Continuous-Time Planning and Control Synthesis under Temporal Logic Specifications with Applications to Quadrotors

by

Yating Yuan

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

Waterloo, Ontario, Canada, 2026

© Yating Yuan 2026

Examining Committee Membership

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

External Examiner: Jie Fu
Associate Professor
Department of Electrical and Computer Engineering
University of Florida

Supervisor(s): Jun Liu
Professor
Department of Applied Mathematics

Internal Member: Hans De Sterck
Professor
Department of Applied Mathematics

Roberto Guglielmi
Associate Professor
Department of Applied Mathematics

Internal-external Member: Yash Vardhan Pant
Assistant Professor
Department of Electrical and Computer Engineering

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

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

Statement of Contributions

The research presented in this thesis has resulted in four papers, including two published papers, one paper currently under review, and one paper in preparation for submission. My contributions to each chapter are detailed below:

- For Chapters 1 and 2, I am the sole author.
- Chapter 3 is based on the paper [155], published in *IEEE Control Systems Letters*. I led the development of the main methodology and algorithmic framework, designed and conducted the numerical experiments, and prepared the manuscript.
- Chapter 4 is based on two papers [127, 154]. The first paper is published in *IEEE Transactions on Automatic Control* [127]. My contribution to the first paper was the experimental evaluation, including numerical studies involving geometric control, nonlinear model predictive control, and control barrier function-based comparison methods. The second paper is available as an arXiv preprint and has been submitted to *Nonlinear Analysis: Hybrid Systems*, where it is currently under review [154]. My contribution to the second paper was the derivation of the diagonal-matrix-gain geometric control framework, the associated mathematical analysis, and the implementation of the numerical experiments.
- Chapter 5 is also based on [154]. Building on the continuous-time signal temporal logic planning framework in [155], I developed the multi-agent extension, carried out the mathematical derivations, designed the numerical experiments, and contributed to the manuscript writing.
- Chapter 6 presents work from a journal manuscript developed with *IEEE Control Systems Letters* as the intended target venue. My contributions included formulating the linear temporal logic specifications and integrating them into the proposed control synthesis framework, designing and implementing simulation studies to evaluate the framework, and developing a control barrier function-based quadratic programming benchmark for comparative evaluation.
- For Chapter 7, I am the sole author.

Abstract

Modern robotic systems are increasingly required to perform complex, safety-critical tasks in which system behaviors must be achieved under specified conditions and temporal constraints. Temporal logic provides a formal language for specifying such requirements in a precise and interpretable manner. In this thesis, signal temporal logic (STL) is used for continuous-time trajectory planning with quantitative temporal constraints, while linear temporal logic (LTL) is considered for control synthesis with qualitative task requirements. This thesis focuses on continuous-time planning and control synthesis for robotic systems under temporal logic specifications, with the goal of generating dynamically feasible control strategies that satisfy high-level task requirements.

The first aspect of this thesis considers continuous-time trajectory planning under STL specifications. We represent trajectories using Bézier curves so that the generated plans are continuous in time and possess inherent smoothness properties. Within this trajectory representation, we introduce a time-varying robustness formulation for STL specifications, which replaces the conventional use of a uniform robustness margin over the entire task horizon. The proposed formulation allows different portions of the trajectory to be associated with different robustness requirements, thereby reducing unnecessary conservatism while maintaining satisfaction of the STL specification. This framework connects continuous-time trajectory generation with STL robustness analysis and yields smooth trajectories that are suitable for control execution.

The second aspect of this thesis studies control synthesis for executing trajectories that satisfy STL specifications. Since the closed-loop trajectory may deviate from the planned trajectory, the proposed framework integrates tracking-error bounds into the STL robustness analysis to ensure that specification satisfaction is preserved during execution. In this thesis, quadrotor systems are considered a representative class of high-dimensional nonlinear robotic systems. Geometric controllers on SE(3) with diagonal matrix gains are developed and analyzed to characterize tracking convergence and execution errors. Compared with controllers using scalar gains, the formulation with diagonal matrix gains yields tighter tracking error bounds, which can be incorporated into the robustness analysis. Building on this framework, the thesis further extends the approach to multi-quadrotor systems by deriving inter-agent safety conditions for STL planning using Bézier curves.

For long-horizon task execution, this thesis develops an abstraction-free LTL control-synthesis framework for continuous-time systems by patching together locally certified control Lyapunov–barrier function (CLBF) solutions. By decomposing temporal-logic specifications into a sequence of safe-stabilization problems, the framework constructs switching

feedback controllers that realize the required transitions over continuous-state regions. This enables efficient execution of long-horizon LTL tasks, without requiring the explicit construction of a global symbolic abstraction.

Through numerical simulations and robotic case studies, we demonstrate the effectiveness of the proposed methods for reach-avoid planning, quadrotor trajectory execution, multi-agent coordination, and long-horizon temporal logic tasks. The proposed LTL control synthesis framework is further demonstrated on a Crazyflie platform.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Jun Liu, for his thoughtful guidance, generous mentorship, and continuous support throughout my research journey. His expertise and insightful feedback were invaluable in helping me develop my ideas into rigorous mathematical formulations, while his trust and encouragement gave me the freedom to explore my research independently. Beyond that, he inspired me to reflect on the essence and philosophy of research itself.

I would like to extend my sincere thanks to my examining committee members, Prof. Hans De Sterck and Prof. Roberto Guglielmi, my internal-external examiner, Prof. Yash Vardhan Pant, and my external examiner, Prof. Jie Fu, for taking the time to read my thesis. I greatly appreciate their expertise and thoughtful engagement with this work. Their insightful comments and constructive feedback have helped to refine and strengthen this work.

I would also like to acknowledge the financial assistance, resources, and other forms of support provided by the Department of Applied Mathematics and the University of Waterloo. My sincere thanks also go to Tina Wang for her kindness and assistance with administrative matters.

I would like to express my heartfelt appreciation to my colleagues at the Hybrid Systems Lab for creating a vibrant and inspiring research community. I truly enjoyed being part of it. I am also grateful to my friends beyond the lab for their companionship, encouragement, and moments of laughter, all of which brought warmth and many cherished memories to my graduate life. In particular, I am deeply grateful to Somayeh Fouladi, who has been like a sister to me and has stood by me through difficult times.

I am endlessly grateful to my family, whose love and support have been the foundation of everything I have achieved. They have encouraged me through uncertainty, celebrated each small step with me, and given me the courage to keep moving forward.

Dedication

To my family, to all who love me, and to all whom I love.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract v

Acknowledgements vii

Dedication viii

List of Figures xiii

List of Tables xvii

List of Abbreviations xviii

List of Symbols xix

1 Introduction 1

1.1 Motivation 1

1.1.1 Planning under Temporal Logic 3

1.1.2 Control Synthesis under Temporal Logic 5

1.2 Contributions and Thesis Organization 6

2 Preliminaries on Temporal Logic and Control **9**
2.1 System Description . . . 9
2.2 Lyapunov Stability . . . 10
2.3 The Special Euclidean Group SE(3) . . . 11
2.3.1 Definitions and Basic Properties . . . 12
2.3.2 Lie Algebra and Rigid-Body Kinematics . . . 12
2.4 Bézier Curves . . . 13
2.4.1 Bernstein Representation . . . 13
2.4.2 Geometric Properties . . . 14
2.5 Signal Temporal Logic . . . 16
2.6 Linear Temporal Logic . . . 18
2.7 Summary . . . 21

3 Signal Temporal Logic Planning with Time-Varying Robustness **22**
3.1 Introduction . . . 22
3.2 Problem Statement . . . 24
3.3 Trajectory Construction . . . 26
3.3.1 Continuity Constraints . . . 26
3.3.2 Kinematic Constraints . . . 26
3.4 Encoding STL Satisfaction with Bézier Curves . . . 27
3.4.1 Encoding STL Satisfaction with Convex Constraints . . . 28
3.4.2 Soundness of STL Encoding . . . 30
3.5 Simulations . . . 35
3.5.1 Mission Scenarios . . . 36
3.5.2 Comparisons . . . 36
3.6 Summary . . . 40

4 Quadrotor STL Control Synthesis with Time-Varying Bounds **41**
4.1 Introduction . . . 41
4.2 Quadrotor Dynamics Model . . . 42
4.3 Geometric Control with Matrix Gains . . . 43
4.4 Problem Statement . . . 48
4.5 Time-varying Tracking Error Bound Analysis . . . 50
4.5.1 Lyapunov Function for Translation Error . . . 50
4.5.2 Lyapunov Function for Rotational Error . . . 50
4.5.3 Time-varying Error Tracking Bounds . . . 52
4.5.4 Optimization-based Control Gains . . . 55
4.6 STL-Constrained Trajectory Generation . . . 56
4.6.1 Continuity Constraints . . . 57
4.6.2 Kinematic Constraints . . . 58
4.7 Simulations . . . 59
4.7.1 Control Gain Optimization . . . 59
4.7.2 Tracking Performance . . . 61
4.8 Summary . . . 74

5 Continuous-Time STL Control Synthesis for Multiple Quadrotors **75**
5.1 Introduction . . . 75
5.2 Problem Statement . . . 76
5.3 Encoding STL Satisfaction for Multi-Agent Systems . . . 77
5.4 Simulations . . . 86
5.4.1 Runtime Analysis . . . 87
5.4.2 Safety Analysis . . . 88
5.4.3 Stability Analysis . . . 92
5.5 Summary . . . 93

6 Control Synthesis for Continuous-Time Systems under Linear Temporal Logic **94**

6.1 Introduction . . . 94

6.2 Preliminaries and Problem Statement . . . 95

6.2.1 State Constraints and Smooth Approximation . . . 96

6.2.2 LTL Satisfaction of Continuous-Time Trajectories . . . 97

6.2.3 Problem Formulation . . . 99

6.3 LTL Specification Decomposition . . . 99

6.4 LTL Control Synthesis with Patching CLBFs . . . 103

6.4.1 Patching CLBFs Construction . . . 103

6.4.2 Finite-Memory Control Synthesis . . . 106

6.5 Case Studies . . . 110

6.6 Summary . . . 117

7 Conclusions and Future Work **118**

References **121**

Appendix **137**

A Appendix for Chapter 4 **138**

A.1 Derivatives of the Error Functions . . . 138

List of Figures

1.1 Quadrotor applications with temporal logic specifications. 2

1.2 Organization of the thesis and relationships among the chapters. 8

2.1 Illustration of a fifth-degree Bézier curve $r(s)$ and its first derivative curve $\dot{r}(s)$. The original curve is defined by six control points $\{c_0, \ldots, c_5\}$, while its derivative is a fourth-degree Bézier curve with control points $c'_i = 5(c_{i+1} - c_i)$, $i \in \{0, \ldots, 4\}$. The shaded regions indicate the convex hulls of the original and derivative control points, respectively. 15

2.2 From system states to a symbolic word. 20

3.1 Comparison of trajectory planning methods on a reach-avoid task: PWL [135], MICP [11], I-Béziers (Bézier curves with improperly chosen control points), and the proposed method. The solid blue lines denote the planned reference trajectories, and the blue circles denote the sampling points. The red dashed lines show the trajectories of a unicycle model controlled by MPC to track the corresponding reference trajectories. The blue shaded regions and blue frames represent the robustness tubes and bounding boxes, respectively. Panels (a)–(c) use uniform real-valued robustness, whereas panel (d) uses the proposed time-varying robustness formulation. 23

3.2 Four benchmark scenarios: (a) reach-avoid, (b) complex reach-avoid, (c) narrow passing, and (d) door puzzle. Left panels show planned and tracked trajectories, denoted by solid and dashed lines, respectively. Blue dots mark the endpoints of the Bézier curves; right panels show tracking errors. The MICP trajectory crosses Region $\mathbf{Y}$ with $N = 11$ in (a); similar cases occur in the other scenarios. For clarity, only the Bézier curves are shown in (b)–(d), and the legend in (a) applies to these panels as well. 39

4.1 The inertial frame $\mathcal{F}_I$, the body frame $\mathcal{F}_B$, and the desired attitude $R_d(t) = \big[b_{d,1}(t),\, b_{d,2}(t),\, b_{d,3}(t)\big]$. 42

4.2 Configuration of the proposed framework. 49

4.3 Environment setup and desired trajectory: The blue curve denotes the desired trajectory parameterized by fourteen Bézier curve segments, with the dots marking the segment endpoints for clarity. The blue tube represents the robustness tube of the Bézier curves. 62

4.4 Environment and Tracking Trajectories. The curves denote tracking trajectories generated from perturbed initial conditions. 64

4.5 Position and velocity tracking errors with theoretical bounds. Panel (a) presents the position tracking errors of DMG-GC, GC, NMPC, and CBF-QP, together with the theoretical position-error bounds for DMG-GC and GC and the STL robustness margin of the desired trajectory. Panel (b) presents the corresponding velocity tracking errors and the theoretical velocity-error bounds for DMG-GC and GC. 66

4.6 Constraint satisfaction across 100 trajectories: $\max_{j\in\{1,\dots,100\}} \|v_j(t)\|_\infty$ and $\max_{j\in\{1,\dots,100\}} |f_j(t)|$ for $t \in [0, T]$. 69

4.7 Condition satisfaction for 100 trajectories: $F_{d,3}(t)$ and $\Psi_K(t)$ over $t \in [0, 20]$. 70

4.8 Robustness under adversarial initial conditions: Position and velocity tracking errors for initial points violating the theoretical conditions and for initial points outside the initial safe set. The plots are restricted to $t \leq 6$ to better show the transient behavior. 72

4.9 DMG-GC results under adversarial initial conditions: Tracking trajectories and initial-point distributions under two types of adversarial initial conditions. The two cases correspond to initial points outside the theoretical error bounds and outside the STL robustness tube at the initial time $t = 0$, respectively. 73

5.1 Two MICP encodings of $\varphi = (\varphi_1 \mathcal{U}_{[0,3]} \varphi_2)$ at time $t = 0$. Dashed circles denote auxiliary binary variables. The bounded-until window contains $N\text{win} = 4$ sampled time instants with sampling interval $\Delta t = 1$. (a) The direct semantic expansion requires $\mathcal{O}(N_{\text{win}}^2)$ logical-variable occurrences and constraints because the prefix conditions $\varphi_1(\cdot)$ (red circles) are repeated for each possible witness time. (b) The backward-recursion encoding introduces auxiliary binary variables for intermediate truth values and requires only $\mathcal{O}(N_{\text{win}})$ auxiliary variables and logical constraints. 86

5.2 Case 0 environment with six quadrotors ($N_\ell = 6$). 87

5.3 Benchmarks and trajectory snapshots for Cases 1–3, shown from top to bottom. For each case, the first two columns present two representative time snapshots, where only the trajectory with the largest $\|e_p\|$ is plotted for clarity. The third column shows an additional view together with the tracking trajectories. In all environment plots, solid lines denote planned trajectories and dashed lines denote tracking trajectories. 90

5.4 Comparison of the position and velocity tracking errors and their corresponding bounds for Cases 1–3, comparing the proposed method with conventional geometric control (GC) [127]. To avoid repetition and improve clarity, the legend in panel (a) also applies to panels (b) and (c). 91

6.1 Visualization on the $x_3 = 0$ slice over the (x_1, x_2)-plane. Left: the certified CLBF for the LTL specification (6.5.1). Right: the projected winning-set approximation, indicated by the dashed green curve, and sampled state trajectories under $u \in [-3, 3]^3$. All state trajectories initialized in this approximation satisfy φ. 111

6.2 Twenty trajectories of the omnidirectional robot initialized from different states in a complex planar environment under the LTL specification using the proposed method. The purple and orange hatched regions denote $\widetilde{\text{Win}}(\varphi)$ and $\widetilde{\text{Win}}(\mathcal{T}_{10})$, respectively. The controller is executed at 50 Hz. . . 113

6.3 Maximum absolute control inputs over 20 trajectories for Case 2. The plotted quantities $|v_{1,\max}(t)|$, $|v_{2,\max}(t)|$, and $|\omega_{\max}(t)|$ remain within the prescribed input bound 3.0, showing that the synthesized controller satisfies the box constraint $u \in [-3, 3]^3$. 115

6.4 Webots simulation scenario and executed trajectory for the Crazyflie quadrotor. The black, green, orange, blue, and pink regions correspond to obstacles, a_0, a_1, a_2, and a_3, respectively. The red curve denotes the trajectory executed by the quadrotor. 116

6.5 The real experiment and executed trajectory for the Crazyflie quadrotor. The colored rectangular boxes represent obstacles. 116

List of Tables

2.1 Representative satisfaction patterns for LTL temporal operators over the positions of an infinite word, adapted from [81]. 19

3.1 Comparisons of benchmarks over time horizons (TH): runtimes, robustness margins, and maximum position tracking errors $e_p^{\max}$. 40

4.1 Physical variables in quadrotor dynamics. 43

4.2 Effects of individual variations of ψ_K, α_ψ, and $\overline{\mathcal{V}}_1$ around the nominal setting $(0.05, 0.7, 0.4)$ on the peak bounds $\mathcal{L}_{\max,p}$, $\mathcal{L}_{\max,v}$, and the IC feasibility rate. 60

4.3 Comparison of tracking runtime and maximum position and velocity tracking errors across 100 trajectories. Tracking errors are evaluated for $t \geq 4\,\mathrm{s}$, and runtime is reported as mean ± standard deviation. 67

5.1 Comparison of runtime (s) for different numbers of robots. 88

5.2 Safety summary across cases, including the number of robots (N_ℓ), number of segments (#S), runtime, worst-case robustness margin $\rho^{\varphi}_{\min}$, and the minimum inter-trajectory distances of the planned and tracking trajectories. 92

5.3 Tracking performance comparison between the proposed method and [127], reported as mean ± standard deviation over pooled trajectories from all agents in each case. Here, $t_{c,p}$ and $t_{c,v}$ denote the convergence times for position and velocity errors, respectively. 93

6.1 Comparison of trajectory performance and computational cost. Values are reported as mean ± sample standard deviation over 20 trajectories. 114

List of Abbreviations

CBF control barrier function, 4–6, 94, 103–105

CBF-QP control barrier function-based quadratic program, 6, 63, 65, 68, 74, 94, 114

CLBF control Lyapunov-barrier function, 5–7, 94, 95, 99, 103, 105, 107, 110, 114, 117, 119, 120

CLF control Lyapunov function, 94, 103–105, 110, 114

LTL linear temporal logic, 1, 4–7, 9, 18, 19, 93–95, 97, 103, 110–112, 115, 117, 119, 120

MICP mixed-integer convex programming, 22, 24, 25, 35–38, 40, 56, 57, 59, 74, 75, 84, 86, 87, 93

MIP mixed-integer programming, 3, 118, 119

MPC model predictive control, 3, 6, 24, 36, 37, 41

NMPC nonlinear model predictive control, 63, 65, 67, 68, 74

PWL piecewise linear, 22, 36–38, 40

QP quadratic program, 94, 95, 112

ROA region of attraction, 96, 109

STL signal temporal logic, 1, 3–7, 9, 16–18, 22, 23, 25, 27, 35, 40–42, 48, 49, 56, 57, 59, 61, 63, 65, 71, 74–77, 84, 85, 92, 93, 118, 119

UAV unmanned aerial vehicle, 3, 5

List of Symbols

To facilitate readability and ensure consistent interpretation, the notation frequently used throughout this thesis is summarized in the list below.

