory [94]. Instead of using energy functions as in Lyapunov analysis, contraction theory verifies the convergence properties using a differential framework by synthesizing contraction metrics [92]. The contraction metric synthesis problem can be cast as an SOS optimization problem and is thus amenable to solvability [111]. However, SOS optimization suffers from the drawbacks mentioned earlier. Very recently, [8] introduced a DNN based framework to search for contraction metrics and control policies jointly. While the approach proposed by [8] removes the issues of using SOS optimization, producing theoretical guarantees still requires the verification of DNN candidate solution after the optimization, which remains challenging. Leveraging Lipschitz continuity bounds can provide theoretical guarantees but suffer from the issues mentioned earlier. Thus, as an alternative, [8] also provides probabilistic guarantees for the considered deterministic system. In a similar vein, [154] proposes the use of spectrally-normalized DNNs to establish incremental stability for a class of stochastic nonlinear systems.

In contrast to methods presented above, we present a verification approach that utilizes interval arithmetic [155] to certify the trained models of the incremental Lyapunov function (ILF) (or control contraction metric (CCM)) and the control policy. As the name suggests, interval arithmetic introduces mathematical operations over real-valued intervals sets. For example, the addition between the intervals  $[1, 2]$  and  $[-1, 1]$  encompasses all possible sums of elements from these two sets given by the interval  $[0, 3]$ . Using interval arithmetic, we can compose many such simple in-

terval operations to produce interval operations over far more complex functions. We present an algorithm that searches for the interval regions of the state-space where the trained DNN models are valid by iteratively subdividing the intervals. This way we are able to recover valid regions of attraction even from poorly trained models of the certificate and controller. Therefore, unlike previous verification approaches, the trained models are not rejected simply because they are not valid over the entire domain.

## 5.2 Problem Formulation

We consider a class of known nonlinear time-invariant systems whose dynamics evolution is governed by

$$\dot{x}(t) = F(x(t), u(t)), \quad x(0) = x_0, \quad (5.1)$$

where the state  $x(t) \in \mathcal{X} \subset \mathbb{R}^n$ , the control input  $u(t) \in \mathcal{U} \subset \mathbb{R}^m$ , and the function  $F(x, u) \in \mathbb{R}^n$ . In certain situations, we may assume that the nonlinear dynamics of the system is control affine given by

$$\dot{x}(t) = F(x(t), u(t)) = f(x(t)) + B(x(t))u(t), \quad x(0) = x_0, \quad (5.2)$$

and we will make the distinction clear whenever necessary. The compact sets  $\mathcal{X}$  and  $\mathcal{U}$  are the state and input spaces, respectively. We place standard regularity assumptions to ensure the well-posedness of (5.1) [80]. We now state the definition of incremental asymptotic stability (IAS).

**Definition 5.1** ([153]). *Consider any open-loop state input pair  $(x^*, u^*)$  realizable by (5.1), i.e.,  $\dot{x}^* = F(x^*, u^*)$ . We say that the system (5.1) is IAS (or contracting) if there exists a feedback law  $k(t, x) \in \mathbb{R}^m$ , such that the state  $x$  of (5.1) with the control policy  $u = k(t, x)$ , i.e.,  $\dot{x} = F(x, k(t, x))$  satisfies*

$$\|x^*(t) - x(t)\| \leq \kappa(\|x^*(0) - x(0)\|, t), \quad \forall x(0) \in \mathcal{X}, t \geq 0.$$

*The function  $\kappa(x, t)$  belongs to class  $\mathcal{KL}$ . Note that incremental exponential stability (IES) is similarly defined if  $\kappa(x, t) := Cxe^{-\lambda t}$  for appropriate  $C, \lambda > 0$ .*

One can establish IAS properties of (5.1) by either searching for ILFs and feedback laws jointly on the ambient Euclidean space or by searching for CCMs and feedback laws jointly on the tangent bundle of the associated manifold. Here, we add the qualifier ‘ambient’ to identify the space to which the state of the system belongs to. The main results for both the approaches are summarised in the propositions below.

**Proposition 5.1** (Ambient Space IAS). *Consider any open-loop state-input pair  $(x^*, u^*)$  realizable by (5.1). Suppose there exists a feedback law  $k(t, x)$  and a function  $V : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ , satisfying*

$$a_1 \|x - x^*\|^2 \leq V(x, x^*) \leq a_2 \|x - x^*\|^2, \quad (5.3a)$$

$$L_{F(x, k(t, x))}V(x, x^*) + L_{F(x^*, u^*)}V(x, x^*) < 0, \quad (5.3b)$$

*for all  $(x, x^*, u^*, t) \in \mathcal{X} \times \mathcal{X} \times \mathcal{U} \times \mathbb{R}_{\geq 0}$  except if  $x = x^*$  and the constants  $a_1, a_2 > 0$ . Then,  $V$  is an ILF and the control policy  $u = k(t, x)$  renders the system (5.1) IAS in the sense of Definition 5.1. Note that if (5.3b) is satisfied with the addition of  $2\lambda V(x, x^*)$ , then the closed-loop system is IES.*

The proof is a trivial extension of [153, Lemma 2.3]. The alternate set of IES conditions expressed over the differential space is provided next.

**Proposition 5.2** (Differential Space IAS). *Consider any open-loop state-input pair  $(x^*, u^*)$  realizable by the control-affine dynamics in (5.2). Suppose there exists a feedback law  $k(t, x)$  and a function  $M : \mathcal{X} \rightarrow \mathbb{S}^n$  satisfying*

$$a_1 \mathbb{I}_n \preceq M(x) \preceq a_2 \mathbb{I}_n, \quad (5.4a)$$

$$\partial_{F(x, k(t, x))}M(x) + \left[ M(x) \frac{\partial F}{\partial x}(x, k(t, x)) \right]_{\mathbb{S}} \prec 0, \quad (5.4b)$$

*for all  $(x, x^*, u^*, t) \in \mathcal{X} \times \mathcal{X} \times \mathcal{U} \times \mathbb{R}_{\geq 0}$ , except if  $x = x^*$  and the constants  $a_1, a_2 > 0$ . Then,  $M(x)$  is a control contraction metric (CCM) and the control policy  $u = k(t, x)$  renders the system (5.2) IAS in the sense of Definition 5.1. Similar to Proposition 5.1, if (5.4b) is satisfied with the addition of  $2\lambda M(x)$ , then the closed-loop system is IES.*

The proof can be found in [92, Theorem 1], and a deeper review of the concepts behind contraction theory are provided in Section 3.3.

Notice that the property of IAS is only valid if the trajectories that start anywhere in the domain  $\mathcal{X}$  converge towards each other. If two trajectories start very far away from each other, one cannot reasonably expect that they will asymptotically converge to each other regardless of the distance between them. During regulation to an equilibrium point, it is common to only expect stable behavior within a region of attraction (ROA) around the equilibrium point [80, Section 8.2]. We extend this notion to hold for situations with trajectory tracking with the following definition.

**Definition 5.2** (Incremental region of attraction (IROA)). *Consider a time-varying nonlinear system  $\dot{x} = f(x, t)$  where  $x \in \mathcal{X}$ . Suppose  $X(t, x_0) \in \mathcal{X}$  defines the trajectory of the system with the initial condition  $x_0$ , then we define the IROA as the following set*

$$\mathcal{D} := \{(x, x^*) \in \mathcal{X} \times \mathcal{X} : \lim_{t \rightarrow \infty} X(t, x) \rightarrow X(t, x^*)\}.$$

The set essentially defines a tube around any desired trajectory, such that all system trajectories originating inside the tube asymptotically converge to the desired trajectory. This is less restrictive than the previous definition of the IAS wherein all trajectories start in a set converge to one another. We restate the definition of IAS so that only *nearby* trajectories converge to each other.

**Definition 5.3** (Relaxed IAS ). *Consider any open-loop state input pair  $(x^*, u^*)$  realizable by (5.1), i.e.,  $\dot{x} = F(x^*, u^*)$ . We say that the system (5.1) is IAS (or contracting) if there exists a feedback law  $k(t, x) \in \mathbb{R}^m$ , such that the state  $x$  of (5.1) with the control policy  $u = k(t, x)$ , i.e.,  $\dot{x} = F(x, k(t, x))$  satisfies*

$$\|x^*(t) - x(t)\| \leq \kappa(\|x^*(0) - x(0)\|, t),$$

for all  $t \geq 0$  and all  $x(0) \in \mathcal{X}$  such that

$$(x(0), x^*(0)) \in \mathcal{D}.$$

The function  $\kappa(x, t)$  belongs to class  $\mathcal{KL}$ . Note that IES is similarly defined if  $\kappa(x, t) := Cxe^{-\lambda t}$  for appropriate  $C, \lambda > 0$ .

Similarly, the definition of the ILF can be relaxed such that the Lie derivative conditions only hold over the IROA set  $\mathcal{D}$ .

**Proposition 5.3** (Relaxed Ambient Space IAS). *Consider any open-loop state-input pair  $(x^*, u^*)$  realizable by (5.1). Suppose there exists a feedback law  $k(t, x)$  and a function  $V : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ , satisfying*

$$a_1\|x - x^*\|^2 \leq V(x, x^*) \leq a_2\|x - x^*\|^2, \quad (5.5a)$$

$$L_{F(x, k(t, x))}V(x, x^*) + L_{F(x^*, u^*)}V(x, x^*) < 0, \quad (5.5b)$$

for all  $t \geq 0$ ,  $u^* \in \mathcal{U}$ , and

$$(x, x^*) \in \mathcal{D},$$

except if  $x = x^*$  and the constants  $a_1, a_2 > 0$ . Then,  $V$  is an ILF and the control policy  $u = k(t, x)$  renders the system (5.1) IAS in the sense of Definition 5.1. Note that if (5.5b) satisfied with the addition of  $2\lambda V(x, x^*)$ , then the closed-loop system is IES.

However, the definition of CCM cannot be extended in a similar fashion because the notion of distances between trajectories is lost in the differential space. One workaround, that will be explored in a later section, is utilize a control Lyapunov function (CLF) associated with the CCM such as the Riemannian energy function such that the CLF is only valid in  $\mathcal{D}$ .



**Problem Statement:** Given the system dynamics (5.1), the goal is to design a feedback law  $k(t, x)$ , the associated certificate (ILF or CCM), and the IROA  $\mathcal{D}$  such that the closed-loop system is IAS with respect to  $\mathcal{D}$ , as defined in Definition 5.3.

## 5.3 Preliminaries on Interval Arithmetic

Interval arithmetic provides a useful mathematical framework to analyze the behavior of functions over closed intervals. For  $a, b \in \mathbb{R}$  such that  $a < b$ , we define an interval  $[a, b]$  as the following compact set

$$[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}.$$

We can now define operations over these intervals. A binary operation  $\circ$  on two intervals  $[a, b]$  and  $[c, d]$  is defined as

$$[a, b] \circ [c, d] := \left[ \{x \circ y : x \in [a, b], y \in [c, d]\} \right],$$

where  $[\cdot]$  denotes an interval enclosure over the set argument. For example, addition and subtraction over intervals are defined as

$$\begin{aligned} [a, b] + [c, d] &:= [a + c, b + d], \\ [a, b] - [c, d] &:= [a - d, b - c]. \end{aligned}$$

Similarly, we can define interval operations over a function  $g : \mathbb{R} \rightarrow \mathbb{R}$  as

$$g([a, b]) = \left[ \{g(x) : x \in [a, b]\} \right].$$

For example, the exponential over an interval is defined as

$$e^{[a, b]} := [e^a, e^b].$$

However, not all functions will have such simple representations. The motivation behind interval arithmetic is that rigorous interval enclosures can be defined for complicated functions by composing several elementary functions and operations. This enables interval arithmetic to even operate over discontinuous ranges, e.g. approximations caused by floating-point arithmetic on computer hardware [156]. However, sometimes the composition over many such functions may cause the interval enclosure to be overly conservative. In such situations, if the function  $g \in \mathcal{C}^1$ , then the

elementary mean-value form [155] may provide a tighter interval enclosure

$$g([a, b]) = g\left(\frac{a+b}{2}\right) + g'([a, b])\left(\left[\frac{a-b}{2}, \frac{b-a}{2}\right]\right), \quad (5.6)$$

where  $g'$  indicates the derivative of  $g$ . For functions with continuity in higher-order derivatives, elementary Taylor forms [157] may provide even tighter interval enclosures. It is also important to note that intervals can also be defined at higher dimensions. Consider  $a, b \in \mathbb{R}^n$  such that  $a < b$  element-wise; then we define vector intervals as

$$[a, b] := \{x \in \mathbb{R}^n : a_k \leq x_k \leq b_k, \quad \forall k\},$$

where  $\cdot_k$  denotes the  $k^{\text{th}}$  element of a vector. Matrix intervals for  $A, B \in \mathbb{R}^{n \times m}$ , where  $A < B$  element-wise, are also similarly defined as

$$[A, B] := \{X \in \mathbb{R}^{n \times m} : A_{i,j} \leq X_{i,j} \leq B_{i,j}, \quad \forall i, j\},$$

where  $\cdot_{i,j}$  denotes the element at the  $(i, j)^{\text{th}}$  index of a matrix. The elementary mean-value and Taylor forms are also extendable to functions on higher dimensional intervals.

## 5.4 Certifying Incremental Lyapunov Functions

In this section, we address the problem of synthesizing a feedback law  $k$  and an ILF  $V(x, x^*)$  such that the induced control policy renders (4.1) IES. As we mentioned previously, a straightforward approach, in principle, is to directly search for feedback laws and ILF certificates for the system in (4.1). In this approach, one attempts to solve a feasibility problem: does there exist a feedback law  $k$  and an ILF  $V$  such that  $\dot{x} = F(x, k)$  is IES with certificate  $V$ ? Of course, the feasibility problem is not guaranteed to terminate in the affirmative. Moreover, the ability to find a solution depends on the search space and the search algorithms. Using DNNs, one can generate rich search spaces and improve the chances of terminating the feasibility problem with a valid solution. In the literature, most work on jointly searching for feasible feedback laws and certificates with DNNs terminate with an unverified candidate solution. Therefore, an additional step of verification is required to check the candidate solutions' validity, which is not a trivial task. For example, in [27], the authors use DNNs to produce candidate solutions and then verify them with SMT solvers. If the solution is invalid, it is added to a counterexample set, and the iterative process is repeated. This requires multiple runs of DNN optimization and SMT verification. Even so, there is no guarantee that the feasibility problem admits a solution in the first place, thus rendering the search problem

ineffectual.

Motivated by these challenges, we propose an alternate approach by constructing regions where the ILF Lie derivative decrease condition from (5.5b) holds. The valid regions are discovered by iteratively subdividing interval sets of the state space and validating if interval set does or does not satisfy (5.5b). We begin by first defining the space of candidate ILFs and feedback laws.

**Definition 5.4.** A function  $V : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is a valid ILF candidate if  $V \in \mathcal{V}$ , where

$$\mathcal{V} = \{p \in \mathcal{C}^2(\mathcal{X} \times \mathcal{X}; \mathbb{R}) \mid \exists a_1, a_2 > 0 \text{ s.t. } p \text{ satisfies (5.5a)}\}. \quad (5.7)$$

Similarly, a function  $k : \mathcal{U} \times \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^m$  is a valid feedback law candidate if  $k \in \mathcal{K}$ , where

$$\mathcal{K} = \{q \in \mathcal{C}^1(\mathcal{U} \times \mathcal{X} \times \mathcal{X}; \mathbb{R}^m) \mid x = x^* \implies q(u^*, x^*, x) = u^*\}. \quad (5.8)$$

We require continuity on the derivatives of the candidate ILFs and feedback laws so that we can employ the mean-value forms (5.6) when computing the interval enclosures of the Lie derivative. With the candidacy definitions in place, the first step is to parameterize the spaces  $\mathcal{V}$  and  $\mathcal{K}$  by DNNs. For the ILF consider DNN of the form

$$V_\theta(x, x^*) = (\alpha_\theta(x, x^*) - \alpha_\theta(x^*, x^*))^2 + a_1 \|x - x^*\|^2, \quad (5.9)$$

where  $\alpha_\theta(x, x^*) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is a feedforward neural network with activation functions at least in  $\mathcal{C}^2$  (e.g. tanh, sigmoid, softplus [158], swish [159], mish [160]) and parameters denoted as  $\theta$ . This parameterization ensures that  $V_\theta(x^*, x^*) = 0$  and  $V_\theta(x, x^*) \geq a_1 \|x - x^*\|^2$  for all  $x, x^* \in \mathcal{X}$ . Additionally, since the activation functions of  $\alpha_\theta$  are well behaved, the outputs of the DNN are bounded for all  $x \in \mathcal{X}$ . This implies that there exists a  $a_2 > 0$  such that (5.5a) holds.

**Remark 5.1.** There are other parameterizations [149, 161] of the LF in literature that also ensure the positivity condition from (5.5a). In [149], the weights of the DNN are constructed in such a way to always be full-column rank and no biases are included in the model. In [161], the LFs are guaranteed to be convex by structuring the DNN to be an input convex neural network [162]. Either forms may be used within our framework, but we introduce our parameterization in (5.9) because of its relative simplicity in construction.

For the controller consider the DNN of the form

$$k_\phi(u^*, x^*, x) = u^* + \beta_\phi(x^*, x) - \beta_\phi(x^*, x^*), \quad (5.10)$$

where  $\beta_\phi(x^*, x) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is a feedforward neural network with at least continuously differentiable activation functions and parameters denoted as  $\phi$ . This parameterization ensures that

when  $x = x^*$ , we have  $k_\phi(u^*, x^*, x) = u^*$ . We now proceed with the design of the loss function to train the networks. Recall that for the closed-loop system to be IES, the condition in (5.5b) must be satisfied. Therefore, one may wish to minimize the following *ambient* loss function

$$\mathcal{L}_A(\theta, \phi) = \mathbb{E}_{(u^*, x^*, x) \sim \rho(\mathcal{B})} \text{ReLU}[\ell(u^*, x^*, x)], \quad (5.11)$$

where  $\rho(\mathcal{B})$  is the uniform distribution over the set  $\mathcal{B} = \mathcal{U} \times \mathcal{X} \times \mathcal{X}$ , and  $\ell$  is given by

$$\ell(u^*, x^*, x) = L_{F(x, k(u^*, x^*, x))} V(x, x^*) + L_{F(x^*, u^*)} V(x, x^*). \quad (5.12)$$

If instead we need IES, then we may instead define  $\ell$  as

$$\ell(u^*, x^*, x) = L_{F(x, k(u^*, x^*, x))} V(x, x^*) + L_{F(x^*, u^*)} V(x, x^*) + 2\lambda V(x, x^*), \quad (5.13)$$

for some hyperparameter  $\lambda > 0$ . Additional terms in the loss function to capture more desirable closed-loop behavior, such as data collected from demonstrations or expert controllers, can be readily used in our framework.