$\mathbb{N}$	Set of positive integers
$\mathbb{N}_0$	Set of nonnegative integers
$\mathbb{R}$	Set of real numbers
$\mathbb{R}_+$	Set of nonnegative real numbers
$\mathbb{R}_{++}$	Set of positive real numbers
$\mathbb{T}$	Finite time interval over which the trajectory is defined, with $\mathbb{T} \subset \mathbb{R}_+$
$\mathbb{R}^n$	Standard Euclidean space of dimension $n > 1$
$\mathbb{R}^n_{++}$	Set of n-dimensional real vectors with strictly positive components
∂A	The boundary of the set A
$\mathrm{Int}(A)$	The interior of the set A
I_n	$n \times n$ identity matrix
$0_{n\times n}$	$n \times n$ zero matrix
$\mathbf{1}_n$	n-dimensional vector of ones
$\mathbb{S}^n_{++}$	Set of $n \times n$ symmetric positive-definite matrices
$\mathbb{S}^n_{+}$	Set of $n \times n$ symmetric positive-semidefinite matrices

$\mathrm{SO}(3)$	Special orthogonal group in three dimensions
$\mathfrak{so}(3)$	Lie algebra of $\mathrm{SO}(3)$
$\mathrm{SE}(3)$	Special Euclidean group in three dimensions
$\mathfrak{se}(3)$	Lie algebra of $\mathrm{SE}(3)$
$\|\cdot\|$	Euclidean norm
$\|\cdot\| \leq \cdot$	Componentwise absolute inequality; unless otherwise specified, for vectors $x, q \in \mathbb{R}^n$, $\|x\| \leq q$ means $\|x_j\| \leq q_j$ for $j = 1, \ldots, n$
$\odot$	The Hadamard product denoting componentwise multiplication, i.e., for vectors $a, b \in \mathbb{R}^n$, $(a \odot b)_i = a_i b_i$
$\lfloor\cdot\rfloor$	Floor function
$\lceil\cdot\rceil$	Ceiling function
AP	Finite set of atomic propositions
2^{AP}	The power set of the set AP
π	Atomic proposition in LTL; predicate in STL
$\top$	A true statement
$\bot$	A false statement
$\neg$	Negation
$\mathcal{U}$	The temporal operator denoting "until"
$\Diamond$	The derived temporal operator denoting "eventually", defined by $\Diamond\varphi \triangleq \top\mathcal{U}\varphi$
$\Box$	The derived temporal operator denoting "always", defined by $\Box\varphi \triangleq \neg\Diamond\neg\varphi$
$\models$	Satisfaction relation; $s \models \varphi$ means that the signal s satisfies φ
$\not\models$	Non-satisfaction relation; $s \not\models \varphi$ means that the signal s does not satisfy φ
$\bigcirc$	The temporal operator denoting "next"
$\mathrm{LTL}_{\setminus\bigcirc}$	Next-free linear temporal logic, which excludes the "next" operator

Chapter 1

Introduction

1.1 Motivation

Autonomous robotic systems are increasingly required to perform complex and safety-critical missions beyond classical objectives, such as stabilization and trajectory tracking. Taking quadrotors as an example, as illustrated in Figure 1.1, drones used in agriculture, forest monitoring, and pipeline inspection may need to repeatedly patrol regions of interest, whereas those used for healthcare delivery, smart-city services, and disaster response may be required to reach specific locations within prescribed time limits. These applications impose temporally structured mission requirements, such as always avoiding unsafe regions, visiting regions in a specified order, maintaining inter-agent separation, and repeatedly accomplishing subtasks over long horizons [66, 100].

The need to specify the mission requirements and verify whether robotic systems satisfy them motivates the use of formal temporal logics, which have been widely used in model checking to describe temporal properties of transition systems and computational models [23]. Signal temporal logic (STL) and linear temporal logic (LTL) are two widely used formal temporal logics. Both can specify mission requirements and support the synthesis of plans or controllers with formal satisfaction guarantees, but they differ in how system behaviors and time are represented. STL is interpreted over continuous-valued signals and allows explicit time intervals, making it particularly suitable for time-bounded requirements in cyber-physical and continuous-time systems [102, 9]. On the other hand, LTL is typically interpreted over sequences of Boolean propositions and is commonly used to express qualitative temporal relationships without explicit quantitative time bounds [65].

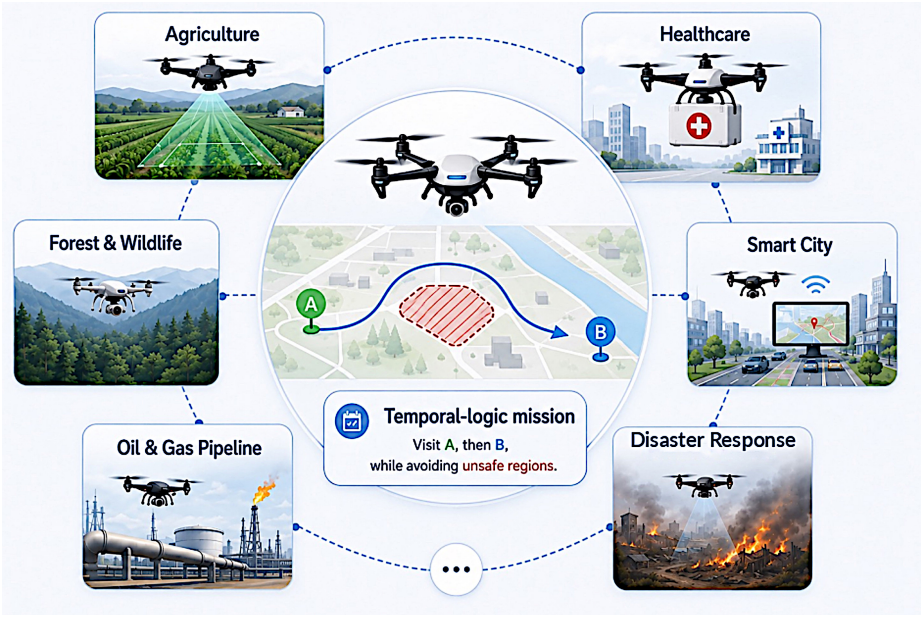


Figure 1.1: Quadrotor applications with temporal logic specifications.

To further ensure that autonomous systems can satisfy temporal logic specifications, it is necessary to address both what behaviors should be generated to meet the mission requirements and how these behaviors can be implemented on the underlying dynamical systems. From this perspective, temporal-logic-based planning and control for continuous-time systems can be viewed as two tightly coupled problems:

- **Planning:** constructing plans or trajectories that satisfy the logical structure of a mission while accounting for the constraints imposed by the system dynamics.
- **Control synthesis:** designing feedback control laws so that the resulting closed-loop trajectories implement the planned behaviors and satisfy the prescribed temporal logic specifications.

This distinction motivates the following discussion from two complementary perspectives: planning under temporal logic specifications and control synthesis under temporal logic specifications.

1.1.1 Planning under Temporal Logic

Planning under temporal logic specifications concerns the generation of feasible trajectories or high-level plans that satisfy a temporal logic formula describing the desired mission, while respecting system, environmental, and task constraints.

In planning problems, STL is particularly suitable for specifying time-bounded requirements over real-valued state or output trajectories [102, 9]. Such requirements include reaching a target set within a prescribed time interval, always avoiding unsafe regions, and maintaining inter-agent distances over a finite horizon. The Boolean semantics of STL determine whether a trajectory satisfies a given specification. To further quantify the degree of satisfaction or violation, quantitative robustness semantics have been introduced [38]. This robustness measure enables optimization-based planning formulations, where robustness can be maximized or imposed as a constraint [11].

Existing STL planning methods often formulate planning as trajectory optimization [11]. A foundational approach encodes STL specifications together with system constraints using mixed-integer programming (MIP), enabling model predictive control (MPC)-based synthesis and trajectory generation under temporal logic requirements [123]. Building on this idea, robust STL planning methods incorporate quantitative robustness into mixed-integer optimization formulations to improve the robustness of the planned trajectories [126, 149]. Nevertheless, optimization-based STL planning can suffer from high computational complexity, especially for long horizons or complex specifications. To improve scalability, a more compact MIP encoding for STL with fewer binary variables has been proposed [71]. Low-complexity MPC-based approaches, such as move-blocking strategies, have also been developed to reduce the computational burden of STL synthesis [149]. Another approach replaces the nonsmooth minimum and maximum operators in the STL robustness semantics with soft-minimum and soft-maximum approximations, yielding differentiable robustness measures that enable gradient-based optimization and improve computational efficiency [115, 50].

Moreover, a key challenge in continuous-time STL planning is to guarantee satisfaction over the entire continuous-time horizon, rather than only at the discrete time instants at which the constraints are enforced. One line of work addresses this issue by combining robustness margins with structured trajectory representations and bounds on the trajectory variation between sampling instants. For multi-drone missions, a sampling-based trajectory generation method maximizes smooth STL robustness and enforces a stricter version of the STL specification to guarantee continuous-time satisfaction [116]. This STL-based planning framework was further integrated with safety and stakeholder-specific operational requirements for missions involving multiple unmanned aerial vehicles

(UAVs) [117]. A planning and control co-design framework additionally incorporates a nonlinear tracking controller and its worst-case tracking-error bound into trajectory planning, ensuring that the tracked trajectories continue to satisfy the STL specification [118]. More recently, continuous-time trajectory generation under bounded STL specifications has been extended to nonholonomic ground robots using model-consistent parametric curves [64]. An alternative line of work uses control barrier functions (CBFs) to enforce continuous-time satisfaction over each sampling interval [87, 88, 148]. However, these approaches typically rely on the construction of suitable barrier functions, may require specific assumptions on the system dynamics and predicate structure, and can result in computationally demanding optimization problems.

Compared with time-bounded STL specifications, LTL is commonly used for high-level planning with proposition-based, long-horizon mission requirements. Classical LTL-based planning follows an automata-theoretic framework. The workspace or continuous-state system is abstracted into a finite transition system, the LTL formula is translated into an automaton, and a satisfying plan is obtained by searching for an accepting run on the product of the transition system and the automaton [63, 65].

For continuous-time robotic systems, a key issue of LTL planning is whether the symbolic plan obtained from the abstraction can be faithfully implemented by the original continuous dynamics. To reduce this abstraction-to-implementation gap, finite abstractions with robustness margins have been developed. Local reachable-set over-approximations can be used to compute such abstractions [83], and robustness margins can be incorporated into finite abstractions to account for inter-sample behaviors in temporal-logic-based control synthesis [96]. This line of work has also been extended to robustly complete control synthesis for nonlinear systems through interval-based branch-and-bound refinement [82].

Despite these developments, finite-abstraction-based LTL planning can remain computationally demanding for nonlinear, continuous-time robotic systems. Fine partitions are often needed to accurately capture the continuous dynamics, but they increase the size of the transition system and the product automaton. Moreover, a symbolic plan obtained on the abstraction still needs to be implemented by continuous-time controllers that drive the system between regions while maintaining safety and invariance properties.

To mitigate these limitations, specification-guided approaches have been developed to synthesize controllers for nonlinear systems with temporal logic specifications without explicitly constructing a global finite abstraction. In particular, such methods directly characterize the winning set over the continuous state space of the original system, guided by the automaton representation of the LTL specification [84]. Related abstraction-free approaches have also used CBFs to decompose temporal logic specifications into subtasks

and synthesize continuous-time controllers directly [113]. Such controller-based methods are therefore discussed further from the perspective of control synthesis under temporal logic specifications.

Overall, motivated by these challenges in continuous-time planning under temporal logic, this thesis focuses on the following two objectives. For STL, the goal is to generate continuous-time trajectories that satisfy time-bounded specifications without relying on small sampling periods or overly conservative robustness margins. For LTL, the goal is to synthesize long-horizon task plans that can be realized on continuous-time systems through abstraction-free methods.

1.1.2 Control Synthesis under Temporal Logic

A planned trajectory satisfying a temporal logic specification does not necessarily imply that the trajectory generated by the controlled system also satisfies the specification. Tracking errors, actuator limits, nonlinear dynamics, and imperfect initial conditions may cause the real system trajectory to deviate from the planned one, leading to violations of safety or temporal requirements during execution. This motivates control synthesis under temporal logic, where feedback laws are designed to implement planned behaviors while ensuring satisfaction of the prescribed specifications.

Because STL imposes time-bounded requirements on real-valued signals, tracking errors and disturbances may affect whether the controlled trajectory satisfies the specification. Motivated by this concern, prescribed performance control has been used to constrain the transient behavior of system trajectories to satisfy atomic STL specifications [91, 16]. However, prescribed-performance-based methods often rely on suitable performance functions for individual predicates or tasks, which may become restrictive when complex temporal requirements or safety-critical interactions are involved. CBFs have been used to encode temporal requirements and synthesize feedback controllers for STL tasks, thereby enforcing STL constraints during execution [86]. Decentralized CBFs further address multi-agent settings with coupling and communication constraints [87]. To handle specifications and systems with higher relative degree, high-order control Lyapunov-barrier functions (CLBFs) have been developed for temporal logic specifications [146]. Nevertheless, CBF-based formulations must ensure the feasibility of the resulting constraints, especially when multiple temporal, safety, and inter-agent requirements are imposed simultaneously. To connect trajectory generation more directly with controller capabilities, control and planning have been co-designed for multi-rotor UAVs by generating dynamically feasible trajectories that satisfy STL specifications and can be tracked by a controller with bounded

tracking error [118]. MPC-based methods have also been used to synthesize multi-agent behaviors under STL specifications, including multi-drone coordination tasks such as collision avoidance, inter-agent separation, and persistent surveillance [116]. These approaches explicitly connect planning, control, and coordination, but their applicability may depend on the chosen optimization formulations or communication structures.

On the other hand, control synthesis under LTL specifications aims to generate feedback controllers such that the resulting system trajectories satisfy long-horizon logical tasks. For mobile robots, temporal logic specifications over fragments of LTL have been translated into sequences of control barrier function-based quadratic program (CBF-QP) problems, with barrier composition and prioritization used to alleviate infeasibility [132]. Finite-time convergence CBFs have also been used to synthesize controllers for continuous-time multi-agent systems guided by LTL specifications [133]. CBFs have further been used for abstraction-free control synthesis under LTL constraints by constructing an accepting run and enforcing the resulting sequence of formulae through quadratic programs [113]. CLBF-based controllers have been used to combine stabilization and safety requirements for nonlinear systems [125]. However, the resulting controllers may become infeasible when multiple continuous constraints conflict or exceed the available control authority. Therefore, connecting continuous-control certificates with long-horizon LTL satisfaction remains a key challenge.

Overall, the central challenge is to connect high-level temporal logic specifications with nonlinear control synthesis for continuous-time robotic systems. This requires planning methods that generate satisfying trajectories or task plans, and controllers that preserve the desired logical requirements during execution. These challenges motivate the methods developed in this thesis, whose main contributions are summarized next.

1.2 Contributions and Thesis Organization

Motivated by the challenges discussed above, the main contributions of this thesis can be grouped into the following two parts.

- **Continuous-time STL planning and control synthesis.** This part develops planning and control synthesis methods for continuous-time systems under STL specifications, with an emphasis on continuous-time satisfaction and reduced conservatism. The proposed methods are demonstrated through quadrotor and multi-quadrotor control applications. This contribution is developed in Chapters 3–5.

- **Abstraction-free LTL control synthesis.** This part develops an abstraction-free control synthesis approach for continuous-time systems under LTL specifications. The approach uses patching CLBF level sets to connect high-level temporal logic requirements with continuous-time system dynamics. This contribution is developed in Chapter 6.

As shown in Figure 1.2, the thesis is organized as follows.

Chapter 2 presents the main preliminaries used throughout the thesis, including nonlinear system models, SE(3) geometry, Lyapunov stability, Bézier curves, and temporal logic. These tools provide the mathematical foundation for the planning, control, and stability analysis developed in subsequent chapters.

Chapter 3 develops an STL planning framework based on time-varying robustness, reducing the conservatism of conventional uniform robustness measures and enabling smooth continuous-time trajectory generation.

Chapter 4 presents an STL control synthesis framework for a quadrotor system by integrating geometric control on SE(3) with time-varying tracking-error bounds.

Chapter 5 extends the STL control synthesis framework to multiple quadrotors and incorporates multi-agent constraints into the specification-satisfaction analysis.

Chapter 6 develops an abstraction-free LTL control synthesis approach based on patching CLBF level sets.

Chapter 7 concludes the thesis and discusses future work.

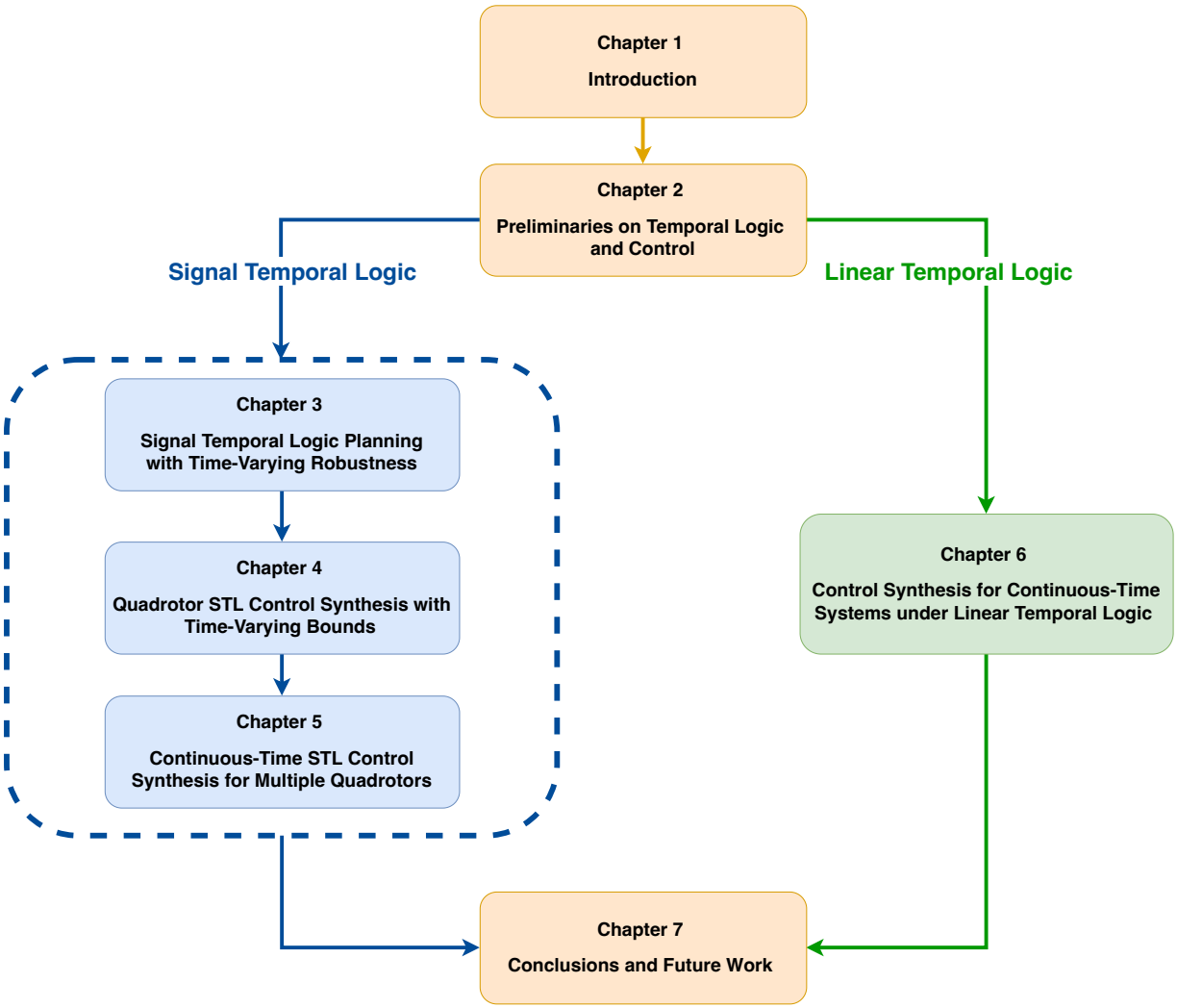


Figure 1.2: Organization of the thesis and relationships among the chapters.

Chapter 2

Preliminaries on Temporal Logic and Control

This chapter introduces the mathematical preliminaries and control concepts used throughout this thesis. It covers the system description, Lyapunov stability, the special Euclidean group SE(3) for rigid-body motion, and Bézier curves for trajectory representation. The chapter also reviews the basic definitions of STL and LTL used in subsequent chapters.

2.1 System Description

Consider a continuous-time dynamical system of the form

$$\dot{x}(t) = f(x(t), u(t)), \quad t \in \mathbb{R}_+, \tag{2.1.1}$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ denotes the system state and $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ denotes the control input at time t. Here, $\mathcal{X}$ and $\mathcal{U}$ denote the state space and the control input space, respectively, and $f : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$ is a smooth function that specifies the system dynamics. Given an initial condition $x(0) = x_0$, let $\mathbb{T} := [0, T]$, where $T \in \mathbb{R}_{++}$ is the time horizon of interest. We assume that, for every admissible control input $u : \mathbb{T} \to \mathcal{U}$ and initial condition considered in this thesis, the solution to (2.1.1) exists, is unique on $\mathbb{T}$, and remains in $\mathcal{X}$. The system evolution is then characterized by a state trajectory $x : \mathbb{T} \to \mathcal{X}$.