In order to rigorously assess the violation or validation of (5.5b), we construct interval sets in the control input and state space defined as

$$[x] = \{z \in \mathcal{X} : \underline{x} < z < \bar{x}\}, \quad (5.14a)$$

$$[x^*] = \{z \in \mathcal{X} : \underline{x}^* < z < \bar{x}^*\}, \quad (5.14b)$$

$$[u^*] = \{z \in \mathcal{U} : \underline{u}^* < z < \bar{u}^*\}, \quad (5.14c)$$

where the inequalities hold element-wise and the constants satisfy  $\underline{x} < \bar{x}$ ,  $\underline{x}^* < \bar{x}^*$ , and  $\underline{u}^* < \bar{u}^*$ . While (5.12) (or (5.13)) may simply be used to evaluate the interval enclosure over the interval sets in (5.14a), we observe tighter enclosures with the mean-value form (5.6). For completeness we provide the mean-value form of  $\ell$  given by

$$\begin{aligned} \ell([u^*], [x^*], [x]) &= \ell(u_m^*, x_m^*, x_m) + \nabla_{u^*} \ell([u^*], [x^*], [x])([u^*] - u_m^*) \\ &\quad + \nabla_{x^*} \ell([u^*], [x^*], [x])([x^*] - x_m^*) + \nabla_x \ell([u^*], [x^*], [x])([x] - x_m), \end{aligned} \quad (5.15)$$

where  $[\cdot]_m$  indicates the midpoint value of the interval. The gradients  $\nabla_{u^*} \ell$ ,  $\nabla_{x^*} \ell$ ,  $\nabla_x \ell$  can be easily computed using automatic differentiation methods [163, 164], and the forward pass of these functions can be computed using interval arithmetic libraries [165, 166]. Now we proceed to describe our procedure for verifying the ILF, Algorithm 5.

The algorithm takes as input the interval set approximations of  $\mathcal{X}$  and  $\mathcal{U}$ . As the output of the

---

**Algorithm 5** ILF Verification Procedure

---

```
1: Input
2:    $\mathcal{U}$    Operating input space set
3:    $\mathcal{X}$    Operating state space set
4: Output
5:    $\gamma_{\text{roa}}$  ROA sublevel set value
6: function ILFVERIFY( $\mathcal{U}, \mathcal{X}, \gamma_{\text{roa}}^{\text{init}}$ )
7:   queue  $\leftarrow \{(\mathcal{X}, \mathcal{X})\}$ 
8:   repeat
9:      $[x^*], [x] \leftarrow \text{pop}(\text{queue})$ 
10:    if  $\sup(\ell(\mathcal{U}, [x^*], [x])) < 0$  or  $\sup(\|[x] - [x^*]\|) \leq \epsilon$  then ▷ do nothing
11:    else
12:       $([x^*]_a, [x]_a), ([x^*]_b, [x]_b) \leftarrow \text{bisect}([x^*], [x])$ 
13:      queue  $\leftarrow \text{sortedinsert}(\text{queue}, ([x^*]_a, [x]_a), \inf(V_\theta([x]_a, [x^*]_a)))$ ,
14:      queue  $\leftarrow \text{sortedinsert}(\text{queue}, ([x^*]_b, [x]_b), \inf(V_\theta([x]_b, [x^*]_b)))$ ,
15:    end if
16:    until maximum number of iterations or  $([x^*], [x]) \in \partial(\mathcal{X} \times \mathcal{X})$ 
17:     $[x_{\text{nearest}}^*], [x_{\text{nearest}}] \leftarrow \text{pop}(\text{queue})$ 
18:     $\gamma_{\text{roa}} \leftarrow \inf(V_\theta([x_{\text{nearest}}], [x_{\text{nearest}}^*]))$ 
19:    return  $\gamma_{\text{roa}}$ 
20: end function
```

---

verification, the algorithm reports a scalar value  $\gamma_{\text{roa}}$ , such that  $\mathcal{D}_{\gamma_{\text{roa}}} \subseteq \mathcal{D}$  which forms a subset of the true IROA given by

$$\mathcal{D}_{\gamma_{\text{roa}}} = \{(z, z^*) \in \mathcal{X} \times \mathcal{X} : V(z, z^*) < \gamma_{\text{roa}}\}. \quad (5.16)$$

Unlike the traditional ROA, this forms a ROA for every desired point  $z^* \in \mathcal{X}$ . Through our computation we rigorously ensure that for all  $(z, z^*) \in \mathcal{D}_{\gamma_{\text{roa}}}$  the condition in (5.5b) holds. At the beginning of our verification, in line 7 a queue is initialized with the interval region  $\mathcal{X} \times \mathcal{X}$ . At each iteration, the algorithm pops the element at the front of the queue for analysis. If the interval enclosure over  $\ell$  is negative or if the interval is inside a small  $\epsilon$ -ball around  $x = x^*$ , then that interval is considered valid and no longer tracked in the procedure, line 10. If the interval is not valid, then it is bisected, usually along its longest edge, and inserted back into the queue, lines 12-14. When inserting the bisected intervals back into the queue, the sorting order must be retained based on the infimum over the interval enclosure of the ILF computed at those intervals. After either the maximum number of iterations are reached or an element at the boundary of the set  $(\mathcal{X} \times \mathcal{X})$  is encountered, the iterations are stopped, line 16. Encountering an element at the boundary of the set implies that the largest sublevel set of the ILF in the interior  $\mathcal{X} \times \mathcal{X}$  has been validated. Since any larger sublevel sets will exceed the interval boundary the solver interrupts the progress

of the algorithm and returns the  $\gamma_{\text{roa}}$  value at this location. On the other hand the iterations may be stopped if it exceeds some specified time limit or maximum number of iterations. The sorting order of the queue implies that the first element has to have the smallest under-approximation of ILF. The minimum between this value and  $\gamma_{\text{roa}}^{\text{init}}$  is returned as  $\gamma_{\text{roa}}$ , line 18.

**Remark 5.2.** *While Algorithm 5 is presented as serial procedure, it can be also evaluated batch-wise if necessary. On specific computer hardware and low-level routines [167] it may be more efficient to process the forward-pass of the neural networks in a batch-wise manner.*

**Remark 5.3.** *Algorithm 5 does not check in an  $\epsilon$ -ball around the region defined by  $x = x^*$ , because the strict inequality in (5.5b) is relaxed. This type of check is also performed by the SMT-based verification method presented in [27].*

**Remark 5.4.** *The intervals are searched over  $\mathcal{X} \times \mathcal{X}$ , which may become cumbersome for high dimensional systems. In such situations, adding structure to the problem may reduce the computational burden. For example, one can parameterize the ILF to depend on the error dynamics rather than  $x$  and  $x^*$  independently.*

## 5.5 Certifying Control Contraction Metrics

In this section, we address the problem of synthesizing a feedback law  $k$  and a CCM  $M(x)$  such that the induced control policy renders (5.2) IES. Recall that unlike the previous section where we considered any general nonlinear system, when searching for CCMs we are restricted to control-affine systems. We begin by first defining the space of candidate ILFs and feedback laws.

**Definition 5.5.** *A function  $M : \mathcal{X} \rightarrow \mathbb{S}_{>0}^n$  is a valid CCM candidate if  $M \in \mathcal{M}$ , where*

$$\mathcal{M} = \{p \in \mathcal{C}^2(\mathcal{X}; \mathbb{S}_{>0}^n) \mid \exists a_1, a_2 > 0 \text{ s.t. } p \text{ satisfies (5.4a)}\}. \quad (5.17)$$

*The space of feedback laws is defined exactly as (5.8).*

With the candidacy definitions in place, the first step is to parameterize the space  $\mathcal{M}$  by DNNs. We use the parameterization proposed in [8] and present it here for completeness. For the CCM, consider DNN of the form

$$M_\theta(x, x^*) = \alpha_\theta(x)^\top \alpha_\theta(x) + a_1 \mathbb{I}_n, \quad (5.18)$$

where  $\alpha_\theta(x) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^{n \times n}$  is a feedforward neural network with activation functions at least in  $\mathcal{C}^2$  and parameters denoted as  $\theta$ . The matrix output of the DNN can be obtained by reshaping a vector of  $\mathbb{R}^{n^2}$  to a matrix. For the feedback law we specify the same parameterization as in (5.10).

We now proceed with the design of the loss function to train the networks. Recall that for the closed-loop system to be IES the condition in (5.4b) must be satisfied. Therefore, one may wish to minimize the following *ambient* loss function

$$\mathcal{L}_D(\theta, \phi) = \mathbb{E}_{(u^*, x^*, x) \sim \rho(\mathcal{B})} \text{ReLU} \left[ \bar{\lambda}(\ell(u^*, x^*, x)) \right], \quad (5.19)$$

where  $\rho(\mathcal{B})$  is the uniform distribution over the set  $\mathcal{B} = \mathcal{U} \times \mathcal{X} \times \mathcal{X}$ ,  $\bar{\lambda}$  denotes the largest eigenvalue, and  $\ell$  is given by

$$\ell(u^*, x^*, x) = \partial_{F(x, k(t, x))} M(x) + \left[ M(x) \frac{\partial F}{\partial x}(x, k(t, x)) \right]_{\mathbb{S}}. \quad (5.20)$$

If instead we need IES, then we may instead define  $\ell$  as

$$\ell(u^*, x^*, x) = \partial_{F(x, k(t, x))} M(x) + \left[ M(x) \frac{\partial F}{\partial x}(x, k(t, x)) + 2\lambda M(x) \right]_{\mathbb{S}}, \quad (5.21)$$

for some hyperparameter  $\lambda > 0$ . The losses defined above are the minimum requirements to adequately train the CCM, but as shown in [8], additional regularization terms improve the training process.

Similar to the previous section, we use the mean-value form (5.6) to describe the interval enclosures over  $\ell$ , given by (5.15).

However unlike the previous section, we have to solve an eigenvalue problem over interval matrices. Eigenvalue problems over real symmetric interval matrices are a well studied problem with algorithms that provide both exact [168] and overapproximate [169] interval enclosures of possible eigenvalues. In this work we use the results from [169], where it was shown that for a real symmetric interval matrix  $[A]$  the largest eigenvalue is enclosed in the interval

$$\bar{\lambda}([A]) = [\bar{\lambda}([A]_m) - \|[A]_r\|, \bar{\lambda}([A]_m) + \|[A]_r\|], \quad (5.22)$$

where  $[\cdot]_m$  indicates the midpoint value of the interval and  $[\cdot]_r$  indicates the radius of the interval. Recall from Section 3.3 that the Riemannian energy is the ILF representation of CCM in the ambient space. Using the mean-value form (5.6) and the result from Lemma B.4, we can specify the interval enclosure over the Riemannian energy as

$$\mathcal{E}([x], [x^*]) = \mathcal{E}([x]_m, [x^*]_m) \pm [a_2 \sup(\|[x] - [x^*]\|)] (\|[x]_r\| + \|[x^*]_r\|), \quad (5.23)$$

where  $a_2$  is the constant from (5.4a). For verifying CCM, we use Algorithm 5 with the following differences. First, the condition being checked in line 10 is given by  $\bar{\lambda}(\ell([u^*], [x^*], [x]))$ . Second,

instead of computing interval enclosures over the ILF in lines 13,14,18 we compute the interval enclosures over the Riemannian energy as specified in (5.23). Similar to certifying an ILF, the algorithm reports a scalar value  $\gamma_{\text{roa}}$  such that  $\mathcal{D}_{\gamma_{\text{roa}}} \subseteq \mathcal{X}$  forms the IROA given by

$$\mathcal{D}_{\gamma_{\text{roa}}} = \{(z, z^*) \in \mathcal{X} \times \mathcal{X} : \mathcal{E}(z, z^*) < \gamma_{\text{roa}}\}, \quad (5.24)$$

where the region is specified using the sublevel sets of the Riemannian energy function.

## 5.6 Robustness to Disturbances

Suppose that the system in (5.1) is perturbed by some disturbance  $w \in \mathcal{W} \subset \mathbb{R}^w$  given by

$$\dot{x}(t) = F(x(t), u(t), w(t)), \quad x(0) = x_0. \quad (5.25)$$

In such situations, the presented approach for certifying the IROA can also be used to estimate an incremental robust positively invariant set (RPI) set defined as the compact set  $\Omega \subset \mathcal{X} \times \mathcal{X}$  such that if  $(x_0, x_0^*) \in \Omega$  then the trajectories  $(X(t, x_0), X(t, x_0^*)) \in \Omega$  for all  $t \geq 0$ . The difference between the RPI set and the IROA is that the system trajectories do not have to converge to each other but instead only that they must remain close to each other. Similar to IROA, the RPI set can be under-approximated as

$$\Omega_{\gamma_{\text{pi}}} = \{(z, z^*) \in \mathcal{X} \times \mathcal{X} : V(z, z^*) < \gamma_{\text{pi}}\}. \quad (5.26)$$

The Lie derivative expression for the perturbed system is given by

$$\ell(u^*, x^*, x, w) = L_{F(x, k(u^*, x^*, x), w)} V(x, x^*) + L_{F(x^*, u^*)} V(x, x^*).$$

When the decrease condition is verified over some intervals of  $u^*$ ,  $x^*$  and  $x$  in line 10 of Algorithm 5, the above condition is checked over all possible disturbances  $\mathcal{W}$  defined by an interval set. The advantage of such a computational analysis is that the formulation works for any deterministic representation of the uncertainty (e.g. parametric, additive, etc.) without adding any conservatism that usually appears when such analyses are carried out by hand.



## 5.7 Numerical Examples

We now present numerical experimentation results by applying the proposed methodology on a torque-limited inverted pendulum. The dynamics of the pendulum is given by

$$\ddot{\theta}(t) = \frac{mgL \sin(\theta(t)) - \mu \dot{\theta}(t) + u}{mL^2},$$

where  $m = 0.15\text{kg}$  is the mass,  $L = 0.5\text{m}$  is the half-length,  $g = 9.8\text{m/s}^2$  is the gravitational constant, and  $\mu = 0.1\text{Nms/rad}$  is the friction coefficient. The torque is limited to  $\pm 0.6\text{Nm}$ , which implies that if the pendulum is starting from outside of  $\approx \pm 55^\circ$ , then no amount of constant torque will bring the pendulum to its inverted position. Typically such problem requires energy shaping [170] or trajectory optimization-based solutions.

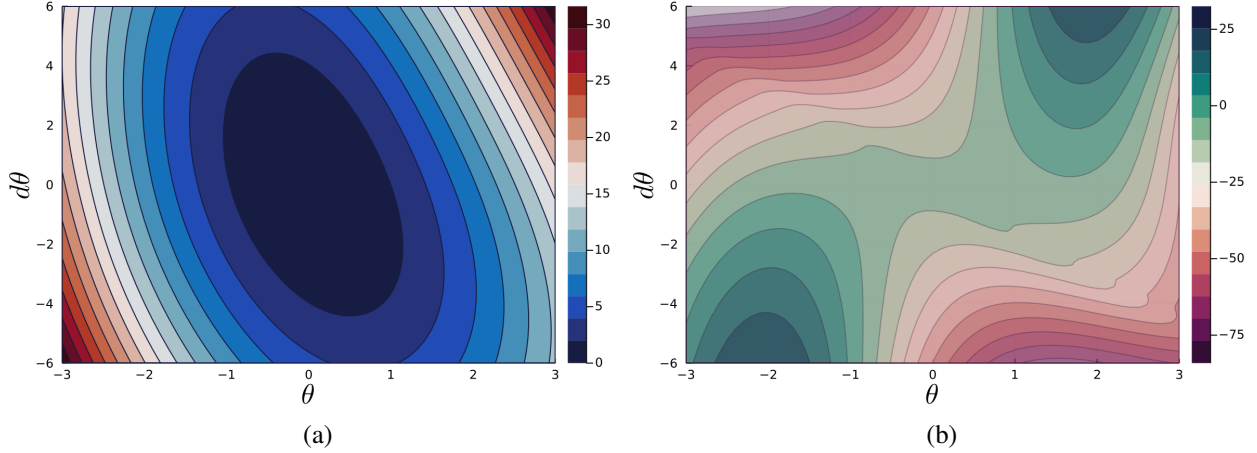


Figure 5.1: The learned LF  $V_\theta$  (a) and the Lie derivative  $L_F V_\theta$  (b).

We construct a LF using a DNN with 3 layers and 32 neurons on each hidden layer with softplus activation functions on each layer except the output layer. The feedback law is a DNN with 2 layers and 32 neurons on the hidden layer with a tanh activation function. A training dataset of 32768 states was uniformly sampled from  $[-3, 3] \times [-6, 6]$  with a batch size of 512 states. The ADAM optimizer was used to train the network parameters with a learning rate of 0.001. The training process completed in 9 epochs. The learned LF and the Lie derivative of the LF with respect to the closed-loop system are shown in Fig. 5.1.

We proceed to validate the learned LF over  $\mathcal{X}$  and obtain the ROA. The computation follows Algorithm 5 and was carried out for 5000 iterations. The following region of attraction was verified

$$\mathcal{D}_{\gamma_{\text{roa}}} = \{z \in \mathcal{X} : V(z) < 1.58\}.$$

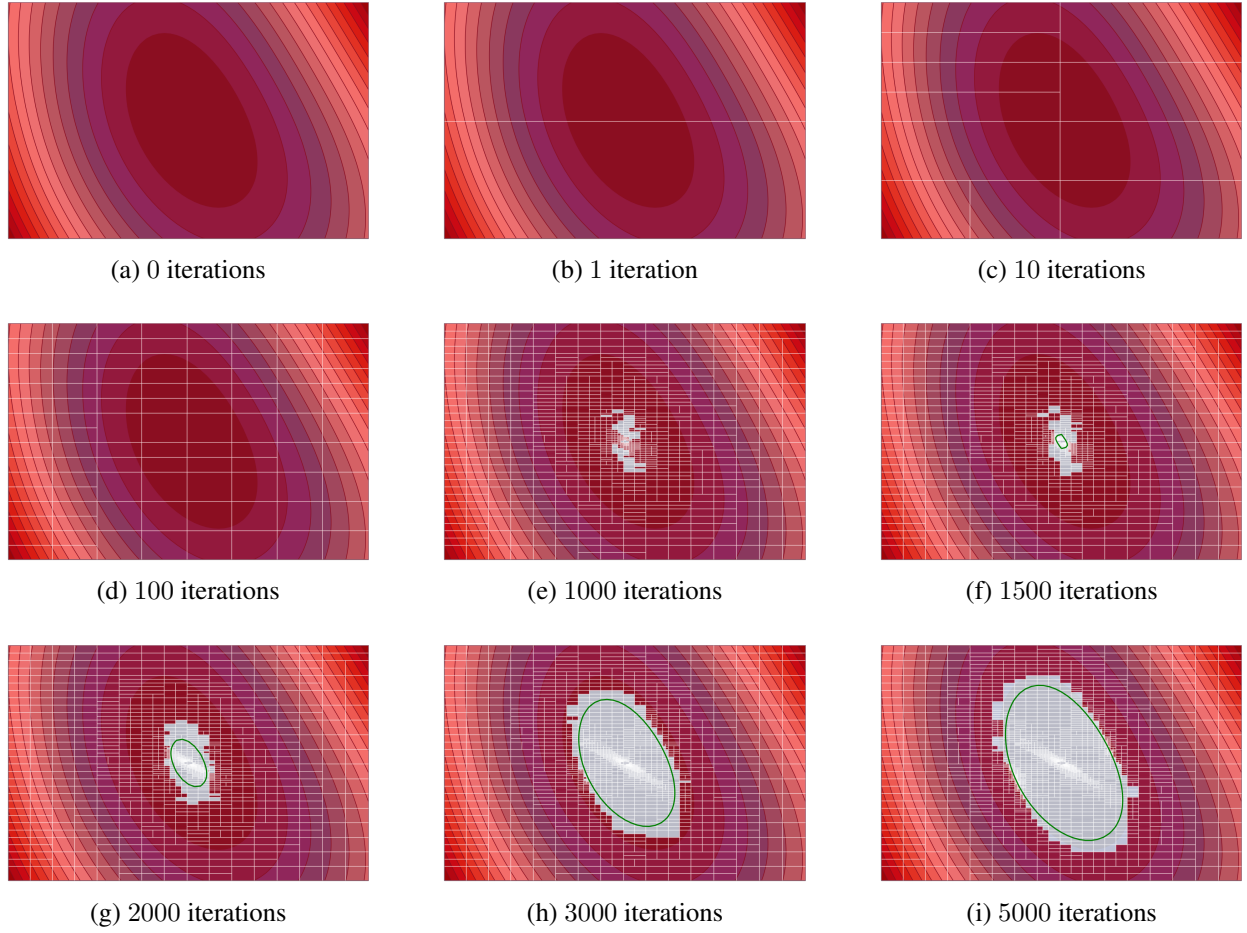


Figure 5.2: A visual representation of the verification process at different iterations overlaid over the contour plot of the learned LF. The (translucent) white boxes represent the intervals that were validated according to line 10 in Algorithm 5. The red boxes indicate the intervals that are still in the queue. The green boundary denotes the boundary of the verified ROA.

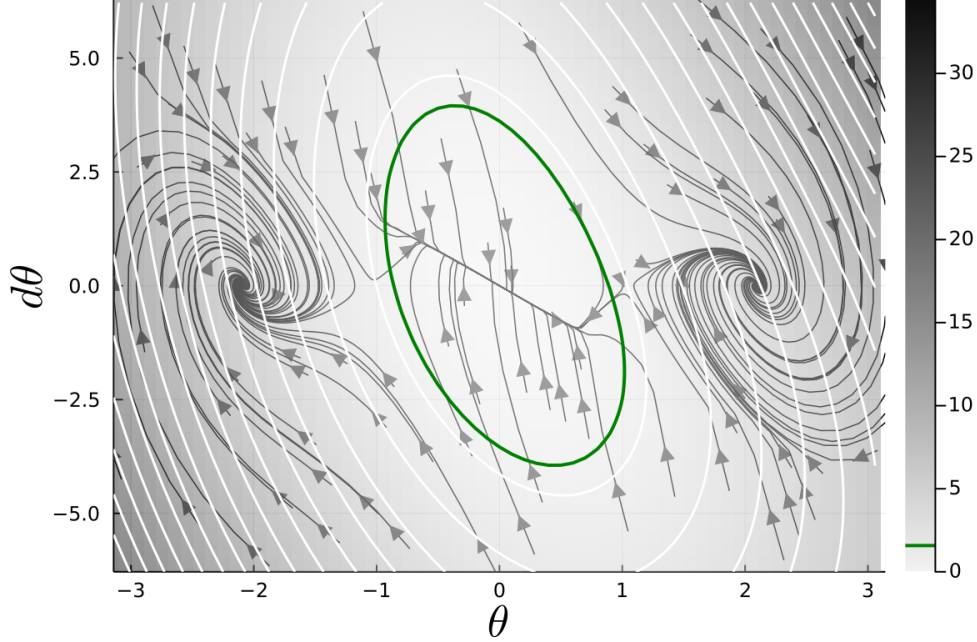


Figure 5.3: All trajectories that start or enter the ROA determined by the green boundary remain in the region and converge to the equilibrium asymptotically. The black lines denote the trajectories of the system that are randomly initialized in  $\mathcal{X}$ .

and is represented as a green ellipse-like region in Fig. 5.2i. In Fig. 5.3 we numerically verify that the certified ROA is indeed correct by simulating 100 trajectories in  $\mathcal{X}$  and observing that the trajectories that enter  $\mathcal{D}_{\gamma_{\text{roa}}}$  do not leave it and converge asymptotically to the origin.

Suppose that the system is perturbed as

$$\ddot{\theta}(t) = \frac{mgL \sin(\theta(t)) - \mu \dot{\theta}(t) + u}{mL^2} + w(t),$$

where  $w(t)$  is an additive disturbance such that  $|w(t)| \leq 0.2$  for all  $t \geq 0$ . Following Section 5.6, the learned LF can be validated over  $\mathcal{X}$  and  $\mathcal{W}$  to obtain the following RPI set:

$$\mathcal{D}_{\gamma_{\text{pi}}} = \{z \in \mathcal{X} : V(z) < 0.56\}.$$

In Fig. 5.4 we numerically verify that the certified RPI set is indeed correct by simulating 100 trajectories in  $\mathcal{X}$  with the disturbance  $w(t) = 0.2 \sin(5t)$  and observing that the trajectories that enter  $\mathcal{D}_{\gamma_{\text{pi}}}$  do not leave it.

The numerical results presented so far only consider regulation to the equilibrium point. However, our approach also handles the estimation of the IROA around any reference trajectory. We obtain a swing-up trajectory of the torque-limited pendulum using the ALTRO solver provided

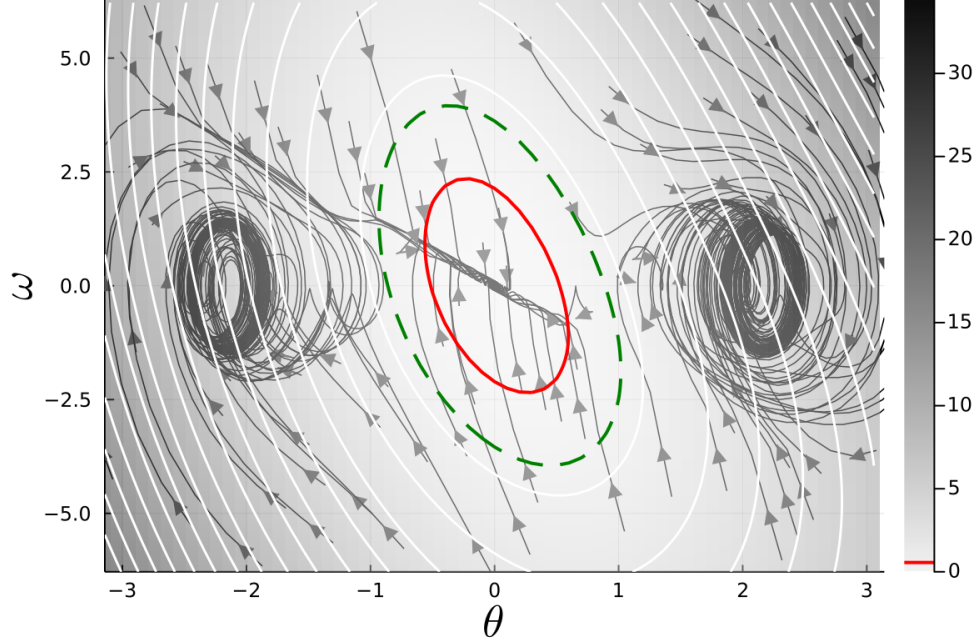


Figure 5.4: All trajectories that start or enter the RPI set determined by the red boundary remain in the region and converge to the equilibrium asymptotically. The green dashed boundary indicates the ROA obtained without considering the disturbances. The black lines denote the trajectories of the system that are randomly initialized in  $\mathcal{X}$ .

by [40] as shown in Fig. 5.5. The swing-up is a necessary maneuver since the limited torque in the system does not allow the pendulum to become inverted in one motion. The energy is slowly added to the system by swinging back and forth until there is enough momentum to balance upright. Since the trajectory optimization is usually performed over a discretized representation of the trajectory, we validate the learned ILF over individual points along the trajectory to obtain the following IROA set:

$$\mathcal{D}_{\gamma_{\text{roa}}} = \{(z, z^*) \in \mathcal{X} \times \mathcal{X} : V(z, z^*) < 0.30\}.$$

In Fig. 5.5, we show a projection of the above set on the specified swing-up trajectory. We numerically verify that the certified IROA is indeed correct by simulating 100 trajectories that start within the IROA around the desired trajectory and observe that the trajectories asymptotically converge to the planned swing-up trajectory.

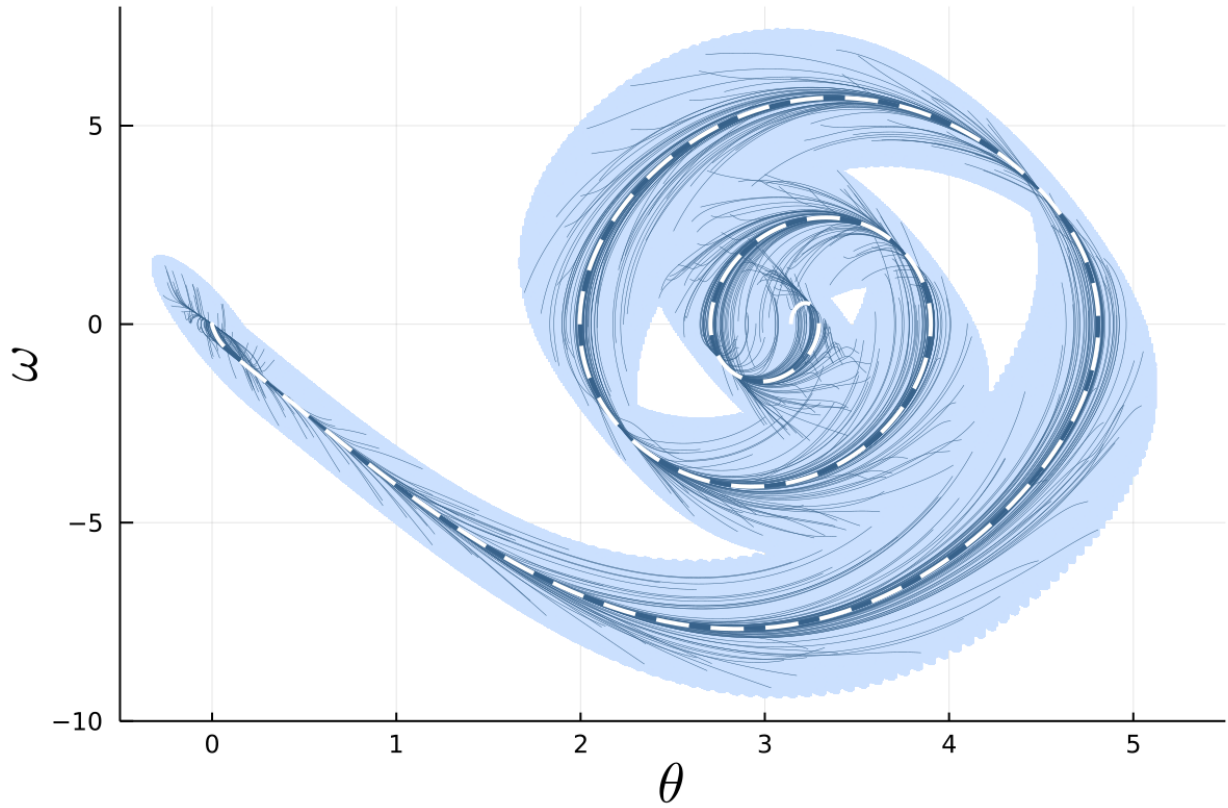


Figure 5.5: The dashed white line indicates the desired swing-up trajectory from the stable position  $[\pi, 0]$  to the inverted position  $[0, 0]$ . The shaded blue areas denote the projection of the IROA on to individual points along the desired trajectory. The thin blue lines denote the trajectories of the system that are randomly initialized in the IROA.

# Chapter 6

## Conclusion

In this dissertation we presented a framework for safe planning and control of nonlinear autonomous systems. We provide several algorithms and control design methods that ensure safe operation of a system by accounting for the uncertainties caused by modeling errors and external disturbances. The framework is composed of four major components. First, an incremental certificate of stability and the associated control policy is established for the nominal system. This ensures that the system is guaranteed to track the desired trajectory if there are no uncertainties affecting the system. Second, a robust adaptive control augmentation is designed that ensures guaranteed transient tracking performance in the presence of bounded uncertainties. In particular, if the uncertainties are matched with the control channel, the framework is capable of tracking with arbitrary performance limited only by the computer and actuator hardware of the system. Third, the modeling uncertainties in the system are learned by collecting data during the system's operation to gradually improve its performance, while maintaining the safety guarantees. Finally, the apriori tracking performance bounds are used to plan collision-free paths in the state space; even if the system is perturbed, the vehicle still remains in the safe regions and away from obstacles and other robots in the environment.

In Chapter 2, we presented algorithms for proximity queries evaluating minimum distance, tolerance verification and collision detection. To our knowledge, this is the first proximity query method that works on such a large class of parametric curves. In addition, the method shows improved computational speed, even compared to the methods taking advantage of a known curve basis. The presented proximity query algorithms are built around a branch-and-bound algorithm that provides  $\epsilon$ -suboptimal solutions. Using this approach, computational results are shown for the evaluation of minimum distance, tolerance verification and collision detection between parametric curves and obstacles.

In Chapter 3, we presented a control methodology to enable safe and guaranteed feedback motion planning based on contraction theory and  $\mathcal{L}_1$ -adaptive control, that we refer to as  $\mathcal{CL}_1$  control architecture. The proposed controller enables the apriori computation of uniform and ultimate bounds which act as safety-certificates. These safety certificates induce ‘tubes’ which can be taken into account by any planner of choice to enable collision checking that accounts for the tracking

performance. In this way, the safety of the system is always guaranteed in the presence of model and environmental uncertainties. Furthermore, by using the control law’s filter bandwidth and rate of adaptation as tuning knobs, the width of the safety tubes can be adjusted if the uncertainties are matched with the control channel.

In Chapter 4, we presented a framework that enables safe simultaneous learning and control. The safety of the method is certified by the tracking error bounds produced by the  $\mathcal{CL}_1$  controller. Data is collected and incorporated into a Gaussian process (GP) model of the uncertainty. The learned GP model is used to generate high probability uniform error bounds on the remainder uncertainty, which are incorporated into the controller to improve the tracking performance. The updated tracking error bounds are taken into account by the planner to plan more optimal but still collision-free paths.

In Chapter 5, we presented an approach for jointly synthesizing certificates and control policies that guarantee incremental asymptotic stability (IAS) (or incremental exponential stability (IES)) to any realizable desired trajectory. We formulated hypothesis spaces for candidate incremental Lyapunov functions (ILFs) and feedback laws and provided deep neural network (DNN) parameterizations for the functions. Using the candidate ILFs and feedback laws, we were able to extract an incremental region of attraction which ensures that trajectories that start in this set remain in this set and converge to each other asymptotically (or exponentially). We showed similar results when jointly searching for a control contraction metric (CCM) and a feedback law.

## 6.1 Future Work

While this work attempts to resolve many problems in the way of guaranteeing safety for nonlinear systems, there are still vast number of problems that have to be solved to truly provide certificates of safety for real world systems. Here we list some directions of research that extend the work proposed here:

- Collision checking methods provided in this dissertation are primarily geared towards trajectories of autonomous mobile robots. However, there are a large of class of systems like manipulators and legged robots, where not just the end-effector but the entire body must remain collision free when operating. For example, in the case of a manipulator, the trajectories must be specified for the position and orientation of each link and checked for collision. This extension will make the work proposed in Chapter 2 more applicable to general robotic systems.
- The  $\mathcal{CL}_1$  framework enables arbitrary tracking performance by compensating for the uncertainties affecting the system. However, it only compensates for uncertainties that are

matched with the control channel, whereas other disturbances are attenuated based on the design of the feedback policy in Chapter 5. However, in most autonomous systems, we are interested in obtaining improved performance on states outside of the control channel (e.g. the position states of a quadrotor are not matched with its control inputs). Extending this framework to handle such uncertainties will enable more useful tracking performance guarantees.

- The learning proposed in Chapter 4 relies on GP regression. However, since most learning and control methods rely on DNNs because of their scalability and ease of use, extending this work to provide guarantees for DNN-learned models will be immensely useful. One possible approach of finding uniform bounds on the learning error can be handled through probably approximately correct Bayes guarantees [171].
- Interval analysis methods suffer from scalability issues which may restrict where the approach presented in Chapter 5 is applicable. Combining these methods with optimization-based verification methods such as Sherlock [172] may alleviate some of these problems. Further investigation of conditions under which a non-empty incremental region of attraction (IROA) is guaranteed to exist would benefit the learning and verification procedure.
- In general the frameworks considered in Chapters 3 to 5 are restricted to state-feedback systems. While this is a necessary first step towards obtaining safety guarantees, most real systems do not have access to noise-free measurements of the state. Providing similar guarantees in case of output-feedback will be crucial in bringing the safe planning and control framework presented in this dissertation to implementations on real systems.



# Appendix A

## Proofs of Chapter 2

*Proof of Lemma 2.1.* Consider the arc length function given in (2.5). Expanding the Euclidean norm and scaling both sides yields

$$\frac{1}{|\mathcal{Q}|} s_\psi(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \int_{\mathcal{Q}} \sqrt{\psi'(t)^\top \psi'(t)} dt.$$

Note that  $\psi'(t)^\top \psi'(t)$  is a strictly positive real-valued function and that the square root function is concave on the interval  $[0, \infty)$ . Using Jensen's inequality, we have

$$\frac{1}{|\mathcal{Q}|} \int_{\mathcal{Q}} \sqrt{\psi'(t)^\top \psi'(t)} dt \leq \sqrt{\frac{1}{|\mathcal{Q}|} \int_{\mathcal{Q}} \psi'(t)^\top \psi'(t) dt}.$$

Multiplying by  $|\mathcal{Q}|$  on both sides gives the result. □

*Proof of Theorem 2.1.* It is obvious that the shortest distance between two points in Euclidean space is the length of the chord joining them. Therefore, for all  $t \in [\alpha, \beta]$  the following must hold

$$\|\psi(\alpha) - \psi(t)\| + \|\psi(t) - \psi(\beta)\| \leq s_\psi([\alpha, t]) + s_\psi([t, \beta]).$$

Then, from the definition of the arc length in (2.5) we have that  $s_\psi([\alpha, t]) + s_\psi([t, \beta])$  is the sum of integrals on adjacent intervals. Since  $\psi$  is rectifiable, the integrals are combined as the total arc length on  $\mathcal{Q}$ , implying that

$$\|\psi(\alpha) - \psi(t)\| + \|\psi(t) - \psi(\beta)\| \leq s_\psi(\mathcal{Q}).$$

Recall that since  $s_\psi(\mathcal{Q}) \leq u_\psi(\mathcal{Q})$  from Lemma 2.1, then every point on the curve evaluated over the interval  $\mathcal{Q}$  is an element of the set  $\mathcal{U}_\mathcal{Q}$ , i.e.

$$\psi(t) \in \mathcal{U}_\mathcal{Q}.$$

From the definition of the curve in (2.1) it follows that  $\Psi_\mathcal{Q} \subset \mathcal{U}_\mathcal{Q}$ . □

*Proof of Theorem 2.2.* It is clear that for all  $t' \in \mathcal{Q} \subseteq \mathcal{I}$

$$\min_{t \in \mathcal{Q}, b \in \mathcal{B}} \|\psi(t) - b\| \leq \min_{b \in \mathcal{B}} \|\psi(t') - b\|,$$

which proves the result that  $d_{\min}(\Psi_{\mathcal{Q}}, \mathcal{B}) \leq d_{\text{ub}}(\Psi_{\mathcal{Q}}, \mathcal{B})$ . Recall from Theorem 2.1 that  $\Psi_{\mathcal{Q}} \subset \mathcal{U}_{\mathcal{Q}}$ , which implies

$$\min_{a \in \mathcal{U}_{\mathcal{Q}}, b \in \mathcal{B}} \|a - b\| \leq \min_{a \in \Psi_{\mathcal{Q}}, b \in \mathcal{B}} \|a - b\|.$$