2.2 Lyapunov Stability

Lyapunov stability theory [62] provides a systematic method for analyzing the stability of dynamical systems without explicitly solving their trajectories. Consider the autonomous system

$$\dot{x} = f(x), \quad x(0) = x_0, \tag{2.2.1}$$

where $f : \mathcal{X} \to \mathbb{R}^n$ is locally Lipschitz on a domain $\mathcal{X} \subseteq \mathbb{R}^n$, and suppose that $x_{eq} \in \mathcal{X}$ is an equilibrium point satisfying

$$f(x_{eq}) = 0. \tag{2.2.2}$$

The behavior of the system near the equilibrium point x_{eq} can be characterized using the standard notions of Lyapunov stability, asymptotic stability, and exponential stability [62].

Definition 2.2.1 (Lyapunov Stability)**.** *The equilibrium point $x_{eq} \in \mathcal{X}$ of (2.2.1) is stable if, for every $\epsilon > 0$, there exists a $\delta > 0$ such that, for every initial condition $x(0) \in \mathcal{X}$,*

$$\|x(0) - x_{eq}\| < \delta \implies \|x(t) - x_{eq}\| < \epsilon, \quad \forall t \geq 0. \tag{2.2.3}$$

Definition 2.2.2 (Asymptotic Stability)**.** *The equilibrium point $x_{eq} \in \mathcal{X}$ of (2.2.1) is asymptotically stable if it is stable and there exists a constant $c > 0$ such that, for every initial condition $x(0) \in \mathcal{X}$,*

$$\|x(0) - x_{eq}\| < c \implies \lim_{t \to \infty} x(t) = x_{eq}. \tag{2.2.4}$$

Definition 2.2.3 (Exponential Stability)**.** *The equilibrium point $x_{eq} \in \mathcal{X}$ of (2.2.1) is exponentially stable if there exist positive constants k, λ, and c such that, for every initial condition $x(0) \in \mathcal{X}$,*

$$\|x(0) - x_{eq}\| < c \implies \|x(t) - x_{eq}\| \leq k\|x(0) - x_{eq}\|e^{-\lambda t}, \quad \forall t \geq 0. \tag{2.2.5}$$

A continuously differentiable function $\mathcal{V} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is called a Lyapunov function candidate with respect to the equilibrium point $x_{eq} \in \mathcal{X}$ if

$$\mathcal{V}(x_{eq}) = 0, \quad \mathcal{V}(x) > 0, \quad \forall x \in \mathcal{X} \setminus \{x_{eq}\}. \tag{2.2.6}$$

The derivative of $\mathcal{V}$ along the trajectories of (2.2.1) is given by

$$\dot{\mathcal{V}}(x) = \frac{\partial \mathcal{V}}{\partial x} f(x). \tag{2.2.7}$$

Definition 2.2.4 (Lyapunov Function [62, Chapter 4]). *A Lyapunov function candidate $\mathcal{V}$ is called a Lyapunov function for (2.2.1) if*

$$\dot{\mathcal{V}}(x) \leq 0, \qquad \forall x \in \mathcal{X}. \tag{2.2.8}$$

It is called a strict Lyapunov function if

$$\dot{\mathcal{V}}(x) < 0, \qquad \forall x \in \mathcal{X} \setminus \{x_{eq}\}. \tag{2.2.9}$$

Theorem 2.2.5 (Lyapunov Stability Theorem [62, Chapter 4]). *Let $x_{eq} \in \mathcal{X}$ be an equilibrium point of (2.2.1).*

i) If there exists a Lyapunov function $\mathcal{V}$ with respect to x_{eq}, then x_{eq} is stable.

ii) If there exists a strict Lyapunov function $\mathcal{V}$ with respect to x_{eq}, then x_{eq} is asymptotically stable.

iii) If there exist a Lyapunov function candidate $\mathcal{V}$ with respect to x_{eq} and positive constants c_1, c_2, and c_3 such that

$$c_1\|x - x_{eq}\|^2 \leq \mathcal{V}(x) \leq c_2\|x - x_{eq}\|^2, \qquad \forall x \in \mathcal{X}, \tag{2.2.10}$$

and

$$\dot{\mathcal{V}}(x) \leq -c_3\|x - x_{eq}\|^2, \qquad \forall x \in \mathcal{X}, \tag{2.2.11}$$

then x_{eq} is exponentially stable.

In Chapter 4, Lyapunov functions are constructed to establish stability guarantees for the closed-loop quadrotor system, whose configuration is represented on the special Euclidean group SE(3). The basic structure and properties of SE(3) are introduced in the following section.

2.3 The Special Euclidean Group SE(3)

The configuration of a rigid body in three-dimensional space is described by its position and orientation. The position evolves in the Euclidean space $\mathbb{R}^3$, while the orientation evolves on the nonlinear manifold SO(3). Together, they form the special Euclidean group SE(3), which provides a natural configuration space for rigid-body motion [111, 17, 78].

2.3.1 Definitions and Basic Properties

The special orthogonal group is defined as

$$\mathrm{SO}(3) = \left\{ R \in \mathbb{R}^{3\times 3} \mid R^\top R = \mathrm{I}_3,\ \det(R) = 1 \right\}, \tag{2.3.1}$$

where R is a rotation matrix. The condition $R^\top R = \mathrm{I}_3$ implies that the columns of R form an orthonormal basis, and the condition $\det(R) = 1$ ensures that the transformation preserves orientation. For any $R \in \mathrm{SO}(3)$, the inverse is given by

$$R^{-1} = R^\top. \tag{2.3.2}$$

The set $\mathrm{SO}(3)$ is a compact and connected matrix Lie group [53, 21].

The special Euclidean group is defined as

$$\mathrm{SE}(3) = \left\{ (R, p) \mid R \in \mathrm{SO}(3),\ p \in \mathbb{R}^3 \right\}, \tag{2.3.3}$$

where R represents the attitude of the rigid body and p represents its position.

2.3.2 Lie Algebra and Rigid-Body Kinematics

The Lie algebra of $\mathrm{SO}(3)$ is denoted by $\mathfrak{so}(3)$ and is given by the set of skew-symmetric matrices [101, Chapter 3],

$$\mathfrak{so}(3) = \left\{ A \in \mathbb{R}^{3\times 3} \mid A^\top = -A \right\}. \tag{2.3.4}$$

For $x = [x_1 \quad x_2 \quad x_3]^\top \in \mathbb{R}^3$, the hat map $(\cdot)^\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$ is defined as

$$\hat{x} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{2.3.5}$$

The inverse operation is called the vee map and is denoted by $(\cdot)^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ [111, 21].

Given a vector-valued function $x : \mathbb{R}_+ \to \mathbb{R}^3$, we use the shorthand

$$\hat{x}(t) := \widehat{x(t)}, \quad t \in \mathbb{R}_+. \tag{2.3.6}$$

The following identities of the hat and vee maps will be used in later geometric control analysis [111, 21, 78].

Lemma 2.3.1 (Properties of the hat and vee maps)**.** *For any $x, y \in \mathbb{R}^3$, $A \in \mathbb{R}^{3\times 3}$, and $R \in \mathrm{SO}(3)$, the following identities hold:*

$$\hat{x}^\top = -\hat{x}, \tag{2.3.7}$$

$$\hat{x}y = x \times y = -y \times x = -\hat{y}x, \tag{2.3.8}$$

$$\mathrm{tr}[A\hat{x}] = \frac{1}{2}\,\mathrm{tr}\left[\hat{x}(A - A^\top)\right] = -x^\top (A - A^\top)^\vee, \tag{2.3.9}$$

$$A\hat{x} + \hat{x}A^\top = \left(\left(\mathrm{tr}[A]\,\mathrm{I}_3 - A^\top\right)x\right)^\wedge, \tag{2.3.10}$$

$$R\hat{x}R^\top = \widehat{Rx}. \tag{2.3.11}$$

Since $\mathrm{SO}(3)$ is compact and connected, the exponential map from $\mathfrak{so}(3)$ to $\mathrm{SO}(3)$ is surjective [53]. Thus, for each $R \in \mathrm{SO}(3)$, there exists an $x \in \mathbb{R}^3$ such that

$$R = \exp(\hat{x}). \tag{2.3.12}$$

The exponential map admits the Euler–Rodrigues formula [46, 25]

$$\exp(\hat{x}) = \begin{cases} \mathrm{I}_3 + \dfrac{\sin(\|x\|)}{\|x\|}\hat{x} + \dfrac{1-\cos(\|x\|)}{\|x\|^2}\hat{x}^2, & x \neq 0, \\ \mathrm{I}_3, & x = 0. \end{cases} \tag{2.3.13}$$

These geometric properties will be used to formulate rigid-body dynamics and to design controllers directly on $\mathrm{SE}(3)$ or its subgroups. This avoids the singularities and ambiguities associated with local attitude parameterizations such as Euler angles [21, 78].

2.4 Bézier Curves

Bézier curves provide a convenient and compact parameterization for continuous trajectories. The material in this section follows the standard presentation in [42, Section 4]. We review only the properties needed for the subsequent developments and refer the reader to the cited reference for further details.

2.4.1 Bernstein Representation

A Bézier curve of degree $n \in \mathbb{N}$ is a mapping $r : [0,1] \to \mathbb{R}^d$ defined by

$$r(s) = \sum_{i=0}^{n} b_i^n(s) c_i, \qquad s \in [0,1], \tag{2.4.1}$$

where $c_i \in \mathbb{R}^d$, $i \in \{0, \ldots, n\}$, are the control points, and $b_i^n(s)$ denotes the i-th Bernstein basis polynomial of degree n, given by

$$b_i^n(s) = \binom{n}{i}(1-s)^{n-i}s^i. \tag{2.4.2}$$

Here, $\binom{n}{i} = \frac{n!}{(n-i)!i!}$ is the binomial coefficient. The parameter s is a normalized curve parameter. When the curve represents a trajectory over a finite time interval $[t_0, t_f]$ with $t_f > t_0$, it is parameterized by the normalized time $s = \frac{t-t_0}{t_f-t_0}$ for $t \in [t_0, t_f]$.

2.4.2 Geometric Properties

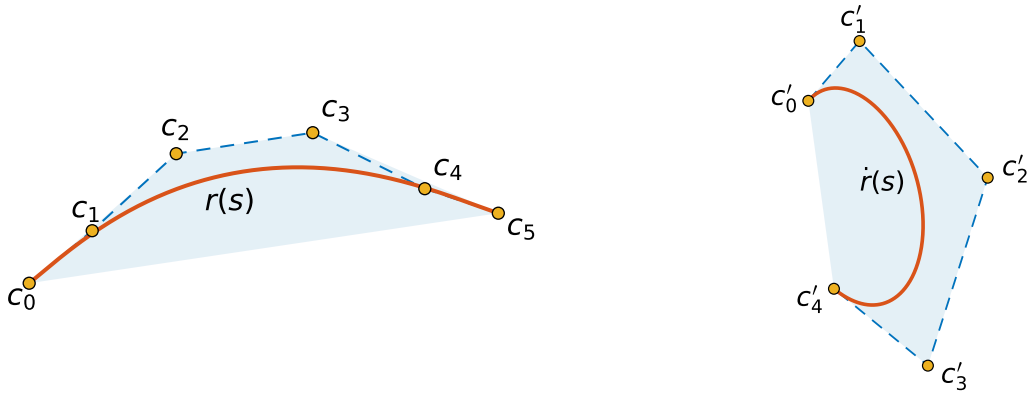
We recall several basic properties of Bézier curves that will be used in the subsequent development. These properties relate the geometry of the curve to its control points, making Bézier curves convenient for trajectory parameterization. They also enable constraints on the curve and its derivatives to be formulated through algebraic conditions on the control points.

Endpoint interpolation. A Bézier curve interpolates its first and last control points; that is, $r(0) = c_0$ and $r(1) = c_n$.

This follows from the endpoint values of the Bernstein basis functions: $b_0^n(0) = 1$, $b_n^n(1) = 1$, and all other basis functions vanish at the corresponding endpoints, as shown in Figure 2.1(a). In the trajectory setting, this property means that the first and last control points directly determine the initial and terminal positions.

Convex hull property. Let $\mathcal{CH} := \mathrm{conv}\{c_0, c_1, \ldots, c_n\}$ denote the convex hull of the control points, illustrated by the shaded region in Figure 2.1. Then, the image of the Bézier curve satisfies $r([0,1]) \subseteq \mathcal{CH}$.

This follows from $b_i^n(s) \geq 0$ and $\sum_{i=0}^n b_i^n(s) = 1$ for all $s \in [0,1]$. Hence, $r(s)$ is a convex combination of the control points. This property enables geometric constraints on the trajectory to be enforced conservatively through constraints on the control points or their convex hull.



(a) Bézier curve $r(s)$.

(b) First derivative curve $\dot{r}(s)$.

Figure 2.1: Illustration of a fifth-degree Bézier curve $r(s)$ and its first derivative curve $\dot{r}(s)$. The original curve is defined by six control points $\{c_0, \ldots, c_5\}$, while its derivative is a fourth-degree Bézier curve with control points $c'_i = 5(c_{i+1} - c_i)$, $i \in \{0, \ldots, 4\}$. The shaded regions indicate the convex hulls of the original and derivative control points, respectively.

Derivative property. The derivative of a degree-n Bézier curve with respect to s is a degree-$(n-1)$ Bézier curve:

$$\dot{r}(s) = n \sum_{i=0}^{n-1} b_i^{n-1}(s)(c_{i+1} - c_i). \tag{2.4.3}$$

Thus, the control points of $\dot{r}(s)$ are the scaled forward differences $n(c_{i+1} - c_i)$, $i = 0, \ldots, n-1$, as illustrated in Figure 2.1(b). This representation is useful for expressing velocity, acceleration, and smoothness constraints in terms of finite differences of the control points.

Extrapolation outside the normalized interval. For $s \notin [0, 1]$, the Bernstein basis functions are not necessarily nonnegative. Hence, $r(s) \notin \mathcal{CH}$ in general.

Therefore, the convex hull containment is only guaranteed on the normalized interval $[0, 1]$. This motivates the use of normalized local parameters when Bézier curves are used as trajectory segments over finite time intervals.

2.5 Signal Temporal Logic

Let $\mathbb{T} = [0, T]$ and let $y : \mathbb{T} \to \mathbb{R}^d$ denote a continuous-time planning trajectory of dimension d, over which STL formulas are evaluated. An atomic predicate is defined as $\pi := \mu(y(t)) \geq 0$, where $\mu : \mathbb{R}^d \to \mathbb{R}$ is a real-valued predicate function. Throughout this thesis, STL formulas are considered in positive normal form (PNF) [126]. The syntax of STL is defined recursively by

$$\varphi ::= \pi \mid \neg\pi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \mathcal{U}_{\mathcal{I}} \varphi_2, \tag{2.5.1}$$

where $\neg$, $\wedge$, $\vee$, and $\mathcal{U}$ are negation, conjunction, disjunction, and until operators, respectively, and $\mathcal{I} = [a, b]$ is a bounded time interval with $a < b$. The temporal operators *Eventually* ($\Diamond$) and *Always* ($\Box$) can be formulated by the above operators, $\Diamond_{\mathcal{I}} \varphi := \text{True}\, \mathcal{U}_{\mathcal{I}} \varphi$ and $\Box_{\mathcal{I}} \varphi := \neg\, (\Diamond_{\mathcal{I}} \neg \varphi)$.

Example 2.5.1. *The following examples illustrate the meanings of the basic temporal operators in STL.*

Always: $\Box_{[0,10]}(20 \leq T(t) \leq 24)$ *means that the office temperature always remains between* $20°\text{C}$ *and* $24°\text{C}$ *during office hours.*

Eventually: $\Diamond_{[0,5]}(T(t) < 25)$ *means that the temperature drops below* $25°\text{C}$ *within 5 time units.*

Until: *Let* μ_{heating} *denote that the kettle is heating. Then,* $\mu_{\text{heating}} \mathcal{U}_{[0,10]}\, (T(t) \geq 98°\text{C})$ *means that the kettle keeps heating until the water temperature reaches* $98°\text{C}$ *within 10 minutes.*

Furthermore, for any $t \in \mathbb{T}$, we denote by (y, t) the suffix of y starting at time t, i.e., the restriction of y to $[t, T] \cap \mathbb{T}$. More generally, for an interval $[a, b] \subseteq \mathbb{T}$, the notation $(y, [a, b])$ denotes the restriction of y to $[a, b]$. The notation $(y, t) \models \varphi$ denotes that the signal suffix (y, t) satisfies the STL formula φ, whereas $(y, t) \not\models \varphi$ denotes that it does not. For brevity, write $y \models \varphi$ when $(y, 0) \models \varphi$.

Definition 2.5.2 (STL Boolean semantics [102])**.** *Given a continuous-time signal* $y : \mathbb{T} \to \mathbb{R}^d$ *and an STL formula* φ*, the Boolean semantics of* φ *at time* $t \in \mathbb{T}$ *are defined inductively*

as follows:

$$
\begin{array}{lcl}
(y,t) \models \pi & \Leftrightarrow & \mu(y(t)) \geq 0; \\
(y,t) \models \neg\pi & \Leftrightarrow & \neg((y,t) \models \pi); \\
(y,t) \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & (y,t) \models \varphi_1 \wedge (y,t) \models \varphi_2; \\
(y,t) \models \varphi_1 \vee \varphi_2 & \Leftrightarrow & (y,t) \models \varphi_1 \vee (y,t) \models \varphi_2; \\
(y,t) \models \Diamond_{\mathcal{I}}\varphi & \Leftrightarrow & \exists t' \in \tilde{\mathcal{I}}, (y,t') \models \varphi; \\
(y,t) \models \Box_{\mathcal{I}}\varphi & \Leftrightarrow & \forall t' \in \tilde{\mathcal{I}}, (y,t') \models \varphi; \\
(y,t) \models \varphi_1 \mathcal{U}_{\mathcal{I}} \varphi_2 & \Leftrightarrow & \exists t' \in \tilde{\mathcal{I}}, (y,t') \models \varphi_2 \ \wedge \ \forall t'' \in [t,t'], (y,t'') \models \varphi_1,
\end{array}
$$

where $\tilde{\mathcal{I}} = (t + \mathcal{I}) \cap \mathbb{T}$ *and* $t + \mathcal{I} = [t + a, t + b]$.

In addition to boolean satisfaction, STL admits a quantitative semantics, often referred to as robustness. The robustness of a formula φ with respect to a signal y at time t is denoted by $\rho^{\varphi}(y,t) \in \mathbb{R}$. The magnitude of the robustness quantifies the degree of satisfaction or violation.

Definition 2.5.3 (STL robustness semantics [31])**.** *Given a continuous-time signal* $y : \mathbb{T} \to \mathbb{R}^d$ *and an STL formula* φ*, the robustness of* φ *with respect to* y *at time* $t \in \mathbb{T}$*, denoted by* $\rho^{\varphi}(y,t) \in \mathbb{R}$*, is defined inductively as follows:*

$$
\begin{array}{lcl}
\rho^{\pi}(y,t) & = & \mu(y(t)); \\
\rho^{\neg\pi}(y,t) & = & -\rho^{\pi}(y,t); \\
\rho^{\varphi_1 \wedge \varphi_2}(y,t) & = & \min\left\{\rho^{\varphi_1}(y,t), \rho^{\varphi_2}(y,t)\right\}; \\
\rho^{\varphi_1 \vee \varphi_2}(y,t) & = & \max\left\{\rho^{\varphi_1}(y,t), \rho^{\varphi_2}(y,t)\right\}; \\
\rho^{\Diamond_{\mathcal{I}}\varphi}(y,t) & = & \max_{t' \in \tilde{\mathcal{I}}} \rho^{\varphi}(y,t'); \\
\rho^{\Box_{\mathcal{I}}\varphi}(y,t) & = & \min_{t' \in \tilde{\mathcal{I}}} \rho^{\varphi}(y,t'); \\
\rho^{\varphi_1 \mathcal{U}_{\mathcal{I}} \varphi_2}(y,t) & = & \max_{t' \in \tilde{\mathcal{I}}} \min\left\{\rho^{\varphi_2}(y,t'), \min_{t'' \in [t,t']} \rho^{\varphi_1}(y,t'')\right\},
\end{array}
$$

Property 2.5.4 (Soundness and completeness of STL robustness semantics [37, 50])**.** *For any signal* $y : \mathbb{T} \to \mathbb{R}^d$ *and STL formula* φ*, nonzero robustness characterizes Boolean satisfaction at each time* $t \in \mathbb{T}$ *as follows:*

$$
\rho^{\varphi}(y,t) > 0 \quad \Longleftrightarrow \quad (y,t) \models \varphi. \tag{2.5.2}
$$

The forward implication expresses soundness, whereas the reverse implication expresses completeness. When $\rho^{\varphi}(y,t) = 0$*, the robustness value is inconclusive.*

Let $\mathcal{P}_\pi \subset \mathbb{R}^d$ be a nonempty closed satisfaction set associated with the predicate π, and let $\mathcal{P}_\pi^c := \mathbb{R}^d \setminus \mathcal{P}_\pi$ denote its complement. Define the predicate function by

$$\mu(x) := \begin{cases} \inf\limits_{z \in \mathcal{P}_\pi^c} \|x - z\|, & x \in \mathcal{P}_\pi, \\ -\inf\limits_{z \in \mathcal{P}_\pi} \|x - z\|, & x \notin \mathcal{P}_\pi, \end{cases} \tag{2.5.3}$$

Under this signed-distance convention, the robustness margin provides the following guarantee against trajectory perturbations.