Thus,  $d_{\text{lb}}(\Psi_{\mathcal{Q}}, \mathcal{B}) \leq d_{\min}(\Psi_{\mathcal{Q}}, \mathcal{B})$ , and we have the result.  $\square$

*Proof of Theorem 2.3.* For any  $\mathcal{Q} \subseteq \mathcal{I}$  recall that

$$d_{\text{ub}}(\Psi_{\mathcal{Q}}, \mathcal{B}) - d_{\text{lb}}(\Psi_{\mathcal{Q}}, \mathcal{B}) = \min_{b \in \mathcal{B}} \|x'(\Psi_{\mathcal{Q}}) - b\| - \|x^* - b^*\|,$$

such that  $x^* \in \mathcal{U}_{\mathcal{Q}}$  and  $b^* \in \mathcal{B}$  minimize (2.9). Since  $\min_{b \in \mathcal{B}} \|x'(\Psi_{\mathcal{Q}}) - b\|$  will be bounded from above by any other element in the set  $\mathcal{B}$ , we have the following inequality:

$$d_{\text{ub}}(\Psi_{\mathcal{Q}}, \mathcal{B}) - d_{\text{lb}}(\Psi_{\mathcal{Q}}, \mathcal{B}) \leq \|x'(\Psi_{\mathcal{Q}}) - b^*\| - \|x^* - b^*\|.$$

From the reverse triangle inequality we have

$$d_{\text{ub}}(\Psi_{\mathcal{Q}}, \mathcal{B}) - d_{\text{lb}}(\Psi_{\mathcal{Q}}, \mathcal{B}) \leq \|x'(\Psi_{\mathcal{Q}}) - x^*\|.$$

Then from Theorem 2.1, we know that the largest variation of elements in  $\mathcal{U}_{\mathcal{Q}}$  is bounded by the upper bound of the arc length,  $u_{\psi}(\mathcal{Q})$ . Since  $x'(\Psi_{\mathcal{Q}}) \in \Psi_{\mathcal{Q}} \subset \mathcal{U}_{\mathcal{Q}}$  and  $x^* \in \mathcal{U}_{\mathcal{Q}}$ , we get

$$d_{\text{ub}}(\Psi_{\mathcal{Q}}, \mathcal{B}) - d_{\text{lb}}(\Psi_{\mathcal{Q}}, \mathcal{B}) \leq u_{\psi}(\mathcal{Q}).$$

From (2.6), we expand  $u_{\psi}(\mathcal{Q})$  with the expression

$$d_{\text{ub}}(\Psi_{\mathcal{Q}}, \mathcal{B}) - d_{\text{lb}}(\Psi_{\mathcal{Q}}, \mathcal{B}) \leq \sqrt{|\mathcal{Q}| \int_{\mathcal{Q}} \psi'(t)^{\top} \psi'(t) dt}.$$

Since  $\psi$  is real valued, the inner product of  $\psi'$  is strictly positive. Thus, we further upper bound our expression by expanding the limits of integration with  $\mathcal{I} \supseteq \mathcal{Q}$ . Define  $0 < k = \int_{\mathcal{I}} \psi'(t)^{\top} \psi'(t) dt$ . Then we have

$$d_{\text{ub}}(\Psi_{\mathcal{Q}}, \mathcal{B}) - d_{\text{lb}}(\Psi_{\mathcal{Q}}, \mathcal{B}) \leq \sqrt{|\mathcal{Q}|k}.$$

Thus, for every  $\epsilon > 0$ , choose  $\delta = \frac{\epsilon^2}{k}$  so that for all  $\mathcal{Q} \subseteq \mathcal{I}$  such that  $|\mathcal{Q}| < \delta$  we get

$$d_{\text{ub}}(\Psi_{\mathcal{Q}}, \mathcal{B}) - d_{\text{lb}}(\Psi_{\mathcal{Q}}, \mathcal{B}) < \epsilon.$$

□

# Appendix B

## Proofs of Chapter 3

### B.1 Technical Results

**Lemma B.1.** *Consider a scalar system with vector inputs*

$$\begin{aligned}\dot{z}(t) &= -az(t) + b(t)^\top \xi(t), \quad z(0) = 0, \\ \xi(s) &= (\mathbb{I}_m - C(s))\sigma(s),\end{aligned}$$

where  $z(t) \in \mathbb{R}$  is the state,  $\sigma(t) \in \mathbb{R}^m$  is the control input with a column vector of transfer functions  $\sigma(s)$ ,  $a > 0$  is a scalar,  $b(t) \in \mathbb{R}^m$  is differentiable, and  $C(s)$  is a low-pass filter of the form  $\frac{\omega}{s+\omega}\mathbb{I}_m$  with bandwidth  $\omega > 0$ . If the following bounds hold

$$\|b\|_{\mathcal{L}_\infty}^{[0,\tau]} \leq \Delta_b, \quad \|\dot{\sigma}\|_{\mathcal{L}_\infty}^{[0,\tau]} \leq \Delta_{\sigma_t},$$

for some  $\tau > 0$ , then the following inequality holds

$$\|z\|_{\mathcal{L}_\infty}^{[0,\tau]} \leq \Delta_b \left( \frac{\|\sigma(0)\|}{|a - \omega|} + \frac{\Delta_{\sigma_t}}{a\omega} \right).$$

*Proof.* The scalar system can be equivalently written as

$$\dot{z}(t) = -az(t) + b(t)^\top \xi(t) \quad z(0) = 0 \quad (\text{B.1})$$

$$\dot{\xi}(t) = -\omega\xi(t) + \dot{\sigma}(t), \quad \xi(0) = \sigma(0). \quad (\text{B.2})$$

The solution to the differential equation in (B.2) is

$$\xi(\lambda) = e^{-\omega\lambda}\sigma(0) + \int_0^\lambda e^{-\omega(\lambda-\nu)}\dot{\sigma}(\nu) \, d\nu,$$

and the solution to (B.1) is

$$z(t) = \int_0^t e^{-a(t-\lambda)}b(\lambda)^\top \xi(\lambda) \, d\lambda.$$

Combining the equations above, we have

$$\|z(t)\| \leq \Delta_b \frac{\|\sigma(0)\|(e^{-\omega t} - e^{-at})}{a - \omega} + \Delta_b \frac{\Delta_{\sigma_t}}{\omega} \left( \frac{1 - e^{-at}}{a} - \frac{e^{-\omega t} - e^{-at}}{a - \omega} \right), \quad \forall t \in [0, \tau].$$

This expression can be further bounded as

$$\|z(t)\| \leq \Delta_b \left( \frac{\|\sigma(0)\|}{|a - \omega|} + \frac{\Delta_{\sigma_t}}{a\omega} \right),$$

for all  $t \in [0, \tau]$ . □

**Lemma B.2.** *Consider a scalar system with vector inputs*

$$\begin{aligned} \dot{z}(t) &= -az(t) + b(t)^\top \xi(t), \quad z(0) = 0, \\ \xi(s) &= (\mathbb{I}_m - C(s))\sigma(s), \end{aligned}$$

where  $z(t) \in \mathbb{R}$  is the state,  $\sigma(t) \in \mathbb{R}^m$  is the control input with a column vector of transfer functions  $\sigma(s)$ ,  $a > 0$  is a scalar,  $b(t) \in \mathbb{R}^m$  is differentiable, and  $C(s)$  is a low-pass filter of the form  $\frac{\omega}{s+\omega} \mathbb{I}_m$  with bandwidth  $\omega > 0$ . If the following bounds hold

$$\|b\|_{\mathcal{L}_\infty}^{[0,\tau]} \leq \Delta_b, \quad \|\dot{b}\|_{\mathcal{L}_\infty}^{[0,\tau]} \leq \Delta_{b_t}, \quad \|\sigma\|_{\mathcal{L}_\infty}^{[0,\tau]} \leq \Delta_\sigma,$$

for some  $\tau > 0$ , then the following inequality holds

$$\|z\|_{\mathcal{L}_\infty}^{[0,\tau]} \leq \Delta_\sigma \frac{2a\Delta_b + \Delta_{b_t}}{a\omega}.$$

*Proof.* We closely follow the analysis in [83, Lemma 1], but show that the inequality holds for vector inputs as well. Since the input is bounded in truncation, the norm of the system solution is bounded as

$$\|z\|_{\mathcal{L}_\infty}^{[0,\tau]} \leq \|\mathcal{Y}\|_{\mathcal{L}_1}^{[0,\tau]} \|\sigma\|_{\mathcal{L}_\infty}^{[0,\tau]},$$

where  $\mathcal{Y}$  is the equivalent of a transfer function between  $\sigma$  and  $z$  [15, Lemma A.7.1]. The impulse response  $y(t)$  of  $\mathcal{Y}$  to the Dirac delta function  $\delta(t)$  is characterized as

$$\begin{aligned} y(t) &= \int_0^t e^{-a(t-\lambda)} b(\lambda)^\top \int_0^\lambda \mathbb{1}_m (\delta(\nu) - \omega e^{-\omega\nu}) \delta(\lambda - \nu) d\nu d\lambda \\ &= \int_0^t e^{-a(t-\lambda)} b(\lambda)^\top \mathbb{1}_m (\delta(\lambda) - \omega e^{-\omega\lambda}) d\lambda \\ &= e^{-at} b(0)^\top \mathbb{1}_m - \int_0^t e^{-a(t-\lambda)} b(\lambda)^\top \mathbb{1}_m \omega e^{-\omega\lambda} d\lambda. \end{aligned}$$

Integrating the second term by parts we obtain the solution of the system as

$$y(t) = e^{-\omega t} b(t)^\top \mathbb{1}_m - \int_0^t \left( e^{-a(t-\lambda)} \dot{b}(\lambda) - a e^{-a(t-\lambda)} b(\lambda) \right)^\top \mathbb{1}_m e^{-\omega \lambda} d\lambda.$$

The  $\mathcal{L}_1$  norm of  $\mathcal{Y}$  is simply the norm of the impulse response, which is given by

$$\begin{aligned} \mathcal{Y}_{\mathcal{L}_1}^{[0,\tau]} &= \|y\|_{\mathcal{L}_1}^{[0,\tau]} = \int_0^\tau \|y(t)\| dt \\ &= \int_0^\tau \left\| e^{-\omega t} b(t)^\top \mathbb{1}_m - \int_0^t \left( e^{-a(t-\lambda)} \dot{b}(\lambda) - a e^{-a(t-\lambda)} b(\lambda) \right)^\top \mathbb{1}_m e^{-\omega \lambda} d\lambda \right\| dt \\ &\leq \Delta_b \frac{1 - e^{-\omega \tau}}{\omega} + \int_0^\tau \left\| \int_0^t \left( e^{-a(t-\lambda)} \dot{b}(\lambda) - a e^{-a(t-\lambda)} b(\lambda) \right)^\top \mathbb{1}_m e^{-\omega \lambda} d\lambda \right\| dt \\ &\leq \Delta_b \frac{1 - e^{-\omega \tau}}{\omega} + (\Delta_{b_t} + a \Delta_b) \int_0^\tau \int_0^t e^{-a(t-\lambda)} e^{-\omega \lambda} d\lambda dt \\ &\leq \Delta_b \frac{1 - e^{-\omega \tau}}{\omega} + (\Delta_{b_t} + a \Delta_b) \left( \frac{1}{a\omega} - \frac{a e^{-\omega \tau} - \omega e^{-a\tau}}{a\omega(a - \omega)} \right). \end{aligned}$$

This expression can be further bounded as

$$\mathcal{Y}_{\mathcal{L}_1}^{[0,\tau]} \leq \frac{2a\Delta_b + \Delta_{b_t}}{a\omega}.$$

Therefore, the solution of the system  $z(t)$  is bounded as

$$z(t) \leq \Delta_\sigma \frac{2a\Delta_b + \Delta_{b_t}}{a\omega},$$

for all  $t \in [0, \tau]$ . □

**Lemma B.3.** Consider the Riemannian manifold  $(\mathcal{X}, M)$ , where the metric satisfies Assumption 3.6, i.e the following uniform bounds hold

$$\underline{\alpha} \mathbb{I}_n \preceq M(x) \preceq \bar{\alpha} \mathbb{I}_n, \quad \forall x \in \mathcal{X},$$

where  $\bar{\alpha} \geq \underline{\alpha} > 0$ . Then the Riemannian energy satisfies the following uniform bounds

$$\underline{\alpha} \|p - q\|^2 \leq \mathcal{E}(p, q) \leq \bar{\alpha} \|p - q\|^2, \quad \forall p, q \in \mathcal{X}.$$

*Proof.* Let  $\Xi(p, q)$  be the set of smooth curves connecting  $p, q \in \mathcal{X}$ . Applying the Cauchy-Schwarz

inequality to the definition of the length of curve given in (3.9) produces

$$l(\gamma)^2 \leq \int_0^1 \gamma_s(s) M(\gamma(s)) \gamma_s(s) ds \triangleq E(\gamma), \quad \forall \gamma \in \Xi(p, q). \quad (\text{B.3})$$

Furthermore, since the minimizing geodesic  $\bar{\gamma} \in \Xi(p, q)$  has the property that  $\sqrt{\gamma_s(s) M(\gamma(s)) \gamma_s(s)}$  is a constant for all  $s \in [0, 1]$  [173, Lemma 5.5], the application of Cauchy-Schwarz lemma also produces

$$l(\bar{\gamma})^2 = \int_0^1 \bar{\gamma}_s(s) M(\bar{\gamma}(s)) \bar{\gamma}_s(s) ds = E(\bar{\gamma}). \quad (\text{B.4})$$

Since, by definition  $\bar{\gamma}$  is a minimizer of  $l(\gamma)$  and  $l(\gamma) \geq 0$  always, it is also a minimizer of  $l^2(\gamma)$ . Using Eqs. (B.3)-(B.4), we conclude that

$$\inf_{\gamma \in \Xi(p, q)} l(\gamma)^2 = l(\bar{\gamma})^2 = E(\bar{\gamma}) = \inf_{\gamma \in \Xi(p, q)} E(\gamma).$$

By the definition of the Riemannian energy given in (3.11), the following chain of equalities are satisfied

$$\mathcal{E}(p, q) = d(p, q)^2 = \left( \inf_{\gamma \in \Xi(p, q)} l(\gamma) \right)^2 = \inf_{\gamma \in \Xi(p, q)} l(\gamma)^2.$$

Using the previous two equalities, we conclude that

$$\mathcal{E}(p, q) = \inf_{\gamma \in \Xi(p, q)} E(\gamma). \quad (\text{B.5})$$

Using the bounds on  $M(x)$ , we further obtain

$$\underline{\alpha} \inf_{\gamma \in \Xi(p, q)} \int_0^1 \gamma_s(s)^\top \gamma_s(s) ds \leq \mathcal{E}(p, q) \leq \bar{\alpha} \inf_{\gamma \in \Xi(p, q)} \int_0^1 \gamma_s(s)^\top \gamma_s(s) ds. \quad (\text{B.6})$$

The expression  $\inf_{\gamma \in \Xi(p, q)} \int_0^1 \gamma_s(s)^\top \gamma_s(s) ds$  represents the Riemannian energy over a manifold with the flat (i.e. constant) metric  $\mathbb{I}_n$ . Under the flat metrics, geodesics are straight lines, i.e.,  $\gamma(s) = (1-s)p + sq$ ,  $s \in [0, 1]$ . Thus,

$$\inf_{\gamma \in \Xi(p, q)} \int_0^1 \gamma_s(s)^\top \gamma_s(s) ds = \|p - q\|^2.$$

Substituting into (B.6) then completes the result.  $\square$

**Lemma B.4.** *Consider a minimizing geodesic  $\bar{\gamma} : [0, 1] \rightarrow \mathcal{X}$  under the metric  $M(x)$ ,  $x \in \mathcal{X}$ , between points  $p, q \in \mathcal{X}$  such that  $\bar{\gamma}(0) = p$  and  $\bar{\gamma}(1) = q$ . If the metric  $M(x)$  satisfies Assump-*

tion 3.6, then the following inequalities are satisfied

$$\|M(\bar{\gamma}(s))\bar{\gamma}_s(s)\| \leq \bar{\alpha}\|p - q\|, \quad (\text{B.7})$$

$$\|\bar{\gamma}_s(s)\| \leq \sqrt{\frac{\bar{\alpha}}{\underline{\alpha}}}\|p - q\|, \quad (\text{B.8})$$

for all  $s \in [0, 1]$ , where  $\Theta(x)^\top \Theta(x) = M(x)$ .

*Proof.* From the proof of Lemma B.3, the Riemannian distance  $d(p, q)$  is bounded as

$$\sqrt{\underline{\alpha}}\|p - q\| \leq d(p, q) \leq \sqrt{\bar{\alpha}}\|p - q\|,$$

since the metric  $M(x)$  satisfies Assumption 3.6. Geodesics exhibit the special property [173, Lemma 5.5] that  $\bar{\gamma}_s(s)^\top M(\bar{\gamma}(s))\bar{\gamma}_s(s) = c > 0$  for all  $s \in [0, 1]$ , i.e. they are of constant speed. Since  $s \in [0, 1]$ , using the proof of Lemma B.3, it can be shown that the speed of a geodesic is also the length of the geodesic

$$\sqrt{\bar{\gamma}_s(s)^\top M(\bar{\gamma}(s))\bar{\gamma}_s(s)} = d(p, q) \leq \sqrt{\bar{\alpha}}\|p - q\|, \quad (\text{B.9})$$

for all  $s \in [0, 1]$ . Therefore given the factorization  $M(x) = \Theta(x)^\top \Theta(x)$ , we obtain the following inequality

$$\|\Theta(\bar{\gamma}(s))\bar{\gamma}_s(s)\| \leq \sqrt{\bar{\alpha}}\|p - q\|.$$

Since  $\|M(\bar{\gamma}(s))\bar{\gamma}_s(s)\| \leq \|\Theta(\bar{\gamma}(s))^\top\| \|\Theta(\bar{\gamma}(s))\bar{\gamma}_s(s)\|$  (because of the submultiplicative property of induced matrix norms and  $\|\Theta(x)\| \leq \sqrt{\bar{\alpha}}$  for all  $x \in \mathcal{X}$ ), we arrive at (B.7). Additionally, since  $M(x)$  is uniformly bounded, the following inequality holds

$$\underline{\alpha}\bar{\gamma}_s(s)^\top \mathbb{I}_n \bar{\gamma}_s(s) \leq \bar{\gamma}_s(s)^\top M(\bar{\gamma}(s))\bar{\gamma}_s(s),$$

for all  $s \in [0, 1]$ . Combining the above inequality with (B.9), we obtain the result in (B.8).  $\square$

**Lemma B.5.** Consider two smooth trajectories  $x_0$  and  $x_1$  such that  $x_0(t), x_1(t) \in \Omega(\rho, x^*(t))$ , for all  $t \in [0, \tau]$ , for some  $\tau > 0$ , and a minimizing geodesic  $\bar{\gamma}(\cdot, t) : [0, 1] \rightarrow \mathcal{X}$  under the metric  $M(x)$  such that  $\bar{\gamma}(0, t) = x_0(t)$  and  $\bar{\gamma}(1, t) = x_1(t)$ . Then, if  $M(x)$  satisfies Assumption 3.6, we have

$$\|\bar{\gamma}_s(1, t)^\top M(\bar{\gamma}(1, t)) - \bar{\gamma}_s(0, t)^\top M(\bar{\gamma}(0, t))\| \leq \frac{\bar{\alpha}}{2\underline{\alpha}} \Delta_{M_x} \|x_0(t) - x_1(t)\|^2, \quad \forall t \in [0, \tau],$$

where  $\Delta_{M_x}$  is defined in (3.14).



*Proof.* Recall that the *differential Lyapunov function* is defined as  $V(x, \delta x) := \delta x^\top M(x) \delta x$ . Then, along the geodesic  $\bar{\gamma}$ , the differential Lyapunov function can be written as

$$V(\bar{\gamma}(s, t), \bar{\gamma}_s(s, t)) = \bar{\gamma}_s(s, t)^\top M(\bar{\gamma}(s, t)) \bar{\gamma}_s(s, t), \quad s \in [0, 1].$$

Thus, we obtain the expressions

$$\begin{aligned} \frac{\partial V}{\partial x}(x, \delta x) &= \left[ \delta x^\top \frac{\partial M}{\partial x_1}(x) \delta x \quad \cdots \quad \delta x^\top \frac{\partial M}{\partial x_n}(x) \delta x \right] := \delta x^\top \frac{\partial M}{\partial x}(x) \delta x \in \mathbb{R}^{1 \times n}, \\ \frac{\partial V}{\partial \delta x}(x, \delta x) &= 2\delta x^\top M(x) \in \mathbb{R}^{1 \times n}, \end{aligned}$$

which evaluated at the geodesic  $\bar{\gamma}$  are given by

$$\begin{aligned} \frac{\partial V}{\partial x}(\bar{\gamma}(s, t), \bar{\gamma}_s(s, t)) &= \left[ \bar{\gamma}_s(s, t)^\top \frac{\partial M}{\partial x_1}(\bar{\gamma}(s, t)) \bar{\gamma}_s(s, t) \quad \cdots \quad \bar{\gamma}_s(s, t)^\top \frac{\partial M}{\partial x_n}(\bar{\gamma}(s, t)) \bar{\gamma}_s(s, t) \right] \\ &= \bar{\gamma}_s(s, t)^\top \frac{\partial M}{\partial x}(\bar{\gamma}(s, t)) \bar{\gamma}_s(s, t), \end{aligned} \quad (\text{B.10})$$

$$\frac{\partial V}{\partial \delta x}(\bar{\gamma}(s, t), \bar{\gamma}_s(s, t)) = 2\bar{\gamma}_s(s, t)^\top M(\bar{\gamma}(s, t)), \quad (\text{B.11})$$

for  $s \in [0, 1]$ .

Now,  $\bar{\gamma}(s, t)$  minimizes the energy functional  $E(\gamma(s, t))$  defined in the proof of Lemma B.3, where  $\gamma(\cdot, t)$  is any smooth curve connecting  $x_0(t), x_1(t) \in \Omega(\rho, x^*(t)) \subset \mathcal{X}$ . In other words,  $\bar{\gamma}(s, t)$  minimizes the functional

$$\int_0^1 \gamma_s(s, t)^\top M(\gamma(s, t)) \gamma_s(s, t) \, ds = \int_0^1 V(\bar{\gamma}(s, t), \bar{\gamma}_s(s, t)) \, ds.$$

Therefore,  $\bar{\gamma}(s, t)$  satisfies the Euler-Lagrange equations [174, Appendix D] given by

$$\frac{d}{ds} \frac{\partial V}{\partial \delta x}(\bar{\gamma}(s, t), \bar{\gamma}_s(s, t)) = \frac{\partial V}{\partial x}(\bar{\gamma}(s, t), \bar{\gamma}_s(s, t)), \quad s \in [0, 1].$$

Using (B.10)-(B.11), we get

$$\frac{d}{ds} \bar{\gamma}_s(s, t)^\top M(\bar{\gamma}(s, t)) = \frac{1}{2} \bar{\gamma}_s(s, t)^\top \frac{\partial M}{\partial x}(\bar{\gamma}(s, t)) \bar{\gamma}_s(s, t), \quad s \in [0, 1].$$

Integrating both sides and applying the fundamental theorem of calculus [175, Section 5.3] produces

$$[\bar{\gamma}_s(s, t)^\top M(\bar{\gamma}(s, t))]_{s=0}^{s=1} = \frac{1}{2} \int_0^1 \bar{\gamma}_s(s, t)^\top \frac{\partial M}{\partial x}(\bar{\gamma}(s, t)) \bar{\gamma}_s(s, t) \, ds.$$

Component-wise, this expression can be written as

$$[\bar{\gamma}_s(s, t)^\top M(\bar{\gamma}(s, t))]_{s=0}^{s=1} [i] = \frac{1}{2} \int_0^1 \bar{\gamma}_s(s, t)^\top \frac{\partial M}{\partial x_i}(\bar{\gamma}(s, t)) \bar{\gamma}_s(s, t) ds, \quad i \in \{1, \dots, n\}. \quad (\text{B.12})$$

From (3.14) and Lemma B.4 the following bounds hold

$$\sum_{i=1}^n \left\| \frac{\partial M}{\partial x_i}(x) \right\| \leq \Delta_{M_x}, \quad \|\bar{\gamma}_s(s, t)\|^2 \leq \frac{\bar{\alpha}}{\underline{\alpha}} \|x_0 - x_1\|^2$$

for all  $x(t) \in \Omega(\rho, x^*(t)) \in \mathcal{X}$ ,  $t \in [0, \tau]$  and  $s \in [0, 1]$ . Since  $\bar{\gamma}(s, t) \in \mathcal{X}$ , for all  $s \in [0, 1]$ , using the aforementioned bounds and (B.12), we arrive at the result.  $\square$

**Lemma B.6.** *Let the state  $x_r(t)$  of the reference system in (3.33) and the state  $x(t)$  of the real system in (3.1) with control input (3.13) satisfy  $x_r(t), x(t) \in \Omega(\rho, x^*(t))$  for all  $t \in [0, \tau]$ , for some  $\tau > 0$ . Additionally, let Assumptions 3.2-3.4 and 3.6 hold. Then, the following inequalities are satisfied*

$$\|\dot{x}_r\|_{\mathcal{L}_\infty}^{[0, \tau]} \leq \Delta_{\dot{x}_r}, \quad (\text{B.13})$$

$$\|\dot{x}\|_{\mathcal{L}_\infty}^{[0, \tau]} \leq \Delta_{\dot{x}}, \quad (\text{B.14})$$

where  $\Delta_{\dot{x}_r}$  and  $\Delta_{\dot{x}}$  are defined in (3.17) and (3.18) respectively.

*Proof.* Using the dynamics of the reference system in (3.33), we obtain

$$\|\dot{x}_r(t)\| \leq \|f(x_r(t))\| + \|B(x_r(t))\| \left( \|u_{c,r}(t)\| + \|\tilde{h}(t, x_r(t))\| \right), \quad \forall t \in [0, \tau],$$

where  $\tilde{h}(t, x_r(t)) = h(t, x_r(t)) - \eta_r(t)$ . Thus, using Assumption 3.3, we obtain

$$\|\dot{x}_r(t)\| \leq \Delta_f + \Delta_B \left( \|u_{c,r}(t)\| + \|\tilde{h}(t, x_r(t))\| \right), \quad \forall t \in [0, \tau]. \quad (\text{B.15})$$

Using the definition of  $u_{c,r}(t)$  in (3.33) and Assumption 3.2, we get the following bound

$$\|u_{c,r}(t)\| \leq \Delta_{u^*} + \|k(x^*(t), x_r(t))\|, \quad \forall t \in [0, \tau].$$

We can use the bound on the feedback term  $\|k(x^*(t), x_r(t))\|$  as in [95, Theorem 5.2 & Eqn. (49)] to get

$$\|u_{c,r}(t)\| \leq \Delta_{u^*} + \rho \Delta_{\delta_u}, \quad \forall t \in [0, \tau]. \quad (\text{B.16})$$

Note that by definition  $\mathcal{L}[\tilde{h}(t, x_r)] = \mathcal{L}[h(t, x_r) - \eta_r(t)] = (\mathbb{I}_m - C(s))\mathcal{L}[h(t, x_r)]$ . Thus, us-

ing [15, Lemma A.7.1], we have

$$\left\| \tilde{h}(t, x_r(t)) \right\| = \|h(t, x_r(t)) - \eta_r(t)\| \leq \|\mathbb{I}_m - C(s)\|_{\mathcal{L}_1} \|h(\cdot, x_r(\cdot))\|_{\mathcal{L}_\infty}^{[0, \tau]}, \quad \forall t \in [0, \tau].$$

Using the fact that  $\|h(t, x_r)\| \leq \Delta_h$  for all  $x_r \in \Omega(\rho, x^*(t))$  and  $t \in [0, \tau]$ , from Assumption 3.4 we obtain

$$\left\| \tilde{h}(t, x_r(t)) \right\| \leq \|\mathbb{I}_m - C(s)\|_{\mathcal{L}_1} \Delta_h, \quad \forall t \in [0, \tau]. \quad (\text{B.17})$$

Then, substituting (B.16)-(B.17) into (B.15) establishes the inequality in (B.13). Using the similar line of reasoning, we obtain the following bound on the state of the real system (3.1) with control input (3.13)

$$\|\dot{x}(t)\| \leq \Delta_f + \Delta_B (\Delta_h + \|u(t)\|), \quad \forall t \in [0, \tau]. \quad (\text{B.18})$$

Using the definition of  $u_a(t)$  in (3.28) and again applying [15, Lemma A.7.1], we get

$$\|u_a(t)\| \leq \|C(s)\|_{\mathcal{L}_1} \|\hat{\sigma}\|_{\mathcal{L}_\infty}^{[0, \tau]}, \quad \forall t \in [0, \tau].$$

We have that  $\|C(s)\|_{\mathcal{L}_1} = 1$  for the chosen first-order low-pass filter. Additionally, since  $\hat{\sigma}(t) \in \mathcal{H}$  due to the adaptation law defined in (3.27), we conclude that  $\|\hat{\sigma}\|_{\mathcal{L}_\infty}^{[0, \tau]} \leq \Delta_h$ . Thus,  $\|u_a(t)\| \leq \Delta_h$  for all  $t \in [0, \tau]$ . And, since  $x(t) \in \Omega(\rho, x^*(t))$  for all  $t \in [0, \tau]$ , the upperbound on  $\|u_c(t)\|$  is equivalent to that of (B.16). Therefore,

$$\|u(t)\| \leq \|u_c(t)\| + \|u_a(t)\| \leq \Delta_u^* + \rho \Delta_{\delta_u} + \Delta_h, \quad \forall t \in [0, \tau]. \quad (\text{B.19})$$

Substituting into (B.18) then establishes the inequality in (B.14).  $\square$

**Lemma B.7.** *Let the state  $x(t)$  of the actual system in (3.1) with control input (3.13) satisfy  $x(t) \in \Omega(\rho, x^*(t))$  for all  $t \in [0, \tau]$ , for some  $\tau > 0$ . Additionally, let Assumptions 3.2-3.5 hold. Then the state prediction error  $\tilde{x}(t)$  defined in (3.26) satisfies*

$$\|\tilde{x}\|_{\mathcal{L}_\infty}^{[0, \tau]} \leq \frac{\Delta_{\tilde{x}}}{\sqrt{\Gamma}}. \quad (\text{B.20})$$

Define  $\tilde{\sigma}(t) := \hat{\sigma}(t) - h(t, x)$  and  $\tilde{\eta}(s) := C(s)\tilde{\sigma}(s)$ . Then the following inequality holds

$$\|\tilde{\eta}\|_{\mathcal{L}_\infty}^{[0, \tau]} \leq \frac{\Delta_{\tilde{\eta}}}{\sqrt{\Gamma}}, \quad (\text{B.21})$$

where  $\Delta_{\tilde{x}}$  and  $\Delta_{\tilde{\eta}}$  are defined in (3.19) and (3.20) respectively.

*Proof.* The state predictor error dynamics are computed using (3.1) and (3.26) as

$$\dot{\tilde{x}}(t) = \dot{\hat{x}}(t) - \dot{x}(t) = A_m \tilde{x}(t) + B(x) \tilde{\sigma}(t), \quad \tilde{x}(0) = 0,$$

where  $\tilde{\sigma}(t) = \hat{\sigma}(t) - h(t, x)$ . Consider the Lyapunov function  $V(\tilde{x}, \tilde{\sigma}) = \tilde{x}(t)^\top P \tilde{x}(t) + \tilde{\sigma}(t)^\top \Gamma^{-1} \tilde{\sigma}(t)$ . Then its time derivative is given by

$$\dot{V}(\tilde{x}, \tilde{\sigma}) = -\tilde{x}(t)^\top Q \tilde{x}(t) + 2\tilde{\sigma}(t)^\top B(x)^\top P \tilde{x}(t) + 2\tilde{\sigma}(t)^\top \Gamma^{-1} (\dot{\hat{\sigma}}(t) - \dot{h}(t, x)),$$

where  $P \succ 0$  and  $Q \succ 0$  define the adaptation law in (3.27). Combing the adaptation law with the equation above, we obtain

$$\dot{V}(\tilde{x}, \tilde{\sigma}) = -\tilde{x}(t)^\top Q \tilde{x}(t) + 2\tilde{\sigma}(t)^\top (B(x)^\top P \tilde{x}(t) + \text{Proj}_{\mathcal{H}}(\dot{\hat{\sigma}}(t), -B(x)^\top P \tilde{x}(t)) - 2\tilde{\sigma}(t)^\top \Gamma^{-1} \dot{h}(t, x)).$$

From [108, Lemma 6], the projection operator ensures that  $\tilde{\sigma}(t)^\top (\text{Proj}_{\mathcal{H}}(\dot{\hat{\sigma}}(t), y) - y) \leq 0$  for all  $y \in \mathbb{R}^n$ . Therefore, the equation above reduces to

$$\dot{V}(\tilde{x}, \tilde{\sigma}) \leq -\tilde{x}(t)^\top Q \tilde{x}(t) - 2\tilde{\sigma}(t)^\top \Gamma^{-1} \dot{h}(t, x).$$

It is easy to show that  $\|\tilde{\sigma}(t)\| \leq 2\Delta_h$ . Since  $\|\dot{x}\|_{\mathcal{L}_\infty}^{[0, \tau]} \leq \Delta_{\dot{x}}$  from (3.18), we have the following bound for the time-derivative of the uncertainty

$$\|\dot{h}(t, x)\| = \left\| \frac{\partial h}{\partial t}(x) + \frac{\partial h}{\partial x}(x) \dot{x} \right\| \leq \Delta_{h_t} + \Delta_{h_x} \Delta_{\dot{x}},$$

for all  $t \in [0, \tau]$ . Combining, these bounds with the equation above results in the following inequality

$$\dot{V}(\tilde{x}, \tilde{\sigma}) \leq -\underline{\lambda}(Q) \|\tilde{x}\|^2 + 4\Delta_h \Gamma^{-1} (\Delta_{h_t} + \Delta_{h_x} \Delta_{\dot{x}}).$$

Now, if  $\dot{V}(\tilde{x}(t), \tilde{\sigma}(t)) \geq 0$ , then

$$\|\tilde{x}(t)\|^2 \leq \frac{4\Delta_h (\Delta_{h_t} + \Delta_{h_x} \Delta_{\dot{x}})}{\Gamma \underline{\lambda}(Q)}, \quad (\text{B.22})$$

for all  $t \in [0, \tau]$ . However, the Lyapunov function always remains bounded as

$$\underline{\lambda}(P) \|\tilde{x}(t)\|^2 \leq V(\tilde{x}(t), \tilde{\sigma}(t)) \leq \bar{\lambda}(P) \|\tilde{x}(t)\|^2 + 4\Delta_h^2 \Gamma^{-1}.$$

Combining the equation above with (B.22), we arrive at the result in (B.21). In order to show that the inequality in (B.21) holds, we start with the error in the uncertainty estimate from the state

predictor error dynamics in terms of the Moore-Penrose inverse of  $B(x)$  as

$$\tilde{\sigma}(t) = B^\dagger(x)\dot{\tilde{x}}(t) - B^\dagger(x)A_m\tilde{x}(t).$$

Since  $\tilde{\eta}(s) = C(s)\tilde{\sigma}(s)$ , we obtain

$$\begin{aligned}\tilde{\eta}(s) &= C(s)\mathcal{L}[B^\dagger(x)\dot{\tilde{x}}(t) - B^\dagger(x)A_m\tilde{x}(t)] \\ &= C(s)\mathcal{L}\left[\frac{d}{dt}(B^\dagger(x)\tilde{x}(t)) - \partial_{\tilde{x}}B^\dagger(x)\tilde{x}(t) - B^\dagger(x)A_m\tilde{x}(t)\right] \\ &= C(s)s\mathcal{L}[B^\dagger(x)\tilde{x}(t)] + C(s)\mathcal{L}[-\partial_{\tilde{x}}B^\dagger(x)\tilde{x}(t) - B^\dagger(x)A_m\tilde{x}(t)].\end{aligned}$$

Since  $\|\dot{x}\|_{\mathcal{L}_\infty}^{[0,\tau]} \leq \Delta_{\dot{x}}$  from (3.18), we have that  $\|\partial_{\tilde{x}}B^\dagger(x)\| = \left\|\sum_{j=0}^n \frac{\partial B^\dagger(x)}{\partial x_j} \dot{x}_j(t)\right\| \leq \Delta_{B_x^\dagger} \Delta_{\dot{x}}$  for all  $t \in [0, \tau]$ . Furthermore, from Assumption 3.5 we have that  $\|B^\dagger(x)\| \leq \Delta_{B^\dagger}$  for all  $x \in \Omega(\rho, x^*(t))$  and  $t \in [0, \tau]$ . Therefore, using the property from [15, Lemma A.7.1], the following inequality holds

$$\|\tilde{\eta}(t)\| \leq \left(\|C(s)s\|_{\mathcal{L}_1} \Delta_{B^\dagger} + \|C(s)\|_{\mathcal{L}_1} \left(\Delta_{B_x^\dagger} \Delta_{\dot{x}} + \Delta_{B^\dagger} \|A_m\|\right)\right) \frac{\Delta_{\tilde{x}}}{\sqrt{\Gamma}},$$

for all  $x \in \Omega(\rho, x^*(t))$  and  $t \in [0, \tau]$ . Since  $\|C(s)\|_{\mathcal{L}_1} = 1$ , we arrive at the result.  $\square$

**Lemma B.8.** *Let the state  $x_r(t)$  of the reference system in (3.33) and the state  $x(t)$  of the real system in (3.1) with control input (3.13) satisfy  $x_r(t), x(t) \in \Omega(\rho, x^*(t))$  for all  $t \in [0, \tau]$ , for some  $\tau > 0$ . Additionally, let Assumptions 3.2-3.4 hold. Then, if  $\bar{\gamma}(\cdot, t) : [0, 1] \rightarrow \mathcal{X}$  is the minimizing geodesic under the metric  $M(x)$  satisfying Assumption 3.6 such that  $\bar{\gamma}(0, t) = x(t)$  and  $\bar{\gamma}(1, t) = x_r(t)$ , the following inequality is satisfied*

$$\left\|\frac{d}{dt}(B(x)^\top M(x)\bar{\gamma}_s(0, t))\right\| \leq \Delta_\Psi \|x_r(t) - x(t)\|, \quad (\text{B.23})$$

for all  $t \in [0, \tau]$ , where  $\Delta_\Psi$  is defined in (3.22).