Corollary 2.5.5 ([38]). *Let $y : \mathbb{T} \to \mathbb{R}^d$ be a trajectory and let φ be an STL formula such that $\rho^\varphi(y, 0) > 0$. Then any trajectory $\hat{y} : \mathbb{T} \to \mathbb{R}^d$ satisfying*

$$\|\hat{y}(t) - y(t)\| < \rho^\varphi(y, 0), \qquad \forall t \in [0, T], \tag{2.5.4}$$

also satisfies φ, i.e., $\hat{y} \models \varphi$.

2.6 Linear Temporal Logic

LTL is a temporal logic for specifying temporal properties over discrete sequences of symbolic observations and is typically used to describe qualitative ordering requirements over proposition-valued traces. Thus, while STL is more suitable for specifying finite-horizon temporal requirements, LTL is better suited for high-level task specifications over an infinite horizon, such as region-precedence constraints, persistent surveillance, and patrol objectives.

Let AP be a finite set of atomic propositions, where each $\pi \in AP$ is a Boolean statement evaluated at every discrete position of a behavior. The syntax of LTL formulas over AP is recursively defined by [8]

$$\varphi ::= \top \mid \pi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathcal{U} \varphi_2 \mid \varphi_1 \mathcal{R} \varphi_2,$$

where $\top$ denotes the Boolean constant true; $\neg$, $\wedge$, and $\vee$ denote negation, conjunction, and disjunction, respectively; and $\bigcirc$, $\mathcal{U}$, and $\mathcal{R}$ denote the next, until, and release operators, respectively. The eventually and always operators, denoted by $\Diamond$ and $\Box$, respectively, are derived as $\Diamond\varphi := \top\mathcal{U}\varphi$ and $\Box\varphi := \neg\Diamond\neg\varphi$. The release operator is the dual of until, defined by $\varphi_1 \mathcal{R} \varphi_2 := \neg\big(\neg\varphi_1 \mathcal{U} \neg\varphi_2\big)$.

The alphabet induced by AP is defined as $\Sigma = 2^{AP}$. An infinite word over Σ is written as $\sigma = \sigma_0\sigma_1\sigma_2 \cdots \in \Sigma^\omega$, where Σ^ω denotes the set of infinite sequences over Σ, and $i \in \mathbb{N}$ denotes a discrete position in the word. Each letter $\sigma_i \in \Sigma$ represents the set of atomic propositions that are true at position i.

Example 2.6.1. *Consider a simple printer system with* $AP = \{\mathsf{idle}, \mathsf{printing}, \mathsf{receivingJob}\}$. *An infinite word is given by*

$$\sigma = \sigma_0\sigma_1\sigma_2\cdots = \{\mathsf{idle}\}\{\mathsf{receivingJob}\}\{\mathsf{printing}, \mathsf{receivingJob}\}\{\mathsf{printing}\}\{\mathsf{idle}\}\cdots .$$

This word describes a recurring cycle in which the printer is idle, receives a job, begins printing while receiving another job, and then returns to the idle state.

Definition 2.6.2 (LTL word semantics [8])**.** *The notation* $\sigma, i \models \varphi$ *means that the formula* φ *holds on the word* σ *when evaluated from position* i*, that is, over the suffix* $\sigma_i\sigma_{i+1}\sigma_{i+2}\cdots$. *The satisfaction relation is defined inductively as follows:*

$$\begin{array}{lcl}
\sigma, i \models \top & \Leftrightarrow & \textit{True}; \\
\sigma, i \models \bot & \Leftrightarrow & \textit{False}; \\
\sigma, i \models \pi & \Leftrightarrow & \pi \in \sigma_i; \\
\sigma, i \models \neg\varphi & \Leftrightarrow & \neg(\sigma, i \models \varphi); \\
\sigma, i \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & \sigma, i \models \varphi_1 \textit{ and } \sigma, i \models \varphi_2; \\
\sigma, i \models \varphi_1 \vee \varphi_2 & \Leftrightarrow & \sigma, i \models \varphi_1 \textit{ or } \sigma, i \models \varphi_2; \\
\sigma, i \models \bigcirc\varphi & \Leftrightarrow & \sigma, i+1 \models \varphi; \\
\sigma, i \models \varphi_1 \mathcal{U} \varphi_2 & \Leftrightarrow & \exists j \geq i \textit{ such that } \sigma, j \models \varphi_2 \textit{ and } \sigma, k \models \varphi_1 \textit{ for all } k \textit{ such that } i \leq k < j; \\
\sigma, i \models \varphi_1 \mathcal{R} \varphi_2 & \Leftrightarrow & (\exists j \geq i \textit{ such that } \sigma, j \models \varphi_1 \textit{ and } \sigma, k \models \varphi_2 \textit{ for all } k \textit{ such that } i \leq k \leq j) \\
 & & \vee\, (\sigma, j \models \varphi_2 \textit{ for all } j \geq i)\,.
\end{array}$$

A word σ *is said to satisfy* φ*, denoted by* $\sigma \models \varphi$*, if* $\sigma, 0 \models \varphi$.

Examples of common LTL temporal operators are illustrated in Table 2.1.

Table 2.1: Representative satisfaction patterns for LTL temporal operators over the positions of an infinite word, adapted from [81].

Position i	0	1	2	3	4	5	$\cdots$
$\bigcirc\varphi$	$*$	φ	$*$	$*$	$*$	$*$	$\cdots$
$\varphi_1\mathcal{U}\varphi_2$	φ_1	φ_1	φ_2	$*$	$*$	$*$	$\cdots$
$\varphi_1\mathcal{R}\varphi_2$	φ_2	φ_2	$\varphi_1 \wedge \varphi_2$	$*$	$*$	$*$	$\cdots$
$\varphi_1\mathcal{R}\varphi_2$	φ_2	φ_2	φ_2	φ_2	φ_2	φ_2	$\cdots$
$\Box\varphi$	φ	φ	φ	φ	φ	φ	$\cdots$
$\Diamond\varphi$	$*$	$*$	φ	$*$	$*$	$*$	$\cdots$
$\Diamond\Box\varphi$	$*$	$*$	$*$	φ	φ	φ	$\cdots$
$\Box\Diamond\varphi$	$*$	φ	$*$	φ	$*$	φ	$\cdots$

$*$ denotes an arbitrary truth value.

Example 2.6.3. *In robot motion planning, let π_1 and π_2 denote the propositions indicating that the robot is in surveillance regions p_1 and p_2, respectively. Let* obs, low, *and* charge *denote the propositions indicating that the robot is in an obstacle region, has a low battery level, and is at a charging station, respectively. A specification requiring the robot to always avoid obstacles, repeatedly visit both surveillance regions, and eventually reach a charging station whenever the battery is low can be written as*

$$\varphi = \square\neg\mathsf{obs} \wedge \square\lozenge\pi_1 \wedge \square\lozenge\pi_2 \wedge \square\big(\mathsf{low} \rightarrow \lozenge\mathsf{charge}\big).$$

The word semantics in Definition 2.6.2 are formulated over symbolic observations rather than directly over the states of the dynamical system in (2.1.1). A labeling function is used to map a given state sequence to a symbolic trace.

Definition 2.6.4 (Symbolic trace of a state sequence)**.** *Let $x_0x_1x_2\cdots$ be an infinite state sequence in a state space $\mathcal{X}$, and let $L : \mathcal{X} \rightarrow 2^{AP}$ be a labeling function. The symbolic trace induced by the state sequence is the infinite word*

$$\mathrm{Trace}(x) := L(x_0)L(x_1)L(x_2)\cdots = \sigma_0\sigma_1\sigma_2\cdots \in (2^{AP})^{\omega}, \tag{2.6.1}$$

where $\sigma_i := L(x_i)$.

The mapping from a state sequence to a symbolic trace is illustrated in Figure 2.2. Definition 2.6.4 concerns a given discrete state sequence. Traces of continuous-time trajectories and the corresponding satisfaction relation are defined separately in Chapter 6.

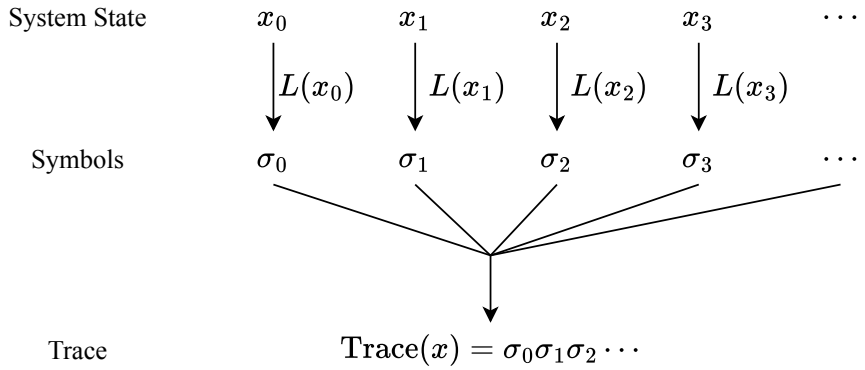


Figure 2.2: From system states to a symbolic word.

To provide a finite representation of an ultimately periodic infinite behavior, we introduce the notion of a lasso-form word.

Definition 2.6.5 (Lasso-form word [132])**.** *An infinite word $\sigma \in \Sigma^{\omega}$ is said to be in* lasso form *if there exist a finite prefix $\sigma_{\text{pre}} \in \Sigma^{*}$ and a nonempty finite suffix $\sigma_{\text{suf}} \in \Sigma^{+}$ such that*

$$\sigma = \sigma_{\text{pre}} \left(\sigma_{\text{suf}}\right)^{\omega}. \tag{2.6.2}$$

Here, Σ^{*} denotes the set of finite words over Σ, including the empty word, and Σ^{+} denotes the set of nonempty finite words over Σ.

Example 2.6.6. *For the previous printer system, the word*

$$\sigma = \{\mathsf{idle}\}\{\mathsf{receivingJob}\}\{\mathsf{printing}, \mathsf{receivingJob}\}\{\mathsf{printing}\}\{\mathsf{idle}\}\{\mathsf{receivingJob}\}\cdots$$

can be written in lasso form as

$$\sigma = \Big(\{\mathsf{idle}\}\{\mathsf{receivingJob}\}\{\mathsf{printing}, \mathsf{receivingJob}\}\{\mathsf{printing}\}\Big)^{\omega}.$$

Here, the prefix is empty.

2.7 Summary

This chapter introduced the notation and background used throughout the thesis. These preliminaries provide a common foundation for continuous-time system modeling, stability analysis, rigid-body geometry, trajectory representation, and temporal logic specifications used in the following chapters.

Chapter 3

Signal Temporal Logic Planning with Time-Varying Robustness

3.1 Introduction

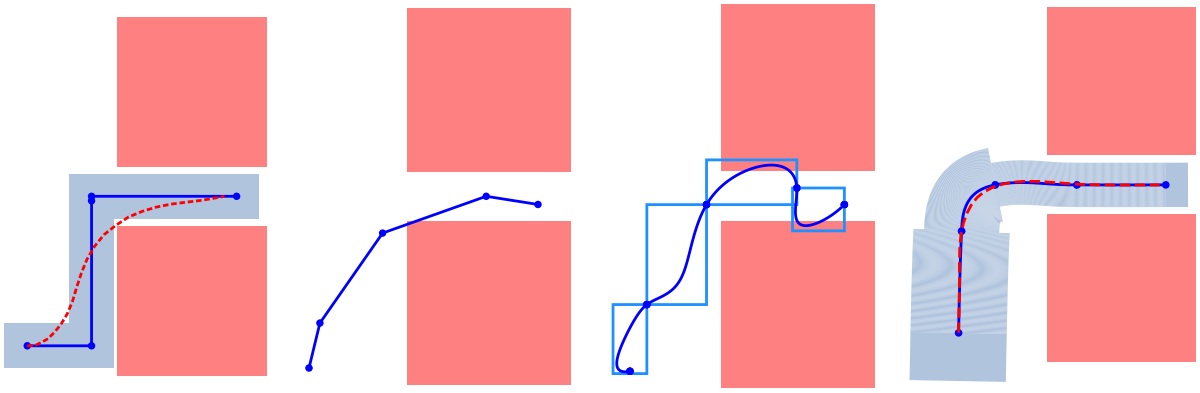
Planning trajectories that satisfy STL specifications in continuous time requires accounting for both logical satisfaction and kinematic feasibility, including velocity and acceleration constraints. As discussed in Section 2.5, a real-valued robustness measure provides a quantitative measure of specification satisfaction, with a larger positive value indicating stronger satisfaction.

However, because a uniform robustness margin must hold over the entire trajectory, it is limited by the most constrained portion and cannot capture the larger robustness margins available in regions with more free space.

Figure 3.1 illustrates these issues in a reach-avoid example: the piecewise linear (PWL) path [135] is conservative, the solution obtained from the discretized mixed-integer convex programming (MICP) formulation [11] may miss collision-avoidance violations, and a Bézier curve with improperly chosen control points may violate the specification. In contrast, the proposed time-varying robustness formulation generates a smoother trajectory with reduced conservatism while preserving continuous-time satisfaction.

Motivated by these observations, this chapter develops a continuous-time trajectory planning method for STL specifications using piecewise Bézier curves. The proposed method introduces a piecewise time-varying robustness formulation to reduce the conservatism of conventional real-valued robustness while preserving continuous-time satisfaction.

By generating smooth trajectories that account for kinematic feasibility, the approach improves trackability and helps maintain satisfaction of the STL specifications during tracking. Comparative experiments demonstrate the efficiency and effectiveness of the method. The source code is available at https://github.com/ViviaY/TimeVaryingBound_STL.



(a) PWL [135] (b) MICP [11] (c) I-Béziers (d) Proposed

Figure 3.1: Comparison of trajectory planning methods on a reach-avoid task: PWL [135], MICP [11], I-Béziers (Bézier curves with improperly chosen control points), and the proposed method. The solid blue lines denote the planned reference trajectories, and the blue circles denote the sampling points. The red dashed lines show the trajectories of a unicycle model controlled by MPC to track the corresponding reference trajectories. The blue shaded regions and blue frames represent the robustness tubes and bounding boxes, respectively. Panels (a)–(c) use uniform real-valued robustness, whereas panel (d) uses the proposed time-varying robustness formulation.

The two main contributions of this chapter are as follows.

- We extend uniform robustness to time-varying robustness for STL specifications, avoiding the conservatism caused by imposing a single worst-case robustness margin over the entire trajectory, as illustrated in Figure 3.1(a), and generating the smoother trajectory shown in Figure 3.1(d).
- We establish theoretical guarantees and enforce acceleration constraints through Bézier control points, thereby ensuring that the resulting Bézier curves satisfy the STL specifications and avoiding violations caused by improperly selected control points, as illustrated in Figure 3.1(c).

Moreover, the proposed approach (i) does not require an explicit system dynamics model in the planning formulation, unlike existing approaches based on MPC [149] and MICP [71]; (ii) generates kinematically feasible trajectories with smaller tracking errors compared to methods with variable sampling times [135]; (iii) addresses the problem of a trajectory traversing obstacles; and (iv) performs effectively in long-horizon scenarios.

3.2 Problem Statement

Let $T > 0$ denote the time horizon, and let $\mathbb{T} = [0, T] \subset \mathbb{R}_+$ be the bounded time domain. The continuous-time trajectory $y : \mathbb{T} \to \mathbb{R}^d$ is parameterized by N piecewise Bézier curves with sampling time $\Delta t = T/N$ and is defined as follows:

$$y(t) = \begin{cases} B_0(t), & t \in [t_0, t_1], \\ B_1(t), & t \in [t_1, t_2], \\ \vdots & \\ B_{N-1}(t), & t \in [t_{N-1}, t_N], \end{cases} \tag{3.2.1}$$

and each $B_k(t)$ is an n-degree Bézier curve defined as follows:

$$B_k(t) = \sum_{i=0}^{n} b_i^n(\tau) c_{k,i}, \ t \in [t_k, t_{k+1}], \tag{3.2.2}$$

where $c_{k,i}$ is the i-th control point of the k-th Bézier curve, $t_k = k\Delta t$ for $k \in \{0, \ldots, N-1\}$, $\tau = \frac{t - t_k}{t_{k+1} - t_k}$, and $b_i^n(\tau) = \frac{n!}{(n-i)!i!}(1-\tau)^{n-i}\tau^i$ is the n-degree Bernstein polynomial [42, Chapter 4.2]. Since $\sum_{i=0}^{n} b_i^n(\tau) = 1$ for any $\tau \in [0, 1]$, the convex hull of the set of control points, denoted by $\mathcal{CH}_k := \text{conv}\{c_{k,0}, c_{k,1}, \ldots, c_{k,n}\}$, contains B_k.

To determine whether a trajectory satisfies an STL specification φ with time-varying robustness, we propose the following definition of a time-varying robust trajectory.

Definition 3.2.1. *(Time-varying robust trajectory) Given a function $\rho^\varphi : \mathbb{T} \to \mathbb{R}_{++}$, a position trajectory $y : \mathbb{T} \to \mathbb{R}^d$ is said to be time-varying robust, denoted by ρ_t^φ-robust, with respect to an STL specification φ if every trajectory $y' : \mathbb{T} \to \mathbb{R}^d$ satisfying*

$$\|y'(t) - y(t)\| \leq \rho^\varphi(t), \quad \forall t \in \mathbb{T}, \tag{3.2.3}$$

also satisfies the STL specification φ.

Definition 3.2.2. *(STL planning problem based on Bézier curves) Given an STL specification φ, the problem is to find a position trajectory $y : \mathbb{T} \to \mathbb{R}^d$, parameterized by N piecewise n-degree Bézier curves as in (3.2.1), satisfying:*

i) Continuity conditions: The Bézier curves have to be at least C^2-continuous over $\mathbb{T}$;

ii) Kinematic conditions[1]: Given the componentwise maximum velocity and acceleration bounds $v_{\max}, a_{\max} \in \mathbb{R}^d$, all vector inequalities below are enforced componentwise. We require $|\dot{B}_k(t)| \leq v_{\max}$ and $|\ddot{B}_k(t)| \leq a_{\max}$ for all $k \in \{0, \dots, N-1\}$;

iii) STL Satisfaction: Given an STL specification φ and a desired minimum robustness $\rho_{\min}$, we require that the signal $y(t)$ be ρ_t^φ-robust, with its robustness $\rho^\varphi(t) = \rho_k^\varphi(r_k, \epsilon_k) := r_k - \epsilon_k \geq \rho_{\min} > 0$ for all $t \in [t_k, t_{k+1}]$. In the encoding presented in Section 3.4.1, the continuous variables $r_k, \epsilon_k \in \mathbb{R}_{++}$ and the binary variables $z_k^\varphi \in \{0, 1\}$ are used to encode the satisfaction of φ. Satisfaction of the complete formula at the initial index is enforced by setting the Boolean variable $z_0^\varphi = 1$. The corresponding constraints are expressed in terms of the maximum acceleration $a_k \in \mathbb{R}^d$ and the set of control points of B_k, denoted by $\mathcal{C}_k := \{c_{k,i}\}_{i=0}^n$.