*Proof.* We apply chain rule and triangle inequality to obtain

$$\begin{aligned}\left\|\frac{d}{dt}(B(x)^\top M(x)\bar{\gamma}_s(0, t))\right\| &\leq \|B(x)^\top M(x)\dot{\bar{\gamma}}_s(0, t)\| + \|B(x)^\top \partial_{\dot{x}(t)} M(x)\bar{\gamma}_s(0, t)\| \\ &\quad + \|\partial_{\dot{x}(t)} B(x)^\top M(x)\bar{\gamma}_s(0, t)\|. \quad (\text{B.24})\end{aligned}$$

The time-evolution of the velocity of the minimizing geodesic evaluated at  $s = 0$  is given by the

differential dynamics [95, Theorem 3.2] of the actual system as follows

$$\dot{\bar{\gamma}}_s(0, t) = \left( \left[ \frac{\partial f(x)}{\partial x} + \sum_{j=1}^m (u_j(t) + h_j(t, x)) \frac{\partial b_j(x)}{\partial x} \right] + B(x) \frac{\partial h(t, x)}{\partial x} \right) \bar{\gamma}_s(0, t) + B(x) \delta_u, \quad (\text{B.25})$$

where  $b_j(x)$  is the  $j^{\text{th}}$  column of  $B(x)$ ,  $u_j(t)$  is the  $j^{\text{th}}$  value in control channel, and  $h_j(t, x)$  is the  $j^{\text{th}}$  value of the uncertainty. Previously, from Assumptions 3.3 and 3.4 we know that

$$\begin{aligned} \left\| \frac{\partial f(x)}{\partial x} \right\| &\leq \Delta_{f_x}, \quad \|B(x)\| \leq \Delta_B, \quad \sum_{i=1}^n \left\| \frac{\partial B(x)}{\partial x_i} \right\| \leq \Delta_{B_x}, \quad \sum_{j=1}^m \left\| \frac{\partial b_j(x)}{\partial x} \right\| \leq \Delta_{b_x}, \\ \|h(t, x)\| &\leq \Delta_h, \quad \left\| \frac{\partial h(t, x)}{\partial x} \right\| \leq \Delta_{h_x}, \end{aligned}$$

for all  $x \in \Omega(\rho, x^*(t))$  and  $t \in [0, \tau]$ . Additionally from (B.8), (B.19), and [95, Theorem 5.2], the following hold

$$\|u(t)\| \leq \Delta_{u^*} + \rho \Delta_{\delta_u} + \Delta_h, \quad \|\bar{\gamma}_s(1, t)\| \leq \sqrt{\frac{\bar{\alpha}}{\underline{\alpha}}} \|x_r(t) - x(t)\|, \quad \|\delta_u\| \leq \Delta_{\delta_u} \|x_r(t) - x(t)\|$$

respectively for all  $x \in \Omega(\rho, x^*(t))$  and  $t \in [0, \tau]$ , where  $\Delta_{\delta_u}$  is defined in (3.16). With these considerations, the expression in (B.25) is bounded by  $\Delta_{\bar{\gamma}_s} \|x_r(t) - x(t)\|$  for all  $t \in [0, \tau]$ . Furthermore, from Assumption 3.6 we know that  $\bar{\alpha} \mathbb{I}_n \succ M(x) \succ \underline{\alpha} \mathbb{I}_n \succ 0$  for all  $x \in \mathcal{X}$ , which implies that

$$\|B(x)^\top M(x) \dot{\bar{\gamma}}_s(0, t)\| \leq \Delta_B \bar{\alpha} \Delta_{\bar{\gamma}_s} \|x_r(t) - x(t)\|, \quad \forall t \in [0, \tau]. \quad (\text{B.26})$$

Since,  $\|\partial_{\dot{x}} M(x)\| \leq \left\| \sum_{i=1}^n \frac{\partial M}{\partial x_i} \dot{x}_i(t) \right\| \leq \Delta_{M_x} \Delta_{\dot{x}}$  from (3.14) and (3.18) for all  $x \in \Omega(\rho, x^*(t))$  and  $t \in [0, \tau]$ , we obtain

$$\|B(x)^\top \partial_{\dot{x}} M(x) \bar{\gamma}_s(0, t)\| \leq \Delta_B \Delta_{M_x} \Delta_{\dot{x}} \sqrt{\frac{\bar{\alpha}}{\underline{\alpha}}} \|x_r(t) - x(t)\|, \quad \forall t \in [0, \tau]. \quad (\text{B.27})$$

Finally,  $\|\partial_{\dot{x}} B(x)\| \leq \left\| \sum_{i=1}^n \frac{\partial B}{\partial x_i} \dot{x}_i(t) \right\| \leq \Delta_{B_x} \Delta_{\dot{x}}$  from Assumption 3.3 and (3.18), and  $\|M(x) \bar{\gamma}_s(0, t)\| \leq \bar{\alpha} \|x_r(t) - x(t)\|$  holds from (B.7). Therefore

$$\|\partial_{\dot{x}} B(x)^\top M(x) \bar{\gamma}_s(0, t)\| \leq \Delta_{B_x} \Delta_{\dot{x}} \bar{\alpha} \|x_r(t) - x(t)\|, \quad \forall t \in [0, \tau]. \quad (\text{B.28})$$

Substituting (B.26) to (B.28) in (B.24), we arrive at the main result in (B.23).  $\square$

## B.2 Main Results

In this appendix we provide the proofs of the claims in Chapter 3.

*Proof of Lemma 3.1.* We show that  $\|x^* - x_r\|_{\mathcal{L}_\infty} < \rho_r$  by contradiction. We have  $\|x^*(0) - x_r(0)\| = \|x_0^* - x_0\| < \rho_r$ , since  $\epsilon > 0$  and  $\frac{\bar{\alpha}}{\underline{\alpha}} \geq 1$  from (3.29). Assume that the lemma statement does not hold; then since  $x_r$  and  $x^*$  are continuous, there must exist a  $\tau^* > 0$  such that

$$\begin{aligned}\|x^*(\tau^*) - x_r(\tau^*)\| &= \rho_r, \\ \|x^*(t) - x_r(t)\| &< \rho_r, \quad t \in [0, \tau^*).\end{aligned}$$

Consider the minimizing geodesic  $\bar{\gamma}(s, t)$  such that  $\bar{\gamma}(0, t) = x^*(t)$  and  $\bar{\gamma}(1, t) = x_r(t)$ . From [95, Theorem 3.2], the time derivative of the Riemannian energy satisfies

$$\begin{aligned}\frac{1}{2}\dot{\mathcal{E}}(x^*, x_r) &= \bar{\gamma}_s(s, t)^\top M(\bar{\gamma}(s, t))\dot{\bar{\gamma}}(s, t)|_{s=0}^{s=1} \\ &= \bar{\gamma}_s(1, t)^\top M(x_r)\dot{x}_r(t) - \bar{\gamma}_s(0, t)^\top M(x^*)\dot{x}^*(t).\end{aligned}$$

Substituting the desired and reference system dynamics from (3.2) and (3.33) respectively into the equation above, we obtain

$$\begin{aligned}\frac{1}{2}\dot{\mathcal{E}}(x^*, x_r) &= \bar{\gamma}_s(1, t)^\top M(x_r) (f(x_r) + B(x_r)(u_{c,r}(t) + h(t, x_r) - \eta_r(t))) \\ &\quad - \bar{\gamma}_s(0, t)^\top M(x^*) (f(x^*) + B(x^*)u^*(t)).\end{aligned}\tag{B.29}$$

As shown in [97, Lemma 1], since the uncertainty is matched with the control channel, the metric  $M$  designed for the unperturbed system also satisfies the stronger CCM conditions for the real system. This implies that the unperturbed part of the reference system is exponentially convergent with the rate  $\lambda$ , given by the following expression

$$\bar{\gamma}_s(1, t)^\top M(x_r) (f(x_r) + B(x_r)u_{c,r}(t)) - \bar{\gamma}_s(0, t)^\top M(x^*) (f(x^*) + B(x^*)u^*(t)) \leq -\lambda\mathcal{E}(x^*, x_r).$$

Substituting into (B.29) produces

$$\frac{1}{2}\dot{\mathcal{E}}(x^*, x_r) \leq -\lambda\mathcal{E}(x^*, x_r) + \bar{\gamma}_s(1, t)^\top M(x_r)B(x_r)(h(t, x_r) - \eta_r(t)).$$

Integrating both sides of the equation above, we have

$$\mathcal{E}(x^*, x_r) \leq e^{-2\lambda t}\mathcal{E}(x_0^*, x_0) + 2 \int_0^t e^{-2\lambda(t-\nu)} \bar{\gamma}_s(1, t)^\top M(x_r)B(x_r)(h(\nu, x_r) - \eta_r(t)) d\nu. \tag{B.30}$$

The integral term on the right-hand side in the equation above can be expressed as the solution to the following virtual scalar system

$$\dot{z}(t) = -2\lambda z(t) + \bar{\gamma}_s(1, t)^\top M(x_r)B(x_r)\xi(t), \quad z(0) = 0, \quad (\text{B.31})$$

$$\xi(s) = (1 - C(s))\mathcal{L}[h(t, x_r)]. \quad (\text{B.32})$$

Note that from Lemmas B.4 and B.6 and Assumptions 3.3 and 3.4 the following bounds hold for all  $x_r(t) \in \Omega(\rho_r, x^*(t)) \subset \Omega(\rho, x^*)$  (since  $\rho_r < \rho$ ) and all  $t \in [0, \tau^*]$ :

$$\begin{aligned} \|\bar{\gamma}_s(1, t)M(x_r)B(x_r)\| &\leq \rho\bar{\alpha}\Delta_B, \quad \|h(t, x_r)\| \leq \Delta_h, \\ \left\|\dot{h}(t, x_r)\right\| &= \left\|\frac{\partial h(t, x_r)}{\partial t} + \frac{\partial h(t, x_r)}{\partial x}\dot{x}_r(t)\right\| \leq \Delta_{h_t} + \Delta_{h_x}\Delta_{\dot{x}_r}. \end{aligned}$$

Then the solution of a linear system of the form in (B.31) and (B.32) satisfies the following norm bound from Lemma B.1

$$\|z(t)\| \leq \rho\bar{\alpha}\Delta_B \left( \frac{\Delta_h}{|2\lambda - \omega|} + \frac{\Delta_{h_t} + \Delta_{h_x}\Delta_{\dot{x}_r}}{2\lambda\omega} \right) = \frac{1}{2}\underline{\alpha}\zeta_1(\omega),$$

where  $\zeta_1$  is defined in (3.31a). Substituting this inequality into (B.30) produces

$$\mathcal{E}(x^*, x_r) \leq e^{-2\lambda t}\mathcal{E}(x_0^*, x_0) + \underline{\alpha}\zeta_1(\omega)$$

Moreover, since the metric  $M$  satisfies Assumption 3.6, the Riemannian energy satisfies the following lower bound from Lemma B.3

$$\underline{\alpha}\|x^*(t) - x_r(t)\|^2 \leq \mathcal{E}(x^*, x_r) \leq e^{-2\lambda t}\mathcal{E}(x_0^*, x_0) + \underline{\alpha}\zeta_1(\omega).$$

In our contradiction argument, we had that  $\|x^*(\tau^*) - x(\tau^*)\| = \rho_r$ . Therefore the following inequality must be satisfied

$$\underline{\alpha}\rho_r^2 \leq e^{-2\lambda\tau^*}\mathcal{E}(x_0^*, x_0) + \underline{\alpha}\zeta_1(\omega) \quad (\text{B.33})$$

$$< \mathcal{E}(x_0^*, x_0) + \underline{\alpha}\zeta_1(\omega), \quad (\text{B.34})$$

for some  $\tau^* > 0$ . However, from (3.32a) the bandwidth is chosen such that

$$\underline{\alpha}\rho_r^2 \geq \mathcal{E}(x_0^*, x_0) + \underline{\alpha}\zeta_1(\omega).$$

This directly contradicts our earlier statement in (B.34). Therefore,  $\|x_r - x^*\|_{\mathcal{L}_\infty} < \rho_r$ . Moreover,



from (B.33) we also obtain the following uniform ultimate bound

$$\|x^*(t) - x_r(t)\| \leq \sqrt{\frac{e^{-2\lambda T} \mathcal{E}(x_0^*, x_0)}{\underline{\alpha}}} + \zeta_1(\omega),$$

for all  $t \geq T \geq 0$ . □

*Proof of Lemma 3.2.* We prove this lemma by contradiction. Assume that

$$\|x_r - x\|_{\mathcal{L}_\infty}^{[0, \tau]} \geq \rho_a, \quad (\text{B.35})$$

for some  $\tau > 0$ , where  $\rho_a$  is an arbitrary positive scalar used in the definition of  $\rho$  in (3.30). Since  $x_r(0) = x(0)$ , there exists a  $\tau^* \in (0, \tau]$  such that

$$\begin{aligned} \|x_r(\tau^*) - x(\tau^*)\| &= \rho_a, \\ \|x_r(t) - x(t)\| &< \rho_a, \quad t \in [0, \tau^*). \end{aligned} \quad (\text{B.36})$$

Let  $\bar{\gamma}(s, t)$  be the minimizing geodesic between  $x_r$  and  $x$  such that  $\bar{\gamma}(1, t) = x_r(t)$  and  $\bar{\gamma}(0, t) = x(t)$ . Consider the Riemannian energy  $\mathcal{E}(x_r, x)$  between  $x_r$  and  $x$ . Then, the time derivative of the Riemannian energy as shown in [95, Theorem 3.2] is given by

$$\begin{aligned} \frac{1}{2} \dot{\mathcal{E}}(x_r, x) &= \bar{\gamma}_s(s, t)^\top M(\bar{\gamma}(s, t)) \dot{\bar{\gamma}}(s, t) \Big|_{s=0}^{s=1} \\ &= \bar{\gamma}_s(1)^\top M(x_r) \dot{x}_r - \bar{\gamma}_s(0)^\top M(x) \dot{x}. \end{aligned}$$

Substituting the reference system dynamics from (3.33) and the actual system dynamics from (3.1) with the control law from (3.13), we get

$$\begin{aligned} \frac{1}{2} \dot{\mathcal{E}}(x_r, x) &= \gamma_s(1)^\top M(x_r) [f(x_r) + B(x_r)(u_{c,r}(t) + h(t, x_r) - \eta_r(t))] \\ &\quad - \gamma_s(0)^\top M(x) [f(x) + B(x)(u_c(t) + h(t, x) - \hat{\eta}(t))], \end{aligned}$$

where  $\hat{\eta}(t) = -u_a(t)$ . Similar to the reasoning used in Lemma 3.1, the metric  $M$  designed for the ideal system also satisfies the stronger CCM conditions for the real system [97, Lemma 1]. This implies that the nominal parts of the system are contracting, given by

$$\frac{1}{2} \dot{\mathcal{E}}(x_r, x) \leq -\lambda \mathcal{E}(x_r, x) + \Psi(x_r)^\top (h(t, x_r) - \eta_r(t)) - \Psi(x)^\top (h(t, x) - \hat{\eta}(t)),$$

where  $\Psi(x_r) := B(x_r)^\top M(x_r) \bar{\gamma}_s(1, t)$  and  $\Psi(x) := B(x)^\top M(x) \bar{\gamma}_s(0, t)$  are introduced for clarity. Define  $\eta(s) = C(s) \mathcal{L}[h(t, x)]$ ; then by adding and subtracting  $\Psi(x)^\top (h(t, x_r) - \eta_r(t) + \eta(t))$

on the right-hand side, we obtain

$$\begin{aligned} \frac{1}{2}\dot{\mathcal{E}}(x_r, x) \leq & -\lambda\mathcal{E}(x_r, x) + (\Psi(x_r) - \Psi(x))^\top (h(t, x_r) - \eta_r(t)) + \Psi(x)^\top (h(t, x_r) - \eta_r(t) - h(t, x) + \eta(t)) \\ & + \Psi(x)^\top (\hat{\eta}(t) - \eta(t)). \end{aligned}$$

Since  $h(t, x_r) - \eta_r(t) = \mathcal{L}^{-1}[(1 - C(s))\mathcal{L}[h(t, x_r)]]$ ,  $h(t, x) - \eta(t) = \mathcal{L}^{-1}[(1 - C(s))\mathcal{L}[h(t, x)]]$ , with  $\tilde{\eta}(t) = \hat{\eta}(t) - \eta(t)$ , we rewrite the equation above as

$$\frac{1}{2}\dot{\mathcal{E}}(x_r, x) \leq -\lambda\mathcal{E}(x_r, x) + \Phi_1(x_r, x) + \Phi_2(x_r, x) + \Phi_3(x_r, x), \quad (\text{B.37})$$

where

$$\begin{aligned} \Phi_1(x_r, x) &:= (\Psi(x_r) - \Psi(x))^\top \mathcal{L}^{-1}[(1 - C(s))\mathcal{L}[h(t, x_r)]], \\ \Phi_2(x_r, x) &:= \Psi(x)^\top \mathcal{L}^{-1}[(1 - C(s))\mathcal{L}[h(t, x_r) - h(t, x)]], \\ \Phi_3(x_r, x) &:= \Psi(x)^\top \tilde{\eta}(t). \end{aligned}$$

Solving the differential equation in (B.37), we obtain

$$\mathcal{E}(x_r, x) \leq \mathcal{E}(x_r(0), x(0)) + 2 \int_0^t e^{-2\lambda(t-\nu)} (\Phi_1(x_r, x) + \Phi_2(x_r, x) + \Phi_3(x_r, x)) d\nu.$$

Since  $x_r(0) = x(0) \implies \mathcal{E}(x_r(0), x(0)) = 0$ , the inequality above reduces to

$$\mathcal{E}(x_r, x) \leq 2 \int_0^t e^{-2\lambda(t-\nu)} (\Phi_1(x_r, x) + \Phi_2(x_r, x) + \Phi_3(x_r, x)) d\nu. \quad (\text{B.38})$$

Notice that  $\|\Psi(x_r) - \Psi(x)\|$  satisfies the following bounds

$$\|\Psi(x_r) - \Psi(x)\| \leq \|B(x_r)^\top M(x_r)\bar{\gamma}_s(1, t) - B(x)M(x)\bar{\gamma}_s(0, t)\|.$$

Adding and subtracting  $B(x)M(x_r)\bar{\gamma}_s(1, t)$ , from the right hand side of the equation above we obtain

$$\begin{aligned} \|\Psi(x_r) - \Psi(x)\| &\leq \|(B(x_r) - B(x))^\top M(x_r)\bar{\gamma}_s(1, t) + B(x)^\top (M(x_r)\bar{\gamma}_s(1, t) - M(x)\bar{\gamma}_s(0, t))\| \\ &\leq \|B(x_r) - B(x)\| \|M(x_r)\bar{\gamma}_s(1, t)\| + \|B(x)\| \|M(x_r)\bar{\gamma}_s(1, t) - M(x)\bar{\gamma}_s(0, t)\|. \end{aligned} \quad (\text{B.39})$$

Since  $x_r(t) \in \Omega(\rho_r, x^*(t))$  from Lemma 3.1 and  $x(t) \in \Omega(\rho, x^*(t))$  for all  $t \in [0, \tau^*]$  by assump-

tion in (B.36), the following bounds hold for  $t \in [0, \tau^*]$  as a result of Assumption 3.3:

$$\|B(x_r) - B(x)\| \leq \Delta_{B_x} \|x_r(t) - x(t)\|, \quad \|B(x)\| \leq \Delta_B.$$

From Lemmas B.4 and B.5 we obtain the following respectively

$$\|M(x_r)\bar{\gamma}_s(1, t)\| \leq \bar{\alpha} \|x_r(t) - x(t)\|, \quad \|M(x_r)\bar{\gamma}_s(1, t) - M(x)\bar{\gamma}_s(0, t)\| \leq \frac{\bar{\alpha}}{2\underline{\alpha}} \Delta_{M_x} \|x_r(t) - x(t)\|^2,$$

where  $\Delta_{M_x}$  is defined in (3.14). Substituting these bounds in (B.39) produces

$$\|\Psi(x_r) - \Psi(x)\| \leq \frac{1}{2} \bar{\alpha} \Delta_{\Psi_x} \|x_r(t) - x(t)\|^2, \quad (\text{B.40})$$

which holds for all  $t \in [0, \tau^*]$ , and where  $\Delta_{\Psi_x}$  is a scalar defined in (3.15). Additionally, from Assumption 3.4 we know that the following inequalities hold

$$\|h(t, x_r)\| \leq \Delta_h, \quad \|h(t, x_r) - h(t, x)\| \leq \Delta_{h_x} \|x_r(t) - x(t)\|. \quad (\text{B.41})$$

Since  $\left\|\frac{\partial h}{\partial t}(t, x_r)\right\| \leq \Delta_{h_t}$  and  $\left\|\frac{\partial h}{\partial x}(t, x_r)\right\| \leq \Delta_{h_x}$  from Assumption 3.4, and  $\|\dot{x}_r\|_{\mathcal{L}_\infty} \leq \Delta_{\dot{x}_r}$  from Lemma B.6, the following inequality is satisfied

$$\begin{aligned} \|\dot{h}(t, x_r)\| &= \left\| \frac{\partial h}{\partial t}(t, x_r) + \frac{\partial h}{\partial x}(t, x_r) \dot{x}_r \right\| \\ &\leq \Delta_{h_t} + \Delta_{h_x} \Delta_{\dot{x}_r}. \end{aligned} \quad (\text{B.42})$$

Since  $\|M(x)\bar{\gamma}(0, t)\| \leq \bar{\alpha} \|x_r(t) - x(t)\|$  from (B.7) for all  $t \in [0, \tau^*]$ , the following holds

$$\begin{aligned} \|\Psi(x)\| &= \|B(x)^\top M(x)\bar{\gamma}(0, t)\| \leq \|B(x)\| \|M(x)\bar{\gamma}(0, t)\| \\ &\leq \Delta_B \bar{\alpha} \|x_r(t) - x(t)\| \end{aligned} \quad (\text{B.43})$$

for all  $t \in [0, \tau^*]$ . From Lemma B.8 we have the following result for all  $t \in [0, \tau^*]$

$$\left\| \dot{\Psi}(x) \right\| \leq \Delta_{\dot{\Psi}} \|x_r(t) - x(t)\|. \quad (\text{B.44})$$

In order to derive bounds on (B.38), define the following scalar trajectories

$$z_1(t) := \int_0^t e^{-2\lambda(t-\nu)} \Phi_1(x, x_r) d\nu, \quad z_2(t) := \int_0^t e^{-2\lambda(t-\nu)} \Phi_2(x, x_r) d\nu.$$

Then, the functions  $z_i, i \in \{1, 2\}$ , are the states of the following systems

$$\dot{z}_i(t) = -2\lambda z_i(t) + b_i(t)\xi_i(t), \quad z_i(0) = 0, \quad (\text{B.45a})$$

$$\xi_i(s) = (1 - C(s))\sigma_i(s), \quad (\text{B.45b})$$

where

$$b_1(t) = \Psi(x_r) - \Psi(x), \quad \sigma_1(t) = h(t, x_r),$$

$$b_2(t) = \Psi(x), \quad \sigma_2(t) = h(t, x_r) - h(t, x).$$

From (B.35) we assumed that  $\|x_r(t) - x(t)\| \leq \rho_a$  for  $t \in [0, \tau^*]$ . Using Lemma B.1 for the  $z_1(t)$  system, Lemma B.2 for the  $z_2(t)$  system, and the bounds in (B.40) to (B.44), we have the following inequalities

$$\|z_1(t)\| \leq \frac{1}{2}\zeta_2(\omega)\rho_a^2, \quad \|z_2(t)\| \leq \frac{1}{2}\zeta_3(\omega)\rho_a^2, \quad (\text{B.46})$$

for all  $t \in [0, \tau^*]$ , and where  $\zeta_2$  and  $\zeta_3$  are defined in (3.31b) and (3.31c) respectively. Moreover, it is easy to show from (B.21) that

$$\left\| \int_0^t e^{-2\lambda(t-\nu)} \Phi_3(x, x_r) d\nu \right\| \leq \frac{\Delta_\theta \rho_a}{2\sqrt{\Gamma}}, \quad (\text{B.47})$$

for all  $t \in [0, \tau^*]$ , where  $\Delta_\theta$  is defined in (3.21). Substituting (B.46) and (B.47) into (B.37), we obtain the following bound on the Riemannian energy

$$\mathcal{E}(x_r, x) \leq \zeta_2(\omega)\rho_a^2 + \zeta_3(\omega)\rho_a^2 + \frac{\Delta_\theta \rho_a}{\sqrt{\Gamma}},$$

for all  $t \in [0, \tau^*]$ . Recall from the contradiction argument that  $\|x_r(\tau^*) - x(\tau^*)\| = \rho_a$ . Since  $\mathcal{E}(x_r, x) \geq \underline{\alpha}\|x_r(t) - x(t)\|^2$  from Lemma B.3, the following inequality must be satisfied at  $t = \tau^*$

$$\underline{\alpha}\rho_a^2 \leq \zeta_2(\omega)\rho_a^2 + \zeta_3(\omega)\rho_a^2 + \frac{\Delta_\theta \rho_a}{\sqrt{\Gamma}} \implies \sqrt{\Gamma} \leq \frac{\Delta_\theta}{\rho_a(\underline{\alpha} - \zeta_2(\omega) - \zeta_3(\omega))}.$$

However, from (3.32c) the adaptation rate is chosen such that

$$\sqrt{\Gamma} > \frac{\Delta_\theta}{\rho_a(\underline{\alpha} - \zeta_2(\omega) - \zeta_3(\omega))}.$$

This directly contradicts our earlier statement, and therefore  $\|x_r - x\|_{\mathcal{L}_\infty}^{[0, \tau]} < \rho_a$ . □

*Proof of Theorem 3.2.* We will prove this by contradiction: assume that

$$\|x^* - x\|_{\mathcal{L}_\infty} \geq \rho.$$

From the definition of  $\rho_r$  in (3.29), we have  $\|x^*(0) - x(0)\| < \rho_r$ , which implies that  $\|x^*(0) - x(0)\| < \rho$ . Then there must exist a  $\tau^* > 0$  such that

$$\|x^*(\tau^*) - x(\tau^*)\| = \rho, \tag{B.48}$$

$$\|x^*(t) - x(t)\| < \rho, \quad t \in [0, \tau^*). \tag{B.49}$$

According to Lemma 3.1, we have  $\|x^* - x_r\|_{\mathcal{L}_\infty} < \rho_r$ , and from Lemma 3.2 it follows that  $\|x_r - x\|_{\mathcal{L}_\infty}^{[0, \tau^*]} < \rho_a$ . Therefore, from triangle inequality we have  $\|x^* - x\|_{\mathcal{L}_\infty}^{[0, \tau^*]} < \rho_r + \rho_a = \rho$ , which implies that  $\|x^*(\tau^*) - x(\tau^*)\| < \rho$ . This directly contradicts our assumption in (B.48), and therefore  $\|x^* - x\|_{\mathcal{L}_\infty} < \rho$ . Additionally, from Lemma 3.1 the reference system satisfies a uniform ultimate bound given by  $\|x^*(t) - x(t)\| \leq \epsilon(\omega, T)$  for all  $t \geq T \geq 0$ . Therefore,  $\|x^*(t) - x(t)\| < \epsilon(\omega, T) + \rho_a$  for all  $t \geq T \geq 0$ .  $\square$

## References

- [1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [2] R. E. Mahony, V. Kumar, and P. I. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” *IEEE Robotics & Automation Magazine*, vol. 19, pp. 20–32, 2012.
- [3] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, “Continuum robots for medical applications: A survey,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.
- [4] R. D’Andrea, “Guest editorial: A revolution in the warehouse: A retrospective on KIVA systems and the grand challenges ahead,” *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 4, pp. 638–639, 2012.
- [5] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, “Optimization based full body control for the ATLAS robot,” in *2014 IEEE-RAS International Conference on Humanoid Robots*. Madrid, Spain: IEEE, 2014, pp. 120–127.
- [6] M. Kleinbort, O. Salzman, and D. Halperin, “Collision detection or nearest-neighbor search? On the computational bottleneck in sampling-based motion planning,” *arXiv preprint arXiv:1607.04800*, 2016.
- [7] M. Greiff, “Modelling and control of the Crazyflie quadrotor for aggressive and autonomous flight by optical flow driven state estimation,” M.S. thesis, Lund University, 2017.
- [8] D. Sun, S. Jha, and C. Fan, “Learning certified control using contraction metric,” in *Conference on Robot Learning (CoRL)*, Virtual Conference, 2020.
- [9] S. Waslander and C. Wang, “Wind disturbance estimation and rejection for quadrotor position control,” in *AIAA Infotech@Aerospace Conference*, Seattle, WA, USA, 2009.
- [10] G. Williams, B. Goldfain, P. Drews, K. Saigol, J. Rehg, and E. Theodorou, “Robust sampling based model predictive control with sparse objective information,” in *Proceedings of Robotics: Science and Systems*, Pittsburgh, PA, USA, June 2018.
- [11] Q. Nguyen and K. Sreenath, “Optimal robust control for bipedal robots through control lyapunov function based quadratic programs,” in *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

- [12] K. Zhou and J. C. Doyle, *Essentials of Robust Control*. Prentice hall Upper Saddle River, NJ, 1998, vol. 104.
- [13] G. E. Dullerud and F. Paganini, *A Course in Robust Control Theory: A Convex Approach*. Springer Science & Business Media, 2013, vol. 36.
- [14] K. J. Åström and B. Wittenmark, *Adaptive Control*. Courier Corporation, 2013.
- [15] N. Hovakimyan and C. Cao,  $\mathcal{L}_1$  *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. SIAM, 2010.
- [16] T. Lee, “Robust adaptive attitude tracking on  $SO(3)$  with an application to a quadrotor UAV,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1924–1930, 2012.
- [17] Q. Nguyen and K. Sreenath, “ $\mathcal{L}_1$  adaptive control for bipedal robots with control Lyapunov function based quadratic programs,” in *2015 American Control Conference (ACC)*. Chicago, IL, USA: IEEE, 2015, pp. 862–867.
- [18] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, “Regret bounds for robust adaptive control of the linear quadratic regulator,” *arXiv preprint arXiv:1805.09388*, 2018.
- [19] M. Deisenroth and C. E. Rasmussen, “PILCO: A model-based and data-efficient approach to policy search,” in *Proceedings of the 28th International Conference on machine learning (ICML-11)*. Bellevue, WA, USA: Citeseer, 2011, pp. 465–472.
- [20] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, “Safe model-based reinforcement learning with stability guarantees,” in *Proceedings of 31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 908–918.
- [21] L. Hewing, J. Kabzan, and M. N. Zeilinger, “Cautious model predictive control using Gaussian process regression,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019.
- [22] B. Amos, I. D. J. Rodriguez, J. Sacks, B. Boots, and J. Z. Kolter, “Differentiable MPC for end-to-end planning and control,” *arXiv preprint arXiv:1810.13400*, 2018.
- [23] I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel, “Model-based reinforcement learning via meta-policy optimization,” in *Conference on Robot Learning*. Zürich, Switzerland: PMLR, 2018, pp. 617–629.
- [24] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, “Sample-efficient reinforcement learning with stochastic ensemble value expansion,” *arXiv preprint arXiv:1807.01675*, 2018.
- [25] A. Papachristodoulou and S. Prajna, “On the construction of Lyapunov functions using the sum of squares decomposition,” in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 3. IEEE, 2002, pp. 3482–3487.

- [26] E. M. Aylward, P. A. Parrilo, and J.-J. E. Slotine, “Stability and robustness analysis of nonlinear systems via contraction metrics and SOS programming,” *Automatica*, vol. 44, no. 8, pp. 2163–2170, 2008.
- [27] Y.-C. Chang, N. Roohi, and S. Gao, “Neural Lyapunov control,” in *Proceedings of the 33rd Conference on Neural Information Processing Systems*, vol. 32, Vancouver, Canada, 2019.
- [28] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.
- [29] I. M. Ross and M. Karpenko, “A review of pseudospectral optimal control: From theory to flight,” *Annual Reviews in Control*, vol. 36, no. 2, pp. 182–197, 2012.
- [30] C. Richter, A. Bry, and N. Roy, “Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments,” in *Robotics Research*. Springer, 2016, pp. 649–666.
- [31] V. Cichella, I. Kaminer, C. Walton, and N. Hovakimyan, “Optimal motion planning for differentially flat systems using Bernstein approximation,” *IEEE Control Systems Letters*, vol. 2, no. 1, pp. 181–186, 2017.
- [32] D. Manocha and J. Demmel, “Algorithms for intersecting parametric and algebraic curves I: Simple intersections,” *ACM Transactions on Graphics (TOG)*, vol. 13, no. 1, pp. 73–100, 1994.
- [33] G. Elber and T. Grandine, “Hausdorff and minimal distances between parametric freeforms in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ ,” in *International Conference on Geometric Modeling and Processing*. Hangzhou, China: Springer, 2008, pp. 191–204.
- [34] X.-D. Chen, L. Chen, Y. Wang, G. Xu, J.-H. Yong, and J.-C. Paul, “Computing the minimum distance between two Bézier curves,” *Journal of Computational and Applied Mathematics*, vol. 229, no. 1, pp. 294–301, 2009.
- [35] Y.-T. Oh, Y.-J. Kim, J. Lee, M.-S. Kim, and G. Elber, “Efficient point-projection to freeform curves and surfaces,” *Computer Aided Geometric Design*, vol. 29, no. 5, pp. 242–254, 2012.
- [36] J.-W. Chang, Y.-K. Choi, M.-S. Kim, and W. Wang, “Computation of the minimum distance between two Bézier curves/surfaces,” *Computers & Graphics*, vol. 35, no. 3, pp. 677–684, 2011.
- [37] Y. Tassa, T. Erez, and E. Todorov, “Synthesis and stabilization of complex behaviors through online trajectory optimization,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, 2012, pp. 4906–4913.
- [38] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, “Information-theoretic model predictive control: Theory and applications to autonomous driving,” *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018.
- [39] S. M. LaValle and J. J. Kuffner Jr, “Randomized kinodynamic planning,” *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.



- [40] T. A. Howell, B. E. Jackson, and Z. Manchester, “ALTRO: A fast solver for constrained trajectory optimization,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Macau, China, 2019, pp. 7674–7679.
- [41] A. Lakshmanan, A. Gahlawat, and N. Hovakimyan, “Safe feedback motion planning: A contraction theory and  $\mathcal{L}_1$ -adaptive control based approach,” *arXiv preprint arXiv:2004.01142*, 2020.
- [42] E. D. Sontag, “A ‘universal’ construction of Artstein’s theorem on nonlinear stabilization,” *Systems & control letters*, vol. 13, no. 2, pp. 117–123, 1989.
- [43] A. Lakshmanan, A. Patterson, V. Cichella, and N. Hovakimyan, “Proximity queries for absolutely continuous parametric curves,” in *Proceedings of Robotics: Science and Systems*, Freiburg im Breisgau, Germany, June 2019.
- [44] A. Gahlawat, A. Lakshmanan, L. Song, A. Patterson, Z. Wu, N. Hovakimyan, and E. A. Theodorou, “Contraction  $\mathcal{L}_1$ -adaptive control using gaussian processes,” in *Learning for Dynamics and Control*. Virtual Conference: PMLR, 2021, pp. 1027–1040.
- [45] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, “A fast procedure for computing the distance between complex objects in three-dimensional space,” *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.
- [46] M. C. Lin and J. F. Canny, “A fast algorithm for incremental distance calculation,” in *IEEE International Conference on Robotics and Automation (ICRA), 1991*. Sacramento, CA, USA: IEEE, 1991, pp. 1008–1014.
- [47] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, “Fast proximity queries with swept sphere volumes,” TR99-018, Department of Computer Science, University of North Carolina, Tech. Rep., 1999.
- [48] S. Gottschalk, M. C. Lin, and D. Manocha, “OBBTree: A hierarchical structure for rapid interference detection,” in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: ACM, 1996, pp. 171–180.
- [49] S. A. Ehmann and M. C. Lin, “Accurate and fast proximity queries between polyhedra using convex surface decomposition,” in *Computer Graphics Forum*, vol. 20, no. 3. Manchester, UK: Wiley Online Library, 2001, pp. 500–511.
- [50] X. Zhang, M. Lee, and Y. J. Kim, “Interactive continuous collision detection for non-convex polyhedra,” *The Visual Computer*, vol. 22, no. 9-11, pp. 749–760, 2006.
- [51] M. Tang, Y. J. Kim, and D. Manocha, “ $C^2A$ : Controlled conservative advancement for continuous collision detection of polygonal models,” in *IEEE International Conference on Robotics and Automation (ICRA), 2009*. Kobe, Japan: IEEE, 2009, pp. 849–854.
- [52] J. Pan, L. Zhang, and D. Manocha, “Collision-free and smooth trajectory computation in cluttered environments,” *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1155–1175, 2012.

- [53] D. Mellinger, A. Kushleyev, and V. Kumar, “Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. Saint Paul, MN, USA: IEEE, 2012, pp. 477–483.
- [54] V. Cichella, I. Kaminer, C. Walton, and N. Hovakimyan, “Optimal motion planning for differentially flat systems using Bernstein approximation,” *IEEE Control Systems Letters*, vol. 2, no. 1, pp. 181–186, 2018.
- [55] V. Cichella, I. Kaminer, C. Walton, N. Hovakimyan, and A. Pascoal, “Bernstein approximation of optimal control problems,” *arXiv preprint arXiv:1812.06132*, 2018.
- [56] L. A. Ricciardi and M. Vasile, “Direct transcription of optimal control problems with finite elements on Bernstein basis,” *Journal of Guidance, Control, and Dynamics*, pp. 1–15, 2018.
- [57] W. Van Loock, G. Pipeleers, and J. Swevers, “Optimal motion planning for differentially flat systems with guaranteed constraint satisfaction,” in *American Control Conference (ACC), 2015*. Chicago, IL, USA: IEEE, 2015, pp. 4245–4250.
- [58] P. Váňa and J. Faigl, “Optimal solution of the generalized Dubins interval problem,” in *Proceedings of Robotics: Science and Systems*, Pittsburgh, PA, USA, June 2018.
- [59] J. Puig-Navarro, N. Hovakimyan, N. Alexandrov, and B. D. Allen, “Silhouette-informed trajectory generation through a wire maze for UAS,” in *2018 Aviation Technology, Integration, and Operations Conference*, Atlanta, GA, USA, 2018, p. 3845.
- [60] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis: With examples in parameter and state estimation, robust control and robotics*. Springer Science & Business Media, 2001, vol. 1.
- [61] K. Iseki, “On certain properties of parametric curves,” *Journal of the Mathematical Society of Japan*, vol. 12, no. 2, pp. 129–173, 1960.
- [62] A. N. Kolmogorov and S. V. Fomin, *Elements of the Theory of Functions and Functional Analysis*. Dover, 1957.
- [63] E. Rimón and S. P. Boyd, “Obstacle collision detection using best ellipsoid fit,” *Journal of Intelligent and Robotic Systems*, vol. 18, no. 2, pp. 105–126, 1997.
- [64] N. Chakraborty, J. Peng, S. Akella, and J. E. Mitchell, “Proximity queries between convex objects: An interior point approach for implicit surfaces,” *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 211–220, 2008.
- [65] T. W. Sederberg, S. C. White, and A. K. Zundel, “Fat arcs: A bounding region with cubic convergence,” *Computer Aided Geometric Design*, vol. 6, no. 3, pp. 205–218, 1989.
- [66] M. Bartoň and G. Elber, “Spiral fat arcs—Bounding regions with cubic convergence,” *Graphical Models*, vol. 73, no. 2, pp. 50–57, 2011.
- [67] S. Boyd and J. Mattingley, “Branch and bound methods,” *Notes for EE364b, Stanford University*, 2007.