We formulate the problem of finding a trajectory that meets the conditions in Definition 3.2.2 as a constrained optimization problem while maximizing time-varying robustness. We derive linear constraints for the condition i) in Section 3.3.1, and convex constraints for conditions ii) and iii) in Sections 3.3.2 and 3.4.1, respectively. This optimization problem is encoded as an MICP as follows:

$$\begin{aligned} \underset{\mathcal{C}_k, v_k, a_k, r_k, \epsilon_k, z_k^\varphi}{\arg\min} \; & \sum_{k=0}^{N-1} -\lambda \rho_k^\varphi(r_k, \epsilon_k) + Q \left\| v_k \right\|_1 + R \left\| a_k \right\|_1 \\ \text{s.t. } & B_k \text{ satisfies the conditions in Definition 3.2.2,} \\ & \forall k \in \{0, \dots, N-1\}, \end{aligned} \tag{3.2.4}$$

where $\lambda, Q, R > 0$ are the weights, and $v_k, a_k \in \mathbb{R}^d$ are the maximum velocity and acceleration of B_k, respectively. The program generates $y(t)$, as defined in (3.2.1), consisting of segments B_k obtained by solving (3.2.4), while robustly satisfying the specification with $\rho^\varphi(t)$. Theorem 3.4.2 establishes that $\rho^\varphi(t)$ is a robustness margin for the signal $y(t)$. Section 3.4.2 proves that the STL encoding in the proposed approach is sound.

[1]Following [117, 64], a trajectory satisfying the prescribed velocity and acceleration bounds is referred to as *kinematically feasible*. These bounds alone do not establish feasibility with respect to the full system dynamics and input constraints. The term *dynamic feasibility* is reserved for cases where the trajectory is shown to be realizable or trackable by the full dynamical model under admissible inputs, as considered in [118].

3.3 Trajectory Construction

In this section, leveraging the derivatives of the Bézier curves [42, Chapter 4.2], we derive constraints for the control points to ensure the continuity conditions and kinematic conditions outlined in Definition 3.2.2.

3.3.1 Continuity Constraints

In this work, since all Bézier segments have the same duration $\Delta t = T/N$.[2] The continuity conditions reduce to the following constraints.

i) *C^0-Continuity:* For $B_k(t) = B_{k+1}(t)$ at $t = t_{k+1}$, force

$$c_{k,n} = c_{k+1,0}. \tag{3.3.1}$$

ii) *C^1-Continuity:* For $\dot{B}_k(t) = \dot{B}_{k+1}(t)$ at $t = t_{k+1}$, force

$$c_{k,n} - c_{k,n-1} = c_{k+1,1} - c_{k+1,0}. \tag{3.3.2}$$

iii) *C^2-Continuity:* Let $\Delta^2 c_{k,i} := c_{k,i+2} - 2c_{k,i+1} + c_{k,i}$. To ensure $\ddot{B}_k(t) = \ddot{B}_{k+1}(t)$ at $t = t_{k+1}$, we enforce

$$\Delta^2 c_{k,n-2} = \Delta^2 c_{k+1,0}. \tag{3.3.3}$$

3.3.2 Kinematic Constraints

In this work, all inequalities involving vector-valued quantities are interpreted componentwise unless otherwise stated. For example, for $x, y \in \mathbb{R}^d$, the inequality $|x| \leq y$ means $|x_j| \leq y_j$ for all $j = 1, \ldots, d$. In this section, the Bézier curves are required to satisfy the convex constraints $|\dot{y}(t)| \leq v_{\max}$ and $|\ddot{y}(t)| \leq a_{\max}$, where $v_{\max}, a_{\max} \in \mathbb{R}^d_{++}$ represent the maximum velocity and acceleration over the whole time horizon, respectively.

[2] Different time intervals can be assigned to individual Bézier segments. In this work, however, introducing time variables into the acceleration constraints would result in bilinear constraints and increase the computational burden.

- *Velocity Constraints:* For $t \in [t_k, t_{k+1}]$, it follows that

$$\begin{aligned}
|\dot{y}(t)| &= |\dot{B}_k(t)| \\
&= \frac{n}{\Delta t} \left| \sum_{i=0}^{n-1} b_i^{n-1}(\tau)(c_{k,i+1} - c_{k,i}) \right| \\
&\leq \frac{n}{\Delta t} \sum_{i=0}^{n-1} b_i^{n-1}(\tau) |c_{k,i+1} - c_{k,i}| \\
&\leq \frac{n}{\Delta t} \left(\sum_{i=0}^{n-1} b_i^{n-1}(\tau) \right) \max_i |c_{k,i+1} - c_{k,i}| \\
&= \frac{n}{\Delta t} \max_i |c_{k,i+1} - c_{k,i}| \\
&\leq v_k.
\end{aligned}$$

  Since $b_i^{n-1}(\tau) \geq 0$ and $\sum_{i=0}^{n-1} b_i^{n-1}(\tau) = 1$ for any $\tau \in [0, 1]$, the velocity constraints can be enforced by the convex constraints:

$$|c_{k,i+1} - c_{k,i}| \leq \frac{v_k \Delta t}{n}, \forall i \in \{0, \ldots, n-1\}, \tag{3.3.4}$$

$$0 < v_k \leq v_{\max}. \tag{3.3.5}$$

- *Acceleration Constraints:* Similarly, the acceleration constraints of the Bézier curves are enforced as

$$|\Delta^2 c_{k,i}| = |c_{k,i+2} - 2c_{k,i+1} + c_{k,i}| \leq \frac{a_k \Delta t^2}{n(n-1)}, \quad \forall i \in \{0, \ldots, n-2\}, \tag{3.3.6}$$

$$0 < a_k \leq a_{\max}. \tag{3.3.7}$$

3.4 Encoding STL Satisfaction with Bézier Curves

In this section, we discuss how to encode an STL specification φ such that the Bézier curves satisfy the specification with time-varying robustness. We consider STL constraints defined over convex polytopic regions, where a convex polytope is denoted by $\text{Poly}(H, b) := \{x \in \mathbb{R}^d \mid Hx \leq b\} \subseteq \mathbb{R}^d$, with $H \in \mathbb{R}^{r \times d}$ and $b \in \mathbb{R}^r$. We use H_i to denote the i-th row of H and b_i to denote the i-th element of the vector b. Denote $\text{Row}(H)$ as the number of rows of H, corresponding to the number of defining half-spaces.

3.4.1 Encoding STL Satisfaction with Convex Constraints

For each Bézier segment B_k, defined over the time interval $[t_k, t_{k+1}]$, and each STL formula φ, a binary variable $z_k^\varphi \in \{0,1\}$ is introduced. The assignment $z_k^\varphi = 1$ serves as a sufficient certificate that φ is satisfied continuously over the corresponding segment, i.e., $(y,t) \models \varphi$, for all $t \in [t_k, t_{k+1}]$.

We begin with the atomic predicate π and its negation $\neg\pi$, which require B_k to lie entirely inside and outside $\mathrm{Poly}(H,b)$, respectively. As defined in Section 3.2, the segment-wise margin associated with φ over segment B_k is $\rho_k^\varphi(r_k, \epsilon_k) := r_k - \epsilon_k$. The corresponding encodings incorporating r_k and ϵ_k are presented below.

- Predicate π:

$$\begin{aligned} z_k^\pi = & \bigwedge_{j=1}^{\mathrm{Row}(H)} \left(\bigwedge_{i\in\{0,n\}} \frac{b_j - H_j c_{k,i}}{\|H_j\|} - r_k \geq 0 \right) \\ & \wedge \left(\bigwedge_{i\in\{1,\dots,n\}} |c_{k,i} - c_{k,i-1}| \leq \frac{a_k \Delta t^2}{2n} \right) \\ & \wedge \left(\bigwedge_{j'=1}^{d} \frac{8\epsilon_k}{3\sqrt{d}\Delta t^2} - |a_{k,j'}| \geq 0 \right). \end{aligned} \tag{3.4.1}$$

  Here, $a_{k,j'}$ is the j'-th element of $a_k \in \mathbb{R}^d$.

- Negation $\neg\pi$:

$$\begin{aligned} z_k^{\neg\pi} = & \bigvee_{j=1}^{\mathrm{Row}(H)} \left(\bigwedge_{i\in\{0,n\}} \frac{H_j c_{k,i} - b_j}{\|H_j\|} - r_k \geq 0 \right) \\ & \wedge \left(\bigwedge_{i\in\{1,\dots,n\}} |c_{k,i} - c_{k,i-1}| \leq \frac{a_k \Delta t^2}{2n} \right) \\ & \wedge \left(\bigwedge_{j'=1}^{d} \frac{8\epsilon_k}{3\sqrt{d}\Delta t^2} - |a_{k,j'}| \geq 0 \right). \end{aligned} \tag{3.4.2}$$

Note that the satisfaction set of the predicate π is the convex polytope $\mathrm{Poly}(H,b)$. Moreover, the continuous inequalities in (3.4.1), which consist of affine inequalities and convex

absolute-value bounds, define a convex feasible set. For the negation $\neg\pi$ in (3.4.2), the disjunction over j generally makes the overall feasible set nonconvex, whereas the continuous inequalities associated with each fixed j define a convex feasible set.

Based on the encodings of the atomic predicate and its negation, the Boolean variables associated with the logical and temporal operators are constructed recursively. For each temporal interval $\mathcal{I} = [a, b]$, we assume that its endpoints are aligned with the Bézier-segment partition, i.e., $\frac{a}{\Delta t} \in \mathbb{N}_0$ and $\frac{b}{\Delta t} \in \mathbb{N}_0$. For any $t \in [t_k, t_{k+1}]$, let $\widetilde{\mathcal{I}} = (t + \mathcal{I}) \cap \mathbb{T}$ denote the corresponding truncated continuous-time interval. The lower and upper segment indices are defined by $k_l = k + \left\lfloor \frac{a}{\Delta t} \right\rfloor$ and $k_r = \min\left(k + \left\lfloor \frac{b}{\Delta t} \right\rfloor, N - 1\right)$, with $k_l \leq k_r$.

Since each variable z_k^{φ} certifies continuous satisfaction of φ over the entire time interval $[t_k, t_{k+1}]$, the Boolean and temporal operators are encoded recursively as follows. The conjunction and disjunction operators are evaluated on the same Bézier segment. The always operator requires its subformula to hold over every segment indexed from k_l to k_r, whereas the eventually operator requires its subformula to hold over at least one segment in this range. The until operator requires a witness segment $k' \in \{k_l, \ldots, k_r\}$ on which φ_2 holds and requires φ_1 to hold over every segment from k through k', inclusive.

- Conjunction $\varphi_1 \wedge \varphi_2$: $$z_k^{\varphi_1 \wedge \varphi_2} = z_k^{\varphi_1} \wedge z_k^{\varphi_2}, \tag{3.4.3}$$
- Disjunction $\varphi_1 \vee \varphi_2$: $$z_k^{\varphi_1 \vee \varphi_2} = z_k^{\varphi_1} \vee z_k^{\varphi_2}, \tag{3.4.4}$$
- Always $\square_{\mathcal{I}}\varphi$: $$z_k^{\square_{\mathcal{I}}\varphi} = \bigwedge_{k'=k_l}^{k_r} z_{k'}^{\varphi}, \tag{3.4.5}$$
- Eventually $\Diamond_{\mathcal{I}}\varphi$: $$z_k^{\Diamond_{\mathcal{I}}\varphi} = \bigvee_{k'=k_l}^{k_r} z_{k'}^{\varphi}, \tag{3.4.6}$$
- Until $\varphi_1 \mathcal{U}_{\mathcal{I}} \varphi_2$: $$z_k^{\varphi_1 \mathcal{U}_{\mathcal{I}} \varphi_2} = \bigvee_{k'=k_l}^{k_r} \left(z_{k'}^{\varphi_2} \wedge \bigwedge_{k''=k}^{k'} z_{k''}^{\varphi_1} \right). \tag{3.4.7}$$

To force $z_k^{\varphi} = 1$, we utilize the Big-M method. As in [71, 135], with m constraints LC_i, binary variables z_i and a sufficiently large constant $M > 0$ are introduced to convert disjunctions $\bigvee_{i=1}^{n} \mathrm{LC}_i \geq 0$ into equivalent conjunctions

$$\bigwedge_{i=1}^{m} \mathrm{LC}_i + (1 - z_i)\, M \geq 0 \wedge \sum_{i=1}^{m} z_i \geq 1. \tag{3.4.8}$$

Thus, when $z_i = 1$, $\mathrm{LC}_i \geq 0$ holds; otherwise, $\mathrm{LC}_i + M \geq 0$ is always true. $\sum_{i=1}^{m} z_i \geq 1$ ensures that at least one $z_i = 1$. Ultimately, all conjunctions are encoded without any z_i by directly enforcing $\mathrm{LC}_i \geq 0$.

3.4.2 Soundness of STL Encoding

In this work, following [71], an STL encoding is said to be sound if every feasible solution of the encoding satisfies the corresponding STL specification under the Boolean semantics, and complete if the encoding admits a feasible solution whenever a signal satisfying the specification exists. This subsection establishes the soundness of the proposed encoding.

Soundness is established recursively over the structure of the STL formula, specifically, Proposition 3.4.1 establishes the atomic predicate and its negation as the base cases, and Theorem 3.4.2 extends soundness to the Boolean and temporal operators by structural induction. The key idea of Proposition 3.4.1 is to combine the endpoint and acceleration constraints. For the predicate, the encoding constrains both end control points to lie inside the polytope with a margin of at least r_k. For the negated predicate, the encoding requires both end control points to lie in a common exterior half-space with a margin of at least r_k from its defining hyperplane. The acceleration constraints then ensure that each intermediate control point lies within ϵ_k of one of the end control points. Hence, all control points satisfy the corresponding interior or exterior condition with a residual margin of at least $r_k - \epsilon_k$. Since the corresponding tightened polytope or exterior half-space is convex, the convex-hull property of Bézier curves extends this guarantee to the entire segment B_k.

Proposition 3.4.1. *For each segment k, let B_k be an n-degree Bézier curve defined on $[t_k, t_{k+1}]$. Suppose that the acceleration constraint (3.3.6) holds and that $r_k - \epsilon_k \geq \rho_{\min} > 0$. If the encoding (3.4.1) holds for B_k, then $B_k \subseteq \mathrm{Poly}(H, b)$. Similarly, if the encoding (3.4.2) holds for B_k, then $B_k \cap \mathrm{Poly}(H, b) = \emptyset$.*

Proof. We first need to bound the distance from the other control points to the corresponding end control point. Note that from (3.4.1) we have that $\|c_{k,1} - c_{k,0}\| \leq \frac{\|a_k\|\Delta t^2}{2n}$ and $\|c_{k,n-1} - c_{k,n}\| \leq \frac{\|a_k\|\Delta t^2}{2n}$. The acceleration constraint (3.3.6) on the control points implies that

$$\|c_{k,i+2} - 2c_{k,i+1} + c_{k,i}\| \leq \frac{\|a_k\|\Delta t^2}{n(n-1)}. \tag{3.4.9}$$

Now suppose that $2 \leq i \leq \lfloor \frac{n}{2} \rfloor$ as the case when $\lfloor \frac{n}{2} \rfloor + 1 \leq i \leq n-2$ is similar. By repeated application of the triangle inequality, it can be observed that

$$\begin{aligned}\|c_{k,i} - c_{k,0}\| \leq & \sum_{i'=0}^{i-2} (i'+1)\|c_{k,i-i'} - 2c_{k,i-i'-1} + c_{k,i-i'-2}\| + i\|c_{k,1} - c_{k,0}\| \\ \leq & \sum_{i'=0}^{i-2} (i'+1)\frac{\|a_k\|\Delta t^2}{n(n-1)} + i\|c_{k,1} - c_{k,0}\| \\ \leq & \frac{i(i-1)\|a_k\|\Delta t^2}{2n(n-1)} + i\frac{\|a_k\|\Delta t^2}{2n}.\end{aligned} \tag{3.4.10}$$

Since $i \leq n/2$, we have $\frac{i}{2n} \leq \frac{1}{4}$ and

$$\frac{i(i-1)}{2n(n-1)} \leq \frac{\frac{n}{2}(\frac{n}{2}-1)}{2n(n-1)} = \frac{n-2}{8(n-1)} \leq \frac{1}{8}, \tag{3.4.11}$$

which implies that

$$\|c_{k,i} - c_{k,0}\| \leq \left(\frac{1}{8} + \frac{1}{4}\right)\|a_k\|\Delta t^2 \leq \frac{3\Delta t^2\|a_k\|}{8}. \tag{3.4.12}$$

Therefore, for $2 \leq i \leq n-2$, we have the following bound

$$\min\{\|c_{k,i} - c_{k,0}\|, \|c_{k,i} - c_{k,n}\|\} \leq \frac{3\Delta t^2\|a_k\|}{8}. \tag{3.4.13}$$

Now suppose that (3.4.1) holds. Denote the set obtained by shrinking $\mathrm{Poly}(H, b)$ by some $c > 0$ as

$$\mathrm{P}_c := \{q \in \mathbb{R}^d \mid \bigwedge_{j=1}^{\mathrm{Row}(H)} b_j - H_j \cdot q - c\,\|H_j\| \geq 0\}.$$

Thus, the constraints (3.4.1) implies that $\frac{b_j - H_j c_{k,0}}{\|H_j\|} - r_k \geq 0$ and $\frac{b_j - H_j c_{k,n}}{\|H_j\|} - r_k \geq 0$ for all $j \in \{1, \ldots, \mathrm{Row}(H)\}$, which ensures both $c_{k,0}$ and $c_{k,n}$ are in P_{r_k}. As $\frac{8\epsilon_k}{3\sqrt{d}\Delta t^2} - |a_{k,j'}| \geq 0$ holds for all $j' \in \{1, \ldots, d\}$, this implies that $\frac{3\Delta t^2\|a_k\|}{8} \leq \epsilon_k$. Since we force $\rho_k^{\varphi}(\mathcal{C}_k, a_k) = r_k - \epsilon_k \geq \rho_{\min}$, combining the inequalities gives

$$\min\{\|c_{k,i} - c_{k,0}\|, \|c_{k,i} - c_{k,n}\|\} \leq \epsilon_k < r_k, \tag{3.4.14}$$

for all $i \in \{1, \dots, n-1\}$.

Then, we prove that all control points of B_k lie in the tightened polytope $P_{r_k - \epsilon_k}$. Define the closed ϵ_k-neighborhood of the line segment joining $c_{k,0}$ and $c_{k,n}$ as

$$\mathcal{LS}_k := \left\{ (1-\lambda)c_{k,0} + \lambda c_{k,n} + \delta \;\middle|\; \lambda \in [0,1], \delta \in \mathbb{R}^d,\ \|\delta\| \le \epsilon_k \right\}. \tag{3.4.15}$$

As λ ranges over $[0,1]$, the point $(1-\lambda)c_{k,0} + \lambda c_{k,n}$ traces the line segment connecting $c_{k,0}$ and $c_{k,n}$, while δ ranges over the closed Euclidean ball of radius ϵ_k centered at the origin. Thus, $\mathcal{LS}_k$ is the sum of these two convex sets and is therefore convex.

For each interior control point $c_{k,i}$, $i \in \{1, \dots, n-1\}$, (3.4.14) ensures that there exists $\lambda_i \in \{0,1\}$ such that

$$\left\| c_{k,i} - \left((1-\lambda_i)c_{k,0} + \lambda_i c_{k,n} \right) \right\| \le \epsilon_k, \tag{3.4.16}$$

Setting $\delta_i = c_{k,i} - \left((1-\lambda_i)c_{k,0} + \lambda_i c_{k,n} \right)$ gives $c_{k,i} = (1-\lambda_i)c_{k,0} + \lambda_i c_{k,n} + \delta_i$ with $\|\delta_i\| \le \epsilon_k$. Hence, $c_{k,i} \in \mathcal{LS}_k$. This implies that $B_k \subseteq \operatorname{conv}\{c_{k,0}, \dots, c_{k,n}\} \subseteq \mathcal{LS}_k$.

For any $q \in \mathcal{LS}_k$, there exist $\lambda \in [0,1]$ and $\delta \in \mathbb{R}^d$ such that $q = p + \delta$ where $p = (1-\lambda)c_{k,0} + \lambda c_{k,n}$ and $\|\delta\| \le \epsilon_k$. From the predicate constraint in (3.4.1), we have $c_{k,0}, c_{k,n} \in P_{r_k}$, which implies, for every $j \in \{1, \dots, \mathrm{Row}(H)\}$, $b_j - H_j c_{k,0} \ge r_k \|H_j\|$ and $b_j - H_j c_{k,n} \ge r_k \|H_j\|$. Hence,

$$\begin{aligned} b_j - H_j p &= (1-\lambda)(b_j - H_j c_{k,0}) + \lambda (b_j - H_j c_{k,n}) \\ &\ge r_k \|H_j\|. \end{aligned} \tag{3.4.17}$$

Using $q = p + \delta$ and the Cauchy–Schwarz inequality gives

$$\begin{aligned} b_j - H_j q &= b_j - H_j p - H_j \delta \\ &\ge b_j - H_j p - \|H_j\| \|\delta\| \\ &\ge (r_k - \epsilon_k)\|H_j\|. \end{aligned} \tag{3.4.18}$$

Thus, $q \in P_{r_k - \epsilon_k}$. Since $q \in \mathcal{LS}_k$ was arbitrary, $\mathcal{LS}_k \subseteq P_{r_k - \epsilon_k}$.

Combining the preceding inclusions gives $B_k \subseteq \mathcal{LS}_k \subseteq P_{r_k - \epsilon_k} \subseteq \mathrm{Poly}(H, b)$. Therefore, $B_k \subseteq \mathrm{Poly}(H, b)$. Moreover, the inclusion $B_k \subseteq P_{r_k - \epsilon_k}$ shows that the predicate holds throughout $[t_k, t_{k+1}]$ with a robustness margin of at least $\rho_k^{\varphi}(r_k, \epsilon_k) = r_k - \epsilon_k$.

For the second part, suppose that (3.4.2) holds and define

$$\mathrm{HS}_c^j := \left\{ q \in \mathbb{R}^d \;\middle|\; H_j q - b_j \ge c \|H_j\| \right\}. \tag{3.4.19}$$

The disjunction in (3.4.2) selects an index $j \in \{1, \ldots, \mathrm{Row}(H)\}$ such that $c_{k,0}, c_{k,n} \in \mathrm{HS}^j_{r_k}$. Consider any $q \in \mathcal{LS}_k$, then, we have $q = p + \delta$ and $p = (1-\lambda)c_{k,0} + \lambda c_{k,n}$ with $\|\delta\| \leq \epsilon_k$. Since both endpoints belong to $\mathrm{HS}^j_{r_k}$,

$$\begin{aligned} H_j p - b_j &= (1-\lambda)(H_j c_{k,0} - b_j) + \lambda(H_j c_{k,n} - b_j) \\ &\geq r_k \|H_j\|. \end{aligned} \tag{3.4.20}$$

Consequently,

$$\begin{aligned} H_j q - b_j &= H_j p - b_j + H_j \delta \\ &\geq H_j p - b_j - \|H_j\|\|\delta\| \\ &\geq (r_k - \epsilon_k)\|H_j\|. \end{aligned} \tag{3.4.21}$$

Since $q \in \mathcal{LS}_k$ was arbitrary, $\mathcal{LS}_k \subseteq \mathrm{HS}^j_{r_k - \epsilon_k}$. Together with $B_k \subseteq \mathcal{LS}_k$, this gives $B_k \subseteq \mathrm{HS}^j_{r_k-\epsilon_k}$. Since $r_k - \epsilon_k \geq \rho_{\min} > 0$, $H_j B_k(t) > b_j$, for all $t \in [t_k, t_{k+1}]$. Therefore, $B_k \cap \mathrm{Poly}(H, b) = \emptyset$.

□

Furthermore, the subsequent result establishes the soundness property for both the logical and temporal operations applied to the atomic predicates.

Theorem 3.4.2. *Consider an STL specification φ, a desired trajectory y_d constructed by N piecewise Bézier curves with $y_d(t) = B_k(t)$ for all $t \in [t_k, t_{k+1}]$, and the Boolean encoding of φ given by (3.4.1)–(3.4.8), where z^φ_k denotes the Boolean variable associated with φ on segment $[t_k, t_{k+1}]$. For any position trajectory y satisfying $\|y(t) - B_k(t)\| \leq \rho^\varphi_k(r_k, \epsilon_k)$, $\forall t \in [t_k, t_{k+1}]$ and $\forall k \in \{0, \ldots, N-1\}$, if z^φ_k is true, then y satisfies φ over the segment $[t_k, t_{k+1}]$.*

Proof. We prove the result by structural induction on the formula. Specifically, for every subformula of φ and every segment index k, we show that if its corresponding Boolean variable is equal to one, then the subformula is satisfied at every $t \in [t_k, t_{k+1}]$. For the atomic predicate π, the proof of Proposition 3.4.1 establishes that the predicate encoding implies

$$b_j - H_j B_k(t) \geq (r_k - \epsilon_k)\|H_j\|, \qquad \forall j \in \{1, \ldots, \mathrm{Row}(H)\}, \quad \forall t \in [t_k, t_{k+1}]. \tag{3.4.22}$$

Since $\rho^\varphi_k(r_k, \epsilon_k) = r_k - \epsilon_k$, the assumed perturbation bound and the Cauchy–Schwarz inequality give

$$\begin{aligned} b_j - H_j y(t) &= b_j - H_j B_k(t) - H_j\big(y(t) - B_k(t)\big) \\ &\geq (r_k - \epsilon_k)\|H_j\| - \|H_j\|\|y(t) - B_k(t)\| \\ &\geq 0. \end{aligned} \tag{3.4.23}$$

Therefore, $y(t) \in \mathrm{Poly}(H, b)$ and $(y, t) \models \pi$, $\forall t \in [t_k, t_{k+1}]$.

For the negated predicate $\neg\pi$, the proof of Proposition 3.4.1 establishes that the negation encoding selects an index $j \in \{1, \ldots, \mathrm{Row}(H)\}$ such that

$$H_j B_k(t) - b_j \geq (r_k - \epsilon_k)\|H_j\|, \qquad \forall t \in [t_k, t_{k+1}]. \tag{3.4.24}$$

It follows that

$$\begin{aligned} H_j y(t) - b_j &= H_j B_k(t) - b_j + H_j\big(y(t) - B_k(t)\big) \\ &\geq (r_k - \epsilon_k)\|H_j\| - \|H_j\|\|y(t) - B_k(t)\| \\ &\geq 0. \end{aligned} \tag{3.4.25}$$

This implies $y(t) \notin \mathrm{Poly}(H, b)$ and hence $(y, t) \models \neg\pi$, $\forall t \in [t_k, t_{k+1}]$. Thus, the claim holds for the atomic formulas.

Suppose that the claim holds for the immediate subformulas. For each temporal interval $\mathcal{I} = [a, b]$, the assumption that $a/\Delta t, b/\Delta t \in \mathbb{N}_0$ and the definitions of k_l and k_r give, for every $t \in [t_k, t_{k+1}]$, $(t + \mathcal{I}) \cap \mathbb{T} \subseteq \bigcup_{j=k_l}^{k_r} [t_j, t_{j+1}]$. Moreover, every segment indexed by $k' \in \{k_l, \ldots, k_r\}$ intersects the shifted temporal window $[t_{k'}, t_{k'+1}] \cap \big((t + \mathcal{I}) \cap \mathbb{T}\big) \neq \emptyset$, $\forall k' \in \{k_l, \ldots, k_r\}$, $\forall t \in [t_k, t_{k+1}]$.

- *Conjunction*: Let $\varphi = \varphi_1 \wedge \varphi_2$. If $z_k^{\varphi_1 \wedge \varphi_2} = 1$, then (3.4.3) implies $z_k^{\varphi_1} = 1$ and $z_k^{\varphi_2} = 1$. By the induction hypothesis, both φ_1 and φ_2 are satisfied at every $t \in [t_k, t_{k+1}]$. Therefore, $(y, t) \models \varphi_1 \wedge \varphi_2$, $\forall t \in [t_k, t_{k+1}]$.

- *Disjunction*: Let $\varphi = \varphi_1 \vee \varphi_2$. If $z_k^{\varphi_1 \vee \varphi_2} = 1$, then (3.4.4) implies that at least one of $z_k^{\varphi_1}$ and $z_k^{\varphi_2}$ is equal to one. By the induction hypothesis, the corresponding subformula is satisfied at every $t \in [t_k, t_{k+1}]$. Therefore, $(y, t) \models \varphi_1 \vee \varphi_2$, $\forall t \in [t_k, t_{k+1}]$.

- *Always*: Let $\varphi = \Box_{\mathcal{I}} \varphi_1$. If $z_k^{\Box_{\mathcal{I}} \varphi_1} = 1$, then (3.4.5) implies $z_{k'}^{\varphi_1} = 1$, for all $k' \in \{k_l, \ldots, k_r\}$. By the induction hypothesis, $(y, t') \models \varphi_1$, $\forall t' \in [t_{k'}, t_{k'+1}]$ and $\forall k' \in \{k_l, \ldots, k_r\}$. Since $(t + \mathcal{I}) \cap \mathbb{T} \subseteq \bigcup_{k'=k_l}^{k_r} [t_{k'}, t_{k'+1}]$, it follows that $(y, t') \models \varphi_1$, $\forall t' \in (t + \mathcal{I}) \cap \mathbb{T}$. Therefore, $(y, t) \models \Box_{\mathcal{I}} \varphi_1$, $\forall t \in [t_k, t_{k+1}]$.

- *Eventually*: Let $\varphi = \Diamond_{\mathcal{I}} \varphi_1$. If $z_k^{\Diamond_{\mathcal{I}} \varphi_1} = 1$, then (3.4.6) implies that there exists $k' \in \{k_l, \ldots, k_r\}$ such that $z_{k'}^{\varphi_1} = 1$. For every $t \in [t_k, t_{k+1}]$, choose $t' \in [t_{k'}, t_{k'+1}] \cap ((t + \mathcal{I}) \cap \mathbb{T})$. By the induction hypothesis, $(y, t') \models \varphi_1$. Therefore, $(y, t) \models \Diamond_{\mathcal{I}} \varphi_1$, $\forall t \in [t_k, t_{k+1}]$.

- *Until*: Let $\varphi = \varphi_1 \mathcal{U}_{\mathcal{I}} \varphi_2$. If $z_k^{\varphi_1 \mathcal{U}_{\mathcal{I}} \varphi_2} = 1$, then (3.4.7) implies that there exists $k' \in \{k_l, \ldots, k_r\}$ such that $z_{k'}^{\varphi_2} = 1$ and $z_{k''}^{\varphi_1} = 1$, $\forall k'' \in \{k, \ldots, k'\}$. For every $t \in [t_k, t_{k+1}]$, choose $t' \in [t_{k'}, t_{k'+1}] \cap \big((t + \mathcal{I}) \cap \mathbb{T}\big)$. By the induction hypothesis, $(y, t') \models \varphi_2$. Moreover, $[t, t'] \subseteq \bigcup_{k''=k}^{j} [t_{k''}, t_{k''+1}]$. Since $z_{k''}^{\varphi_1} = 1$ for every $k'' \in \{k, \ldots, k'\}$, the induction hypothesis gives $(y, t'') \models \varphi_1$, for all $t'' \in [t, t']$. Therefore, $(y, t) \models \varphi_1 \mathcal{U}_{\mathcal{I}} \varphi_2$, $\forall t \in [t_k, t_{k+1}]$.

This completes the structural induction. Applying the result to the full specification φ gives $z_k^{\varphi} = 1$, which implies that $(y, t) \models \varphi$ for all $t \in [t_k, t_{k+1}]$. This proves the theorem. □

Generally, we choose $\Delta t \leq \min_i \operatorname{dur}(\mathcal{I}_i)$, where $\mathcal{I}_i = [a_i, b_i]$ denotes the time interval associated with the i-th temporal operator appearing in the STL formula and $\operatorname{dur}(\mathcal{I}_i) := b_i - a_i$, ensuring that there is at least one segment within the interval. A smaller Δt leads to a finer Bézier partition, whereas a larger Δt can reduce the computational burden. Therefore, for the sake of computational efficiency, a larger Δt may be selected, provided that the temporal-alignment condition $a_i/\Delta t, b_i/\Delta t \in \mathbb{N}_0$ remains satisfied. For example, the specification $\varphi = \square_{[0,2]} R$ requires a robot to stay in a region for at least 2 seconds, though it may remain longer due to arrival and departure times. This situation commonly occurs in cases with a long horizon and also motivates the development of time robustness [31]. Moreover, a generic method for predefining the minimum segment number has not been established, as it is influenced by factors such as task complexity, environment, and time horizon [74]. A similar challenge arises in other works, such as [135, 140].

Finally, the encoding complexity is analyzed for a trajectory with N Bézier curve segments. Since binary variables are introduced only for disjunctive operators, let $|\varphi|$ denote the number of operators in the STL formula, and let N_{win} denote the maximum number of segment indices involved in any temporal window. For each segment, the encoding of a temporal operator involves at most $\mathcal{O}(N_{\text{win}})$ disjunctive terms. Since such a z-formula is constructed for each of the N segments, encoding all temporal operators in φ requires $\mathcal{O}(N|\varphi|N_{\text{win}})$ binary variables. In the worst case, $N_{\text{win}} = \mathcal{O}(N)$, yielding $\mathcal{O}(N^2|\varphi|)$ binary variables.

3.5 Simulations

In this section, we evaluate the performance of the proposed algorithm on several benchmarks. The algorithm is implemented in Python, with MICP problems solved using

Gurobi [52], and the tracking problem addressed with cvxpy [28]. All experiments were carried out on an Apple M1 chip with 8 cores and 16 GB RAM.

3.5.1 Mission Scenarios

We implement the algorithms on the four scenarios:

- Reach-avoid:

$$\varphi = \left(\square_{[0,T]} \neg \mathbf{B}\right) \wedge \left(\square_{[0,T]} \neg \mathbf{Y}\right) \wedge \left(\lozenge_{[0,T-2]} \square_{[0,2]} \mathbf{R}\right) \wedge \left(\lozenge_{[0,T-2]} \square_{[0,2]} \mathbf{G}\right).$$

- Complex reach-avoid:

$$\begin{aligned}\varphi = &\square_{[0,T]} \left(\wedge_{i=1}^{4} \neg \mathbf{Obs}_i\right) \wedge \lozenge_{[0,T-3]} \left(\square_{[0,3]} \left(\vee_{i=1}^{4} \mathbf{Chg}_{1,i}\right)\right) \\ &\wedge \lozenge_{[0,T-3]} \left(\square_{[0,3]} \left(\vee_{i=1}^{3} \mathbf{Chg}_{2,i}\right)\right).\end{aligned}$$

- Narrow passing:

$$\varphi = \lozenge_{[0,T-3]} \left(\square_{[0,3]} \left(\vee_{i=1}^{2} \mathbf{Chg}_i\right)\right) \wedge \square_{[0,T]} \left(\wedge_{i=1}^{4} \neg \mathbf{Obs}_i\right) \wedge \left(\lozenge_{[0,T]} \mathbf{Goal}\right).$$

- Door puzzle:

$$\varphi = \wedge_{i=1}^{4} \left(\neg \mathbf{D}_i \ \mathcal{U}_{[0,2]} \mathbf{K}_i\right) \wedge_{i=1}^{5} \left(\square_{[0,T]} \neg \mathbf{W}_i\right) \wedge \left(\lozenge_{[0,T]} \mathbf{G}\right).$$

3.5.2 Comparisons

To evaluate the performance of the proposed approach, we compare it with two baselines: a standard discrete-time MICP method [11] and a PWL path planning method [135]. To compare the trackability of the reference trajectories generated by the different planning methods, we use the same continuous-time 2D kinematic bicycle model and a standard MPC tracking controller [15].

For any $t \in [0, T]$, let $q(t) := [x(t)\ y(t)\ v(t)\ \theta(t)]^\top \in \mathbb{R}^4$ be the vehicle state, where $x(t)$ and $y(t)$ denote the position, $v(t)$ denotes the linear velocity, and $\theta(t)$ denotes the heading angle. The control input is $u(t) := [a(t)\ \delta(t)]^\top \in \mathbb{R}^2$, where $a(t)$ and $\delta(t)$ denote

the acceleration and steering angle, respectively. The continuous-time kinematic bicycle model is

$$\begin{aligned} \dot{x}(t) &= v(t)\cos\theta(t), \\ \dot{y}(t) &= v(t)\sin\theta(t), \\ \dot{v}(t) &= a(t), \\ \dot{\theta}(t) &= \frac{v(t)}{w_l}\tan\delta(t), \end{aligned} \tag{3.5.1}$$

where w_l denotes the wheelbase.

For each planning method, the corresponding reference trajectory is tracked using the same MPC controller with a sampling time of 0.2 s and a prediction horizon of $N_p = 10$ steps. Let $p(t) := \begin{bmatrix} x(t) & y(t) \end{bmatrix}^\top$ denote the position of the trajectory tracked by the MPC controller, and let $p_d(t)$ denote the reference position generated by the corresponding planning method. The position tracking error is defined as

$$e_p(t) := p(t) - p_d(t), \quad t \in \mathbb{T}. \tag{3.5.2}$$

All tracking simulations are initialized at their corresponding reference positions, so that $\|e_p(0)\| = 0$. The reported tracking errors are evaluated at the same sampling instants.

Figure 3.2 presents the planning and tracking results for the benchmark scenarios. The left panels compare the planned trajectories generated by different planning methods. The proposed method generates smoother trajectories than PWL and ensures safety with fewer points than MICP. The right panels show the corresponding position tracking errors $\|e_p(t)\|$ for $t \in \mathbb{T}$. The proposed trajectories achieve smaller tracking errors overall. Moreover, Table 3.1 reports the corresponding runtimes, robustness margins, and maximum position tracking errors $e_p^{\max} := \max_{t\in\mathbb{T}} \|e_p(t)\|$.

Runtimes

The runtimes of the planning methods depend on the number of segments N. Throughout the comparison, N denotes the number of trajectory segments, equivalently the number of discretization intervals, defined by $N+1$ segment endpoints. To ensure a fair comparison, N is selected separately for each baseline to obtain safe trajectories without unnecessary discretization. As shown in Figure 3.2(a), when the MICP method uses the same number of segments as the proposed method ($N = 11$), its trajectory crosses Region $\mathbf{Y}$. A finer discretization with a larger N is therefore needed to avoid the obstacle, which increases the computational cost. Although the PWL method can use fewer segments, too few segments

can lead to sharp turns, which consequently increase the tracking errors, as demonstrated in Figure 3.2(a).

Robustness margins

Table 3.1 compares the time-varying robustness margins of the proposed method with the uniform margins ρ^* used by the PWL and MICP methods under consistent parameter settings, including the same prescribed minimum robustness margin. The uniform margin ρ^* coincides with the minimum time-varying robustness margin at the reported precision. A uniform formulation assigns a single robustness margin to the entire trajectory and is therefore limited by its most constrained portion. In contrast, the proposed formulation retains local robustness margins along the trajectory, with substantially larger maximum margins in less constrained regions. In the constrained portions of these benchmarks, the optimized Bézier segments are nearly linear, resulting in small deviation bounds ϵ_k.

Position tracking errors

As shown in Table 3.1, the proposed approach yields a smaller maximum position tracking error. This performance can be attributed to the smoothness and C^2 continuity of the Bézier reference trajectories, whereas the PWL trajectories are only C^0-continuous. Moreover, the PWL method minimizes travel time, which can induce abrupt velocity changes between adjacent segments and thereby increase the tracking errors. The MICP method also achieves comparatively small tracking errors because its objective penalizes velocities and accelerations, thereby promoting smoother trajectories.

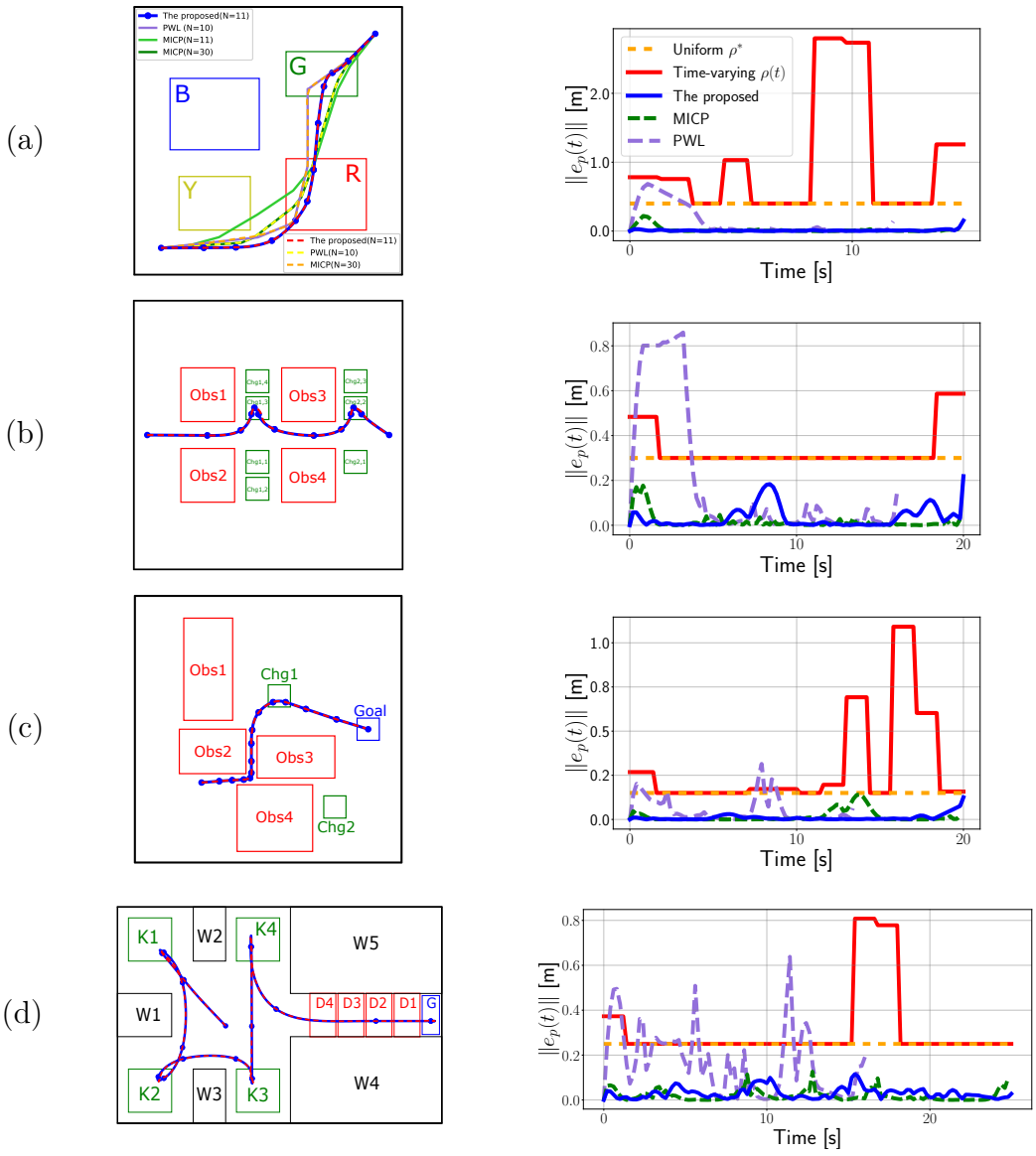


Figure 3.2: Four benchmark scenarios: (a) reach-avoid, (b) complex reach-avoid, (c) narrow passing, and (d) door puzzle. Left panels show planned and tracked trajectories, denoted by solid and dashed lines, respectively. Blue dots mark the endpoints of the Bézier curves; right panels show tracking errors. The MICP trajectory crosses Region $\mathbf{Y}$ with $N = 11$ in (a); similar cases occur in the other scenarios. For clarity, only the Bézier curves are shown in (b)–(d), and the legend in (a) applies to these panels as well.

Table 3.1: Comparisons of benchmarks over time horizons (TH): runtimes, robustness margins, and maximum position tracking errors $e_p^{\max}$.

Mission	TH (s)	Runtime (s) (N: # segments)			Robustness margins			Maximum position tracking error $e_p^{\max}$		
		Proposed approach	Standard MICP[11]	PWL [135]	Time-varying $\rho(t)$(min, max)	Standard MICP ρ^*	PWL ρ^*	Proposed approach	Standard MICP[11]	PWL [135]
Reach-avoid	15	**0.06** ($N = 11$)	0.12 ($N = 30$)	0.58 ($N = 10$)	(0.40, 2.80)	0.40	0.40	**0.15**	0.22	0.68
	50	**0.23** ($N = 15$)	0.47 ($N = 50$)	0.35 ($N = 17$)	(0.40, 3.11)	0.40	0.40	**0.03**	0.03	0.34
Complex reach-avoid	20	2.80 ($N = 12$)	3.22 ($N = 25$)	**0.36** ($N = 12$)	(0.30, 0.59)	0.30	0.30	0.22	**0.18**	0.86
	50	**10.82** ($N = 15$)	15.75 ($N = 34$)	12.54 ($N = 15$)	(0.30, 1.80)	0.30	0.30	**0.04**	0.04	0.44
Narrow passing	20	1.33 ($N = 14$)	2.42 ($N = 33$)	**0.57** ($N = 10$)	(0.15, 1.18)	0.15	0.15	**0.12**	0.14	0.31
	50	**5.23** ($N = 14$)	11.97 ($N = 35$)	5.94 ($N = 16$)	(0.15, 1.17)	0.15	0.15	**0.11**	0.12	0.48
Door puzzle	25	5.19 ($N = 18$)	19.52 ($N = 50$)	**4.67** ($N = 16$)	(0.25, 0.81)	0.25	0.25	**0.11**	0.13	0.64
	50	8.65 ($N = 20$)	17.18 ($N = 52$)	**5.03** ($N = 18$)	(0.25, 0.87)	0.25	0.25	**0.13**	0.27	0.21

3.6 Summary

In this chapter, we proposed a trajectory planning approach for generating Bézier curves under STL specifications with time-varying robustness margins. The simulations show that, compared with PWL and MICP, the proposed method can generate smoother, time-varying robust reference trajectories that are easier to track.

The results of this chapter focus on the planning level, where the generated trajectory is treated as a reference trajectory. However, for robotic systems such as quadrotors, the executed trajectory may deviate from the planned trajectory due to nonlinear dynamics, attitude constraints, and tracking errors. Therefore, the next chapter develops an STL control synthesis framework for the quadrotor system by integrating geometric control on SE(3) with time-varying tracking-error bounds.

Chapter 4

Quadrotor STL Control Synthesis with Time-Varying Bounds

4.1 Introduction

The preceding chapters focused on generating trajectories that satisfy STL specifications. In particular, the previous chapter incorporated an MPC-based controller to demonstrate that the generated trajectories are trackable and that the resulting tracking errors can remain below the corresponding STL robustness margins in simulation. However, this validation does not by itself provide a general theoretical guarantee on the closed-loop tracking performance.

This chapter shifts the focus from STL planning to control synthesis for quadrotor trajectory tracking. Building on geometric tracking control with diagonal matrix gains (DMG-GC) for quadrotor systems evolving on SE(3), we derive tracking-error bounds and sufficient conditions under which the bounds hold. These conditions ensure that the controlled quadrotor trajectory satisfies both the required dynamic constraints and the STL specifications.

The main contributions of this chapter are summarized as follows:

- A closed-loop error analysis framework is established for geometric quadrotor tracking control with diagonal matrix gains on SE(3). The analysis yields time-varying bounds on the position and velocity tracking errors, together with sufficient conditions under which these bounds hold.

- The derived tracking-error bounds are integrated into an STL-guided control-synthesis framework, ensuring that the controlled quadrotor trajectory satisfies the required dynamic constraints and STL specifications. The proposed framework is further evaluated through comparative simulations with representative baseline controllers.

4.2 Quadrotor Dynamics Model

Table 4.1 lists the physical variables of the quadrotor dynamics model. As depicted in Figure 4.1, let $\mathcal{F}_I = \{e_1, e_2, e_3\}$ denote an inertial frame, where e_1 and e_2 span the horizontal plane and e_3 points upward, opposite to the direction of gravity. Let $\mathcal{F}_B(t) = \{b_1(t), b_2(t), b_3(t)\}$ be the body-fixed frame attached to the quadrotor center of mass, where $b_1(t)$, $b_2(t)$, and $b_3(t)$ denote the forward, lateral, and thrust directions, respectively.

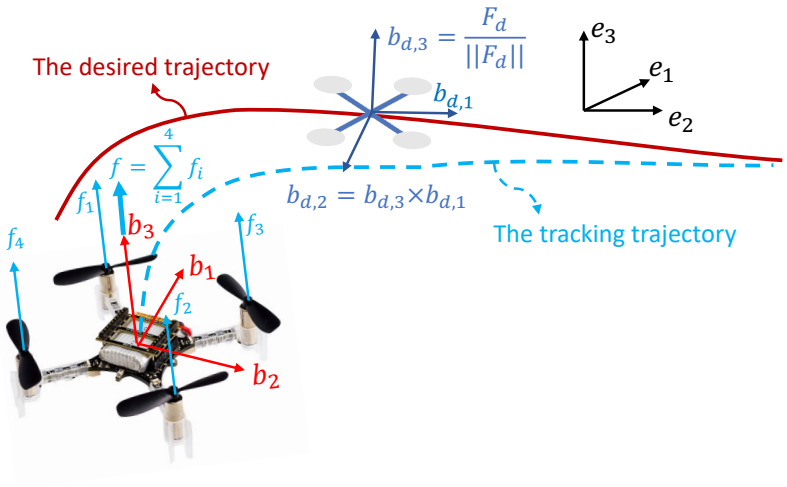


Figure 4.1: The inertial frame $\mathcal{F}_I$, the body frame $\mathcal{F}_B$, and the desired attitude $R_d(t) = \big[b_{d,1}(t),\, b_{d,2}(t),\, b_{d,3}(t)\big]$.

The configuration space of the quadrotor is the special Euclidean group SE(3), parameterized by a position $p \in \mathbb{R}^3$ and a rotation matrix $R \in \mathrm{SO}(3)$. As in Section 2.3, we use $\mathfrak{so}(3)$ to denote the Lie algebra of SO(3), which is the space of skew-symmetric matrices. The hat map $\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$ is defined by $x \mapsto \hat{x}$, where $\hat{x}y = x \times y$ for all $y \in \mathbb{R}^3$. The vee map $(\cdot)^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ denotes the inverse of $\wedge$.

Table 4.1: Physical variables in quadrotor dynamics.

Symbol	Physical meaning
$\mathcal{F}_I$	Inertial frame $\mathcal{F}_I = \{e_1, e_2, e_3\}$
$\mathcal{F}_B(t)$	Body frame $\mathcal{F}_B(t) = \{b_1(t), b_2(t), b_3(t)\}$
$m \in \mathbb{R}$	Total mass
$g \in \mathbb{R}$	Gravitational acceleration, $g = 9.81\,\mathrm{m/s^2}$
$J \in \mathbb{S}^3_{++}$	Diagonal positive-definite inertia matrix, $J = \mathrm{diag}(J_1, J_2, J_3)$
$p(t) \in \mathbb{R}^3$	Position of the center of mass
$v(t) \in \mathbb{R}^3$	Velocity of the center of mass
$R(t) \in \mathrm{SO}(3)$	Rotation matrix with $b_i(t) = R(t)e_i$, $i \in \{1, 2, 3\}$
$\omega(t) \in \mathbb{R}^3$	Angular velocity in the body-fixed frame
$f_j(t) \in \mathbb{R}$	Thrust of the j-th propeller along $b_3(t)$, $j \in \{1, 2, 3, 4\}$
$f(t) \in \mathbb{R}$	Total thrust, $f(t) = \sum_{j=1}^4 f_j(t)$
$\tau(t) \in \mathbb{R}^3$	Torque

The quadrotor dynamics are given by [78]:

$$\dot{p}(t) = v(t), \tag{4.2.1}$$
$$\dot{v}(t) = -ge_3 + m^{-1} f(t) R(t) e_3, \tag{4.2.2}$$
$$\dot{R}(t) = R(t)\hat{\omega}(t), \tag{4.2.3}$$
$$\dot{\omega}(t) = J^{-1}(-\omega(t) \times J\omega(t) + \tau(t)). \tag{4.2.4}$$

4.3 Geometric Control with Matrix Gains

Let $y_d : \mathbb{T} \to \mathbb{R}^d$ be a desired $\mathcal{C}^4$ position trajectory[1]. For the tracking error-bound analysis developed in Section 4.5, we impose the following bounded-acceleration assumption.

Assumption 4.3.1 ([127]). *For the desired trajectory $y_d : \mathbb{T} \to \mathbb{R}^d$, assume that there*

[1] A curve is $\mathcal{C}^k$ if it is k-times continuously differentiable. Note the fourth derivative of the reference trajectory is required to compute the derivative of the desired angular velocity.

exists a vector $b_a \in \mathbb{R}^3_{++}$ *such that the element-wise inequality on the absolute values*

$$|ge_3 + \ddot{y}_d(t)| \leq b_a, \quad \forall t \in \mathbb{T}, \tag{4.3.1}$$

holds.

This condition provides a uniform element-wise bound on the mass-normalized desired force associated with the reference trajectory.

Let $y_{\mathrm{tr}} : \mathbb{T} \to \mathbb{R}^3$ be a tracking trajectory with rotation matrix $R(t) \in \mathrm{SO}(3)$ and angular velocity $\omega(t) \in \mathbb{R}^3$. The gain matrices K_p, K_v, K_R, and $K_\omega \in \mathbb{S}^3_{++}$ are defined as diagonal matrices of the form $K_* = \mathrm{diag}(k_{*_1}, k_{*_2}, k_{*_3})$ for $* \in \{p, v, R, \omega\}$, where, for each $i \in \{1, 2, 3\}$, k_{*_i} is a positive scalar gain to be optimized in Section 4.5.4. Define the position error, velocity error, weighted rotational error, and angular velocity error by $e_p(t)$, $e_v(t)$, $e_{K,R}(t)$, and $e_\omega(t) \in \mathbb{R}^3$, respectively, and define the attitude error function by $\Psi_K(t) \in \mathbb{R}$ [56]:

$$e_p(t) := y_{\mathrm{tr}}(t) - y_d(t), \tag{4.3.2}$$
$$e_v(t) := \dot{y}_{\mathrm{tr}}(t) - \dot{y}_d(t), \tag{4.3.3}$$
$$e_{K,R}(t) := \frac{1}{2}\left(K_R R_d^\top(t) R(t) - R^\top(t) R_d(t) K_R\right)^\vee, \tag{4.3.4}$$
$$e_\omega(t) := \omega(t) - R^\top(t) R_d(t) \omega_d(t), \tag{4.3.5}$$
$$\Psi_K(t) := \frac{1}{2}\,\mathrm{tr}\left(K_R(\mathrm{I}_3 - R_d^\top(t) R(t))\right). \tag{4.3.6}$$

For all $t \in \mathbb{T}$, define $G(t) := R_d^\top(t) R(t) \in \mathrm{SO}(3)$. The error dynamics associated with (4.3.2)–(4.3.6) are given by

$$\dot{e}_p(t) = e_v(t), \tag{4.3.7}$$
$$\dot{e}_v(t) = -\frac{1}{m}(K_p e_p(t) + K_v e_v(t) - \Delta_f(t)), \tag{4.3.8}$$
$$\dot{\Psi}_K(t) = e_{K,R}^\top(t) e_\omega(t), \tag{4.3.9}$$
$$\dot{e}_{K,R}(t) = \mathfrak{C}(t) e_\omega(t), \tag{4.3.10}$$
$$\dot{e}_\omega(t) = -J^{-1}(e_{K,R}(t) + K_\omega e_\omega(t)), \tag{4.3.11}$$

where

$$\mathfrak{C}(t) := \frac{1}{2}\left(\mathrm{tr}(K_R G(t))\mathrm{I}_3 - G^\top(t) K_R\right), \tag{4.3.12}$$
$$\Delta_f(t) = \|F_d(t)\|\left((b_{3,d}^\top(t) b_3(t)) b_3(t) - b_{3,d}(t)\right). \tag{4.3.13}$$

The detailed derivation of the error dynamics in (4.3.7)–(4.3.11) is provided in Appendix A.1.

To prepare for the Lyapunov analysis, we next derive the bound on $\mathfrak{C}(t)$, relate the configuration error $\Psi_K(t)$ to $e_{K,R}(t)$, and establish a bound on the force perturbation $\Delta_f(t)$.

Lemma 4.3.2 ([76, Proposition 2]). *For all $t \in \mathbb{T}$, the matrix $\mathfrak{C}(t)$ satisfies*

$$\|\mathfrak{C}(t)\| \leq \frac{1}{\sqrt{2}} \operatorname{tr}(K_R). \tag{4.3.14}$$

Proposition 4.3.3 ([56, Proposition 1]). *Suppose that $K_R = \operatorname{diag}(k_{R_1}, k_{R_2}, k_{R_3})$ has distinct positive diagonal entries, and define*

$$h_1 = \min\left\{k_{R_1} + k_{R_2}, k_{R_1} + k_{R_3}, k_{R_2} + k_{R_3}\right\}, \tag{4.3.15}$$
$$h_2 = \max\left\{(k_{R_1} - k_{R_2})^2, (k_{R_1} - k_{R_3})^2, (k_{R_2} - k_{R_3})^2\right\}, \tag{4.3.16}$$
$$h_3 = \max\left\{k_{R_1} + k_{R_2}, k_{R_1} + k_{R_3}, k_{R_2} + k_{R_3}\right\}. \tag{4.3.17}$$

If $\Psi_K(t) < \psi < h_1$ for some positive constant ψ, then

$$g_1 \|e_{K,R}(t)\|^2 \leq \Psi_K(t) \leq g_2 \|e_{K,R}(t)\|^2, \tag{4.3.18}$$

where $g_1 = \frac{h_1}{h_2 + h_3^2}$ and $g_2 = \frac{h_3}{h_1(h_1 - \psi)}$, with $0 < g_1 < g_2$.

Following the geometric control structure, we adopt matrix gains to allow anisotropic weighting of the tracking errors. The control force $f(t)$ and torque $\tau(t)$ are defined as

$$f(t) = F_d^\top(t) R(t) e_3, \tag{4.3.19}$$
$$F_d(t) = -K_p e_p(t) - K_v e_v(t) + mge_3 + m\ddot{y}_d(t), \tag{4.3.20}$$
$$\tau(t) = -e_{K,R}(t) - K_\omega e_\omega(t) + \omega(t) \times J\omega(t) \tag{4.3.21}$$
$$- J\left(\hat{\omega}(t) R^\top(t) R_d(t) \omega_d(t) - R^\top(t) R_d(t) \dot{\omega}_d(t)\right). \tag{4.3.22}$$

The desired attitude $R_d(t)$ and desired angular velocity $\omega_d(t)$ are constructed from the desired force direction. For each $t \in \mathbb{T}$, define

$$R_d(t) = \left[b_{d,1}(t),\, b_{d,2}(t),\, b_{d,3}(t)\right] \in \mathrm{SO}(3), \tag{4.3.23}$$
$$\omega_d(t) = \left(R_d^\top(t) \dot{R}_d(t)\right)^\vee \in \mathbb{R}^3, \tag{4.3.24}$$

where the desired body-frame directions are chosen according to [78] as

$$b_{d,1}(t) = \frac{1}{\|F_d(t)\|} \begin{bmatrix} F_{d,3}(t) + \dfrac{F_{d,2}^2(t)}{\|F_d(t)\| + F_{d,3}(t)} \\ -\dfrac{F_{d,1}(t)F_{d,2}(t)}{\|F_d(t)\| + F_{d,3}(t)} \\ -F_{d,1}(t) \end{bmatrix}, \tag{4.3.25}$$

$$b_{d,2}(t) = \frac{1}{\|F_d(t)\|} \begin{bmatrix} -\dfrac{F_{d,1}(t)F_{d,2}(t)}{\|F_d(t)\| + F_{d,3}(t)} \\ F_{d,3}(t) + \dfrac{F_{d,1}^2(t)}{\|F_d(t)\| + F_{d,3}(t)} \\ -F_{d,2}(t) \end{bmatrix}, \tag{4.3.26}$$

$$b_{d,3}(t) = \frac{F_d(t)}{\|F_d(t)\|}. \tag{4.3.27}$$

To ensure that the above constructions are well defined, we assume that, for all $t \in \mathbb{T}$,

$$\|F_d(t)\| \neq 0, \tag{4.3.28}$$

$$\|F_d(t)\| + F_{d,3}(t) \neq 0. \tag{4.3.29}$$

The first condition ensures that the desired thrust direction $b_{d,3}(t)$ is defined, whereas the second ensures that the denominators in the expressions for $b_{d,1}(t)$ and $b_{d,2}(t)$ are nonzero. Similar assumptions have been adopted in related studies, such as [78, 127].

Lemma 4.3.4. *Let $K_R = \mathrm{diag}(k_{R_1}, k_{R_2}, k_{R_3})$ have distinct positive diagonal entries, and suppose that $\Psi_K(t) < \psi < h_1$. Then*

$$\left\| b_{3,d}(t) - b_{3,d}^\top(t) b_3(t)\, b_3(t) \right\| \leq \sqrt{\frac{4g_2}{h_1}}\, \|e_{K,R}(t)\|, \tag{4.3.30}$$

and hence

$$\|\Delta_f(t)\| \leq \left(\|[K_p \;\; K_v] M_1^{-1/2}\| \sqrt{\mathcal{V}_1(t)} + m\|b_a\| \right) \sqrt{\frac{4g_2}{h_1}}\, \|e_{K,R}(t)\|. \tag{4.3.31}$$

Proof. Define $G(t) := R_d^\top(t) R(t) \in \mathrm{SO}(3)$, and let $G_{3,3}(t)$ denote the $(3,3)$ entry of $G(t)$. Since $b_{3,d}(t) = R_d(t) e_3$ and $b_3(t) = R(t) e_3$, one has

$$b_{3,d}^\top(t) b_3(t) = e_3^\top R_d^\top(t) R(t) e_3 = G_{3,3}(t).$$

Therefore,

$$\begin{aligned}\left\|b_{3,d}(t) - b_{3,d}^{\top}(t)b_3(t)\, b_3(t)\right\|^2 &= 1 - \left(b_{3,d}^{\top}(t)b_3(t)\right)^2 \\ &= 1 - G_{3,3}^2(t).\end{aligned} \tag{4.3.32}$$

Let (u, θ) be an axis–angle representation of $G(t)$, where $u = [u_1, u_2, u_3]^{\top} \in \mathbb{R}^3$ is the unit rotation axis satisfying $\|u\| = 1$, and $\theta \in [0, \pi]$ is the rotation angle, so that $G(t) = \exp(\theta\hat{u})$. If $\theta = 0$, then $G(t) = \mathrm{I}_3$ and the desired inequality is trivial. Hence, assume $\theta \neq 0$. By Rodrigues' formula [111, Lemma 2.3],

$$G(t) = \mathrm{I}_3 + \sin\theta\, \hat{u} + (1 - \cos\theta)\hat{u}^2. \tag{4.3.33}$$

Thus,

$$G_{3,3}(t) = \cos\theta + (1 - \cos\theta)u_3^2. \tag{4.3.34}$$

It follows that

$$\begin{aligned}1 - G_{3,3}^2(t) &= \big(1 - G_{3,3}(t)\big)\big(1 + G_{3,3}(t)\big) \\ &\leq 2\big(1 - G_{3,3}(t)\big) \\ &= 2(1 - \cos\theta)(1 - u_3^2) \\ &\leq 2(1 - \cos\theta).\end{aligned} \tag{4.3.35}$$

Moreover, from [76], $\Psi_K(t)$ can be expressed as

$$\Psi_K(t) = \frac{1 - \cos\theta}{2}\Big[(k_{R_2} + k_{R_3})u_1^2 + (k_{R_3} + k_{R_1})u_2^2 + (k_{R_1} + k_{R_2})u_3^2\Big]. \tag{4.3.36}$$

Since $h_1 = \min\{k_{R_1} + k_{R_2},\, k_{R_1} + k_{R_3},\, k_{R_2} + k_{R_3}\}$ in Proposition 4.3.3, and since $\|u\| = 1$, we have

$$\Psi_K(t) \geq \frac{1 - \cos\theta}{2} h_1 \left(u_1^2 + u_2^2 + u_3^2\right) = \frac{h_1}{2}\big(1 - \cos\theta\big). \tag{4.3.37}$$

Therefore,

$$1 - \cos\theta \leq \frac{2}{h_1}\Psi_K(t). \tag{4.3.38}$$

Combining (4.3.35) and (4.3.38) gives

$$\left\|b_{3,d}(t) - b_{3,d}^{\top}(t)b_3(t)\, b_3(t)\right\|^2 \leq \frac{4}{h_1}\Psi_K(t). \tag{4.3.39}$$

Since $\Psi_K(t) < \psi < h_1$, Proposition 4.3.3 gives $\Psi_K(t) \leq g_2\|e_{K,R}(t)\|^2$. Hence,

$$\left\|b_{3,d}(t) - b_{3,d}^{\top}(t)b_3(t)\, b_3(t)\right\|^2 \leq \frac{4g_2}{h_1}\|e_{K,R}(t)\|^2. \tag{4.3.40}$$

Taking the square root yields

$$\left\|b_{3,d}(t) - b_{3,d}^{\top}(t)b_3(t)\, b_3(t)\right\| \leq \sqrt{\frac{4g_2}{h_1}}\, \|e_{K,R}(t)\|, \tag{4.3.41}$$

which proves (4.3.30).

By Assumption 4.3.1 and the monotonicity of the Euclidean norm with respect to element-wise absolute values, we have $\|ge_3 + \ddot{y}_d(t)\| \leq \|b_a\|$, $\forall t \in \mathbb{T}$. Therefore, by (4.3.20) and (4.5.1), one has

$$\begin{aligned} \|F_d(t)\| &\leq \| - K_p e_p(t) - K_v e_v(t)\| + m\|ge_3 + \ddot{y}_d(t)\| \\ &\leq \|[K_p \quad K_v] z_1(t)\| + m\|b_a\| \\ &\leq \|[K_p \quad K_v] M_1^{-1/2}\| \sqrt{\mathcal{V}_1(t)} + m\|b_a\|. \end{aligned} \tag{4.3.42}$$

Therefore,

$$\begin{aligned} \|\Delta_f(t)\| &\leq \|F_d(t)\| \left\| \left(b_{3,d}^{\top}(t)b_3(t)\right) b_3(t) - b_{3,d}(t) \right\| \\ &\leq \left(\|[K_p \;\; K_v] M_1^{-1/2}\| \sqrt{\mathcal{V}_1(t)} + m\|b_a\| \right) \sqrt{\frac{4g_2}{h_1}}\, \|e_{K,R}(t)\|. \end{aligned} \tag{4.3.43}$$

This proves (4.3.31). □

4.4 Problem Statement

Given an STL specification φ, the objective is to synthesize a reference trajectory y_d and a geometric feedback controller for a quadrotor such that the resulting closed-loop tracking trajectory y_{tr} satisfies the specification φ.

To account for the tracking error and physical safety requirements, we impose a time-varying lower bound $\gamma(t)$ on the STL robustness margin $\rho^{\varphi}(t)$ by defining

$$\gamma(t) := \Gamma(t) + \gamma_c, \qquad \forall t \in \mathbb{T}, \tag{4.4.1}$$

where $\gamma_c > 0$ is a constant physical safety margin, and $\Gamma : \mathbb{T} \to \mathbb{R}_{>0}$ is a non-increasing position tracking-error envelope satisfying

$$\|y_{\text{tr}}(t) - y_d(t)\| \leq \Gamma(t), \qquad \forall t \in \mathbb{T}. \tag{4.4.2}$$

The desired trajectory y_d is required to satisfy the STL specification φ with robustness margin $\rho^{\varphi}(t)$ such that $\rho^{\varphi}(t) \geq \gamma(t)$ for all $t \in \mathbb{T}$. Since $\gamma_c > 0$, we have $\Gamma(t) < \gamma(t) \leq \rho^{\varphi}(t)$, and therefore

$$\|y_{\text{tr}}(t) - y_d(t)\| \leq \Gamma(t) < \rho^{\varphi}(t), \qquad \forall t \in \mathbb{T}, \tag{4.4.3}$$

which implies $y_{\text{tr}} \models \varphi$.

As shown in Figure 4.2, this overall objective is divided into the following two sub-objectives:

Objective 1: Establish sufficient conditions under which the geometric controller admits time-varying tracking error bounds, and optimize the multidimensional control gains to obtain tighter bounds on the translational tracking errors.

Objective 2: Given an STL specification φ, construct the non-increasing position tracking-error envelope $\Gamma(t)$ from the position tracking-error bound, and generate a reference trajectory y_d parameterized by piecewise Bézier curves such that y_d satisfies φ with STL robustness margin $\rho^{\varphi}(t) \geq \gamma(t)$, while ensuring safety, smoothness, and compliance with the dynamic requirements imposed by the geometric controller.

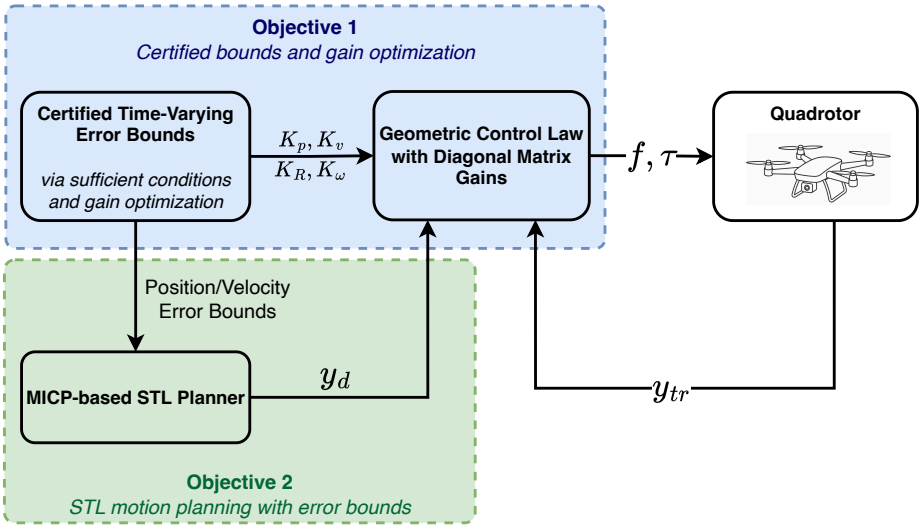


Figure 4.2: Configuration of the proposed framework.

Section 4.5 and Section 4.6 address these two objectives, respectively. In particular, Section 4.5 derives the time-varying tracking error bounds and optimizes the geometric control gains, while Section 4.6 uses the resulting position tracking-error envelope to construct the STL-constrained reference trajectory.

4.5 Time-varying Tracking Error Bound Analysis

This section derives time-varying bounds on the tracking errors of the closed-loop quadrotor system. The tracking error bounds are established via Lyapunov analysis of the translational and rotational tracking errors.

4.5.1 Lyapunov Function for Translation Error

The translational-error vector $z_1(t) \in \mathbb{R}^6$ is defined by $z_1^\top(t) := \begin{bmatrix} e_p^\top(t) & e_v^\top(t) \end{bmatrix}$ for any $t \in \mathbb{T}$. Define the translation Lyapunov candidate function $\mathcal{V}_1 : \mathbb{T} \to \mathbb{R}$ as:

$$\mathcal{V}_1(t) := z_1^\top(t) M_1 z_1(t), \tag{4.5.1}$$

where

$$M_1 = \frac{1}{2} \begin{bmatrix} K_p & c_1 \mathrm{I}_3 \\ c_1 \mathrm{I}_3 & m \mathrm{I}_3 \end{bmatrix} \in \mathbb{S}^6_{++}, \tag{4.5.2}$$

$$c_1 = \nu_1 \min \left\{ \sqrt{m \underline{\lambda}(K_p)}, \min_{i=1,2,3} \left(\frac{4 m k_{p_i} k_{v_i}}{k_{v_i}^2 + 4 m k_{p_i}} \right) \right\} \in \mathbb{R}, \tag{4.5.3}$$

with $\nu_1 \in (0,1)$ is a tuning constant. The cross term scaled by c_1 is included to capture the coupling between the position and velocity errors. By construction, the choice of c_1 in (4.5.3) guarantees that M_1 and the matrix W_1 defined below are positive definite.

Using the translational error dynamics in (4.3.7) and (4.3.8), the time derivative of $\mathcal{V}_1(t)$ is obtained as

$$\dot{\mathcal{V}}_1(t) = -z_1^\top(t) W_1 z_1(t) + \Delta_f^\top(t) \begin{bmatrix} \frac{c_1}{m} \mathrm{I}_3 & \mathrm{I}_3 \end{bmatrix} z_1(t), \tag{4.5.4}$$

where

$$W_1 = \frac{1}{2m} \begin{bmatrix} 2 c_1 K_p & c_1 K_v \\ c_1 K_v & 2m \left(K_v - c_1 \mathrm{I}_3 \right) \end{bmatrix} \in \mathbb{S}^6_{++}, \tag{4.5.5}$$

4.5.2 Lyapunov Function for Rotational Error

The rotational-error vector $z_2(t) \in \mathbb{R}^6$ is defined by $z_2^\top(t) := \begin{bmatrix} e_{K,R}^\top(t) & e_\omega^\top(t) \end{bmatrix}$ for any $t \in \mathbb{T}$. Define the rotational Lyapunov candidate function $\mathcal{V}_2 : \mathbb{T} \to \mathbb{R}$ as

$$\mathcal{V}_2(t) := \frac{1}{2} e_\omega^\top(t) J e_\omega(t) + \Psi_K(t) + c_2 e_{K,R}^\top(t) e_\omega(t), \tag{4.5.6}$$

where

$$c_2 := \nu_2 \min\left\{\sqrt{2g_1\underline{\lambda}(J)},\ \frac{\sqrt{2}\underline{\lambda}(K_\omega)}{\operatorname{tr}(K_R)}, \min_{i\in\{1,2,3\}} \frac{4J_i k_{\omega_i}}{2\sqrt{2}J_i \operatorname{tr}(K_R) + k_{\omega_i}^2}\right\} \in \mathbb{R}_{++}, \tag{4.5.7}$$

with $\nu_2 \in (0,1)$ defined as a tuning constant and g_1 defined in Proposition 4.3.3. Similar to the translational Lyapunov construction, the cross term scaled by c_2 captures the coupling between $e_{K,R}(t)$ and $e_\omega(t)$, while the choice of c_2 in (4.5.7) ensures that the following $M_{2,1}$, $M_{2,2}$, and W_2 are positive definite.

It follows from (4.3.18) in Proposition 4.3.3 that $\mathcal{V}_2(t)$ is bounded as

$$z_2^\top(t) M_{2,1} z_2(t) \le \mathcal{V}_2(t) \le z_2^\top(t) M_{2,2} z_2(t), \tag{4.5.8}$$

where $M_{2,1}, M_{2,2} \in \mathbb{S}^6_{++}$ are defined as

$$M_{2,1} = \frac{1}{2}\begin{bmatrix} 2g_1 \mathrm{I}_3 & c_2 \mathrm{I}_3 \\ c_2 \mathrm{I}_3 & J \end{bmatrix} \in \mathbb{S}^6_{++}, \tag{4.5.9}$$

$$M_{2,2} = \frac{1}{2}\begin{bmatrix} 2g_2 \mathrm{I}_3 & c_2 \mathrm{I}_3 \\ c_2 \mathrm{I}_3 & J \end{bmatrix} \in \mathbb{S}^6_{++}. \tag{4.5.10}$$

where g_1 and g_2 are defined in Proposition 4.3.3.

Using the rotational-error dynamics in (4.3.9)–(4.3.11), the time derivative of $\mathcal{V}_2(t)$ is

$$\dot{\mathcal{V}}_2(t) = -c_2 e_{K,R}^\top(t) J^{-1} e_{K,R}(t) - c_2 e_\omega^\top(t) K_\omega J^{-1} e_{K,R}(t) - e_\omega^\top(t)(K_\omega - c_2 \mathfrak{C}(t)) e_\omega(t). \tag{4.5.11}$$

By Lemma 4.3.2, we have

$$e_\omega^\top(t)\mathfrak{C}(t)e_\omega(t) \le \|\mathfrak{C}(t)\|\,\|e_\omega(t)\|^2 \le \frac{1}{\sqrt{2}}\operatorname{tr}(K_R)\|e_\omega(t)\|^2. \tag{4.5.12}$$

Therefore,

$$\begin{aligned} \dot{\mathcal{V}}_2(t) &\le -c_2 e_{K,R}^\top(t) J^{-1} e_{K,R}(t) - c_2 e_\omega^\top(t) K_\omega J^{-1} e_{K,R}(t) \\ &\quad - e_\omega^\top(t)\left(K_\omega - \frac{c_2}{\sqrt{2}}\operatorname{tr}(K_R)\mathrm{I}_3\right) e_\omega(t) \\ &= -z_2^\top(t) W_2 z_2(t) \end{aligned} \tag{4.5.13}$$

where

$$W_2 = \begin{bmatrix} c_2 J^{-1} & \frac{c_2}{2} K_\omega J^{-1} \\ \frac{c_2}{2} K_\omega J^{-1} & K_\omega - \frac{c_2}{\sqrt{2}}\operatorname{tr}(K_R) I_3 \end{bmatrix} \in \mathbb{S}^6_{++}. \tag{4.5.14}$$

4.5.3 Time-varying Error Tracking Bounds

Theorem 4.5.1 (Time-varying Lyapunov function bounds)**.** *Suppose that the conditions of Lemma 4.3.2, Proposition 4.3.3, and Lemma 4.3.4 hold. Then the following exponentially decaying bounds hold for all $t \in \mathbb{T}$:*

$$\sqrt{\mathcal{V}_1(t)} \leq \mathcal{L}_1(\mathcal{V}_1(0), \mathcal{V}_2(0), t), \qquad \sqrt{\mathcal{V}_2(t)} \leq \mathcal{L}_2(\mathcal{V}_2(0), t), \tag{4.5.15}$$

where

$$\mathcal{L}_1(\mathcal{V}_1(0), \mathcal{V}_2(0), t) := e^{-\frac{\alpha_0}{2}t} e^{\frac{\alpha_1\sqrt{\mathcal{V}_2(0)}}{\beta}} \left[\sqrt{\mathcal{V}_1(0)} + \frac{\alpha_2\sqrt{\mathcal{V}_2(0)}}{2} \int_0^t e^{\frac{1}{2}(\alpha_0-\beta)s}\, ds\right], \tag{4.5.16}$$

$$\mathcal{L}_2(\mathcal{V}_2(0), t) := e^{-\frac{\beta}{2}t}\sqrt{\mathcal{V}_2(0)}, \tag{4.5.17}$$

with

$$\alpha_0 := \underline{\lambda}\left(M_1^{-\frac{1}{2}} W_1 M_1^{-\frac{1}{2}}\right), \tag{4.5.18}$$

$$\beta := \underline{\lambda}\left(M_{2,2}^{-\frac{1}{2}} W_2 M_{2,2}^{-\frac{1}{2}}\right), \tag{4.5.19}$$

$$\alpha_1 := \left\| [K_p \quad K_v]\, M_1^{-\frac{1}{2}} \right\| \beta', \tag{4.5.20}$$

$$\alpha_2 := m\, \|b_a\|\, \beta', \tag{4.5.21}$$

$$\beta' := \left\| \left[\frac{c_1}{m}\mathrm{I}_3 \quad \mathrm{I}_3\right] M_1^{-\frac{1}{2}} \right\| \left\| [\mathrm{I}_3 \quad 0_{3\times 3}]\, M_{2,1}^{-\frac{1}{2}} \right\| \sqrt{\frac{4g_2}{h_1}}. \tag{4.5.22}$$

Proof. From (4.5.13), we have $\dot{\mathcal{V}}_2(t) \leq -z_2^\top(t) W_2 z_2(t)$. Let $\iota(t) := M_{2,2}^{1/2} z_2(t)$. With β in (4.5.19) and the inequality $z_2^\top M_{2,2} z_2 \geq \mathcal{V}_2$ in (4.5.8), we have

$$z_2^\top(t) W_2 z_2(t) = \iota^\top(t) \left(M_{2,2}^{-\frac{1}{2}} W_2 M_{2,2}^{-\frac{1}{2}}\right) \iota(t) \geq \beta \|\iota(t)\|^2 = \beta z_2^\top(t) M_{2,2} z_2(t) \geq \beta \mathcal{V}_2(t),$$

Therefore, $\dot{\mathcal{V}}_2(t) \leq -\beta \mathcal{V}_2(t)$. Applying the comparison principle gives

$$\sqrt