- [68] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, “Julia: A fresh approach to numerical computing,” *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.
- [69] G. Farin and D. Hansford, *The essentials of CAGD*. AK Peters/CRC Press, 2000.
- [70] R. T. Farouki, “The Bernstein polynomial basis: A centennial retrospective,” *Computer Aided Geometric Design*, vol. 29, no. 6, pp. 379–419, 2012.
- [71] E. W. Weisstein, “Heart curve. From MathWorld—A Wolfram Web Resource,” <https://mathworld.wolfram.com/HeartCurve.html>, 2003.
- [72] E. W. Weisstein, “Ranunculoid. From MathWorld—A Wolfram Web Resource,” <https://mathworld.wolfram.com/Ranunculoid.html>, 2003.
- [73] E. W. Weisstein, “Fish curve. From MathWorld—A Wolfram Web Resource,” <https://mathworld.wolfram.com/FishCurve.html>, 2003.
- [74] T. Hermann, “On the stability of polynomial transformations between Taylor, Bernstein and Hermite forms,” *Numerical algorithms*, vol. 13, no. 2, pp. 307–320, 1996.
- [75] M. Kobilarov, “Cross-entropy motion planning,” *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 855–871, 2012.
- [76] A. Patterson, A. Lakshmanan, and N. Hovakimyan, “Intent-aware probabilistic trajectory estimation for collision prediction with uncertainty quantification,” *arXiv preprint arXiv:1904.02765*, 2019.
- [77] R. Mahony and T. Hamel, “Robust trajectory tracking for a scale model autonomous helicopter,” *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 14, no. 12, pp. 1035–1059, 2004.
- [78] S. Jeong and D. Chwa, “Coupled multiple sliding-mode control for robust trajectory tracking of hovercraft with external disturbances,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4103–4113, 2017.
- [79] A. Donaire, J. G. Romero, and T. Perez, “Trajectory tracking passivity-based control for marine vehicles subject to disturbances,” *Journal of the Franklin Institute*, vol. 354, no. 5, pp. 2167–2182, 2017.
- [80] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice Hall Upper Saddle River, NJ, 2002, vol. 3.
- [81] L. Magni, H. Nijmeijer, and A. van der Schaft, “A receding-horizon approach to the nonlinear  $H_\infty$  control problem,” *Automatica*, vol. 37, pp. 429–435, 2001.
- [82] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. F. Camacho, “Min-max model predictive control of nonlinear systems: A unifying overview on stability,” *European Journal of Control*, vol. 15, no. 1, pp. 5–21, 2009.

- [83] X. Wang, L. Yang, Y. Sun, and K. Deng, “Adaptive model predictive control of nonlinear systems with state-dependent uncertainties,” *International Journal of Robust and Nonlinear Control*, vol. 27, no. 17, pp. 4138–4153, 2017.
- [84] S. V. Raković, “Set theoretic methods in model predictive control,” in *Nonlinear Model Predictive Control*. Springer, 2009, pp. 41–54.
- [85] S. V. Raković, W. S. Levine, and B. Açikmeşe, “Elastic tube model predictive control,” in *2016 American Control Conference (ACC)*. Boston, MA, USA: IEEE, 2016, pp. 3594–3599.
- [86] S. Yu, C. Böhm, H. Chen, and F. Allgöwer, “Robust model predictive control with disturbance invariant sets,” in *Proceedings of the 2010 American Control Conference*. Baltimore, MD, USA: IEEE, 2010, pp. 6262–6267.
- [87] B. T. Lopez, J. P. How, and J.-J. E. Slotine, “Dynamic tube MPC for nonlinear systems,” in *Proceedings of the 2019 American Control Conference*. Philadelphia, PA, USA: IEEE, 2019, pp. 1655–1662.
- [88] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *Proceedings of 18th European Control Conference*, Naples, Italy, 2019, pp. 3420–3431.
- [89] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [90] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, “Robustness of control barrier functions for safety critical control,” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.
- [91] B. T. Lopez, J.-J. E. Slotine, and J. P. How, “Robust adaptive control barrier functions: An adaptive & data-driven approach to safety,” *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1031–1036, 2020.
- [92] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3046–3053, 2017.
- [93] I. R. Manchester, J. Z. Tang, and J.-J. E. Slotine, “Unifying robot trajectory tracking with control contraction metrics,” in *Robotics Research*. Springer, 2018, pp. 403–418.
- [94] W. Lohmiller and J.-J. E. Slotine, “On contraction analysis for non-linear systems,” *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.
- [95] S. Singh, B. Landry, A. Majumdar, J.-J. Slotine, and M. Pavone, “Robust feedback motion planning via contraction theory,” *The International Journal of Robotics Research*, 2019, submitted.

- [96] P. Zhao, A. Lakshmanan, K. Ackerman, A. Gahlawat, M. Pavone, and N. Hovakimyan, “Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics,” *arXiv preprint arXiv:2109.04453*, 2021.
- [97] B. T. Lopez and J.-J. E. Slotine, “Contraction metrics in adaptive nonlinear control,” *arXiv preprint arXiv:1912.13138*, 2019.
- [98] I. Gregory, E. Xargay, C. Cao, and N. Hovakimyan, “Flight test of an  $\mathcal{L}_1$  adaptive controller on the NASA AirSTAR flight test vehicle,” in *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario, Canada, August 2010, AIAA 2010-8015.
- [99] K. A. Ackerman, E. Xargay, R. Choe, N. Hovakimyan, M. C. Cotting, R. B. Jeffrey, M. P. Blackstun, T. P. Fulkerson, T. R. Lau, and S. S. Stephens, “Evaluation of an  $\mathcal{L}_1$  adaptive flight control law on Calspan’s variable-stability Learjet,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 40, no. 4, pp. 1051–1060, 2017.
- [100] I. Kaminer, A. Pascoal, E. Xargay, N. Hovakimyan, C. Cao, and V. Dobrokhodov, “Path following for small unmanned aerial vehicles using  $\mathcal{L}_1$  adaptive augmentation of commercial autopilots,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 550–564, 2010.
- [101] H. Jafarnejadsani, D. Sun, H. Lee, and N. Hovakimyan, “Optimized  $\mathcal{L}_1$  adaptive controller for trajectory tracking of an indoor quadrotor,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 40, no. 6, pp. 1415–1427, 2017.
- [102] R. M. Jones, D. Sun, G. Barsi Haberfeld, A. Lakshmanan, T. Marinho, and N. Hovakimyan, “Design and control of a small aerial manipulator for indoor environments,” in *AIAA Information Systems-AIAA Infotech@ Aerospace*, Grapevine, TX, USA, 2017, p. 1374.
- [103] J. Pravitra, K. A. Ackerman, C. Cao, N. Hovakimyan, and E. A. Theodorou, “ $\mathcal{L}_1$ -adaptive MPPI architecture for robust and agile control of multirotors,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Online, 2020, pp. 7661–7666.
- [104] Z. Wu, S. Cheng, K. A. Ackerman, A. Gahlawat, A. Lakshmanan, P. Zhao, and N. Hovakimyan, “ $\mathcal{L}_1$  adaptive augmentation for geometric tracking control of quadrotors,” *arXiv preprint arXiv:2109.06998*, 2021.
- [105] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*. Springer, 2019, vol. 49.
- [106] P. E. Crouch and A. J. van der Schaft, *Variational and Hamiltonian control systems*. Springer, 1987.
- [107] M. P. Do Carmo, *Riemannian geometry*. Birkhäuser, 1992.
- [108] E. Lavretsky, T. E. Gibson, and A. M. Annaswamy, “Projection operator in adaptive systems,” *arXiv preprint arXiv:1112.4232*, 2011.

- [109] J.-B. Pomet and L. Praly, “Adaptive nonlinear regulation: Estimation from the Lyapunov equation,” *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 729–740, 1992.
- [110] C. Cao and N. Hovakimyan, “Stability margins of  $\mathcal{L}_1$  adaptive control architecture,” *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 480–487, 2010.
- [111] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *Proceedings of IEEE International Conference on Robotics and Automation*, Marina Bay Sands, Singapore, 2017, pp. 5883–5890.
- [112] B. Legat, T. Weisser, L. Kapelevich, J. Huchette, A. Bhatia, M. Forets, T. Kelman, J. TagBot, E. Hanson, E. Saba, C. C., and C. Coffrin, “JuliaOpt/SumOfSquares.jl: v0.4.0,” Feb. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.3687019>
- [113] I. Dunning, J. Huchette, and M. Lubin, “JuMP: A modeling language for mathematical optimization,” *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.
- [114] MOSEK ApS, *Mosek.jl: Interface to the Mosek solver in Julia. Version 1.1.2*, 2020. [Online]. Available: <https://github.com/JuliaOpt/Mosek.jl>
- [115] T. A. Howell, B. E. Jackson, and Z. Manchester, “TrajectoryOptimization.jl,” Jan. 2020. [Online]. Available: <https://github.com/RoboticExplorationLab/TrajectoryOptimization.jl>
- [116] B. Recht, “A tour of reinforcement learning: The view from continuous control,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019.
- [117] J. C. Knight, “Safety critical systems: Challenges and directions,” in *Proceedings of the 24th International Conference on Software Engineering*, Orlando, FL, USA, 2002, pp. 547–550.
- [118] T. J. Perkins and A. G. Barto, “Lyapunov design for safe reinforcement learning,” *Journal of Machine Learning Research*, vol. 3, no. 12, pp. 803–832, 2002.
- [119] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, “A Lyapunov-based approach to safe reinforcement learning,” in *Proceedings of 32nd Conference on Neural Information Processing Systems*, Quebec, Canada, 2018, pp. 8092–8101.
- [120] H. K. Khalil, *Nonlinear control*. London, UK: Pearson, 2014.
- [121] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, “Provably safe and robust learning-based model predictive control,” *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.
- [122] K. P. Wabersich and M. N. Zeilinger, “Linear model predictive safety certification for learning-based control,” in *Proceedings of 57th IEEE Conference on Decision and Control*, Miami Beach, FL, USA, 2018, pp. 7130–7135.
- [123] R. Soloperto, M. A. Müller, S. Trimpe, and F. Allgöwer, “Learning-based robust model predictive control with state-dependent uncertainty,” *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 442–447, 2018.

- [124] U. Rosolia and F. Borrelli, “Sample-based learning model predictive control for linear uncertain systems,” in *Proceedings of 58th IEEE Conference on Decision and Control*, Nice, France, 2019, pp. 2702–2707.
- [125] T. Beckers, D. Kulić, and S. Hirche, “Stable gaussian process based tracking control of euler–lagrange systems,” *Automatica*, vol. 103, pp. 390–397, 2019.
- [126] A. Capone and S. Hirche, “Backstepping for partially unknown nonlinear systems using gaussian processes,” *IEEE Control Systems Letters*, vol. 3, no. 2, pp. 416–421, 2019.
- [127] J. Umlauft and S. Hirche, “Feedback linearization based on gaussian processes with event-triggered online learning,” *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4154–4169, 2019.
- [128] M. K. Helwa, A. Heins, and A. P. Schoellig, “Provably robust learning-based approach for high-accuracy tracking control of lagrangian systems,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1587–1594, 2019.
- [129] M. Greeff and A. P. Schoellig, “Exploiting differential flatness for robust learning-based tracking control using gaussian processes,” *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1121–1126, 2020.
- [130] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, “Learning-based model predictive control for safe exploration,” in *Proceedings of 57th IEEE Conference on Decision and Control*, Miami Beach, FL, USA, 2018, pp. 6059–6066.
- [131] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, “Robust constrained learning-based NMPC enabling reliable mobile robot path tracking,” *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1547–1563, 2016.
- [132] B. T. Lopez and J.-J. E. Slotine, “Adaptive nonlinear control with contraction metrics,” *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 205–210, 2020.
- [133] K. Pereida and A. P. Schoellig, “Adaptive model predictive control for high-accuracy trajectory tracking in changing conditions,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, Spain, 2018, pp. 7831–7837.
- [134] N. Wagener, C. an Cheng, J. Sacks, and B. Boots, “An online learning approach to model predictive control,” in *Proceedings of Robotics: Science and Systems*, Freiburg im Breisgau, Germany, 2019.
- [135] W. L. Garrard and J. M. Jordan, “Design of nonlinear automatic flight control systems,” *Automatica*, vol. 13, no. 5, pp. 497–505, 1977.
- [136] A. Mokhtari, A. Benallegue, and Y. Orlov, “Exact linearization and sliding mode observer for a quadrotor unmanned aerial vehicle,” *International Journal of Robotics and Automation*, vol. 21, no. 1, pp. 39–49, 2006.

- [137] I. R. Manchester and J.-J. E. Slotine, “Robust control contraction metrics: A convex approach to nonlinear state-feedback  $\mathcal{H}_\infty$  control,” *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 333–338, 2018.
- [138] A. Van Der Vaart and H. Van Zanten, “Information rates of nonparametric Gaussian process methods,” *Journal of Machine Learning Research*, vol. 12, no. 6, 2011.
- [139] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. Cambridge, MA, USA: MIT press, 2006.
- [140] A. Gahlawat, A. Lakshmanan, L. Song, A. Patterson, Z. Wu, N. Hovakimyan, and E. Theodorou, “Contraction  $\mathcal{L}_1$ -adaptive control using gaussian processes,” *arXiv:2009.03864*, 2020.
- [141] A. Gahlawat, P. Zhao, A. Patterson, N. Hovakimyan, and E. Theodorou, “ $\mathcal{L}_1$ -GP:  $\mathcal{L}_1$  adaptive control with Bayesian learning,” in *Proceedings of 2nd Learning for Dynamics & Control*, vol. 120, Online, 2020, pp. 1–12.
- [142] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Batch informed trees (BIT\*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA: IEEE, 2015, pp. 3067–3074.
- [143] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [144] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, “Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search,” in *Proceedings of the IEEE International Conference on Robotics and Automation*. Stockholm, Sweden: IEEE, 2016, pp. 528–535.
- [145] S. Prajna, A. Papachristodoulou, and F. Wu, “Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach,” in *Proceedings of the 5th Asian Control Conference*, vol. 1, Melbourne, Australia, 2004, pp. 157–165.
- [146] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard, “Control applications of sum of squares programming,” in *Positive Polynomials in Control*. Berlin, Germany: Springer, 2005, vol. 312, pp. 3–22.
- [147] A. Majumdar, A. A. Ahmadi, and R. Tedrake, “Control design along trajectories with sums of squares programming,” in *Proceedings of the IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, 2013, pp. 4054–4061.
- [148] A. A. Ahmadi, G. Hall, A. Papachristodoulou, J. Saunderson, and Y. Zheng, “Improving efficiency and scalability of sum of squares optimization: Recent advances and limitations,” in *Proceedings of the 56th IEEE Conference on Decision and Control*, Melbourne, Australia, 2017, pp. 453–462.



- [149] S. M. Richards, F. Berkenkamp, and A. Krause, “The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems,” in *Proceedings of the 2nd Conference on Robot Learning*, Zurich, Switzerland, 2018, pp. 466–476.
- [150] L. Pulina and A. Tacchella, “Challenging SMT solvers to verify neural networks,” *AI Communications*, vol. 25, no. 2, pp. 117–135, 2012.
- [151] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, “Reluplex: An efficient SMT solver for verifying deep neural networks,” in *Proceedings of the 29th Conference on Computer Aided Verification*, Heidelberg, Germany, 2017, pp. 97–117.
- [152] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, and M. J. Kochenderfer, “Algorithms for verifying deep neural networks,” *arXiv preprint arXiv:1903.06758*, 2019.
- [153] D. Angeli, “A Lyapunov approach to incremental stability properties,” *IEEE Transactions on Automatic Control*, vol. 47, no. 3, pp. 410–421, 2002.
- [154] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control Systems Letters*, 2020.
- [155] R. E. Moore, *Interval analysis*. Prentice-Hall Englewood Cliffs, 1966, vol. 4.
- [156] T. Hickey, Q. Ju, and M. H. Van Emden, “Interval arithmetic: From principles to implementation,” *Journal of the ACM (JACM)*, vol. 48, no. 5, pp. 1038–1068, 2001.
- [157] L. B. Rall, “Mean value and Taylor forms in interval analysis,” *SIAM Journal on Mathematical Analysis*, vol. 14, no. 2, pp. 223–238, 1983.
- [158] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, “Incorporating second-order functional knowledge for better option pricing,” *Advances in neural information processing systems*, pp. 472–478, 2001.
- [159] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” *arXiv preprint arXiv:1710.05941*, 2017.
- [160] D. Misra, “Mish: A self regularized non-monotonic neural activation function,” *arXiv preprint arXiv:1908.08681*, vol. 4, 2019.
- [161] G. Manek and J. Z. Kolter, “Learning stable deep dynamics models,” in *Proceedings of the 33rd Conference on Neural Information Processing Systems*, vol. 32, Vancouver, Canada, 2019.
- [162] B. Amos, L. Xu, and J. Z. Kolter, “Input convex neural networks,” in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017, pp. 146–155.
- [163] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson et al., “JAX: composable transformations of Python+NumPy programs,” <http://github.com/google/jax>, 2018.
- [164] J. Revels, M. Lubin, and T. Papamarkou, “Forward-mode automatic differentiation in Julia,” *arXiv:1607.07892 [cs.MS]*, 2016. [Online]. Available: <https://arxiv.org/abs/1607.07892>

- [165] S. M. Rump, “INTLAB—interval laboratory,” in *Developments in reliable computing*. Springer, 1999, pp. 77–104.
- [166] D. P. Sanders, L. Benet et al., “JuliaIntervals/IntervalArithmetic.jl: v0.18.2,” <https://doi.org/10.5281/zenodo.4739394>, May 2021.
- [167] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, “Basic linear algebra subprograms for FORTRAN usage,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 5, no. 3, pp. 308–323, 1979.
- [168] D. Hertz, “The extreme eigenvalues and stability of real symmetric interval matrices,” *IEEE Transactions on Automatic Control*, vol. 37, no. 4, pp. 532–535, 1992.
- [169] J. Rohn, “A handbook of results on interval linear problems,” 2005.
- [170] M. W. Spong, “Energy based control of a class of underactuated mechanical systems,” *IFAC Proceedings volumes*, vol. 29, no. 1, pp. 2828–2832, 1996.
- [171] D. A. McAllester, “Some PAC-Bayesian theorems,” *Machine Learning*, vol. 37, no. 3, pp. 355–363, 1999.
- [172] S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari, “Output range analysis for deep feed-forward neural networks,” in *NASA Formal Methods Symposium*. Newport News, VA, USA: Springer, 2018, pp. 121–138.
- [173] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*. Springer Science & Business Media, 2006, vol. 176.
- [174] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [175] T. M. Apostol, *Calculus, Volume 1: One-Variable Calculus with an Introduction to Linear Algebra*, 2nd ed. John Wiley & Sons, 1991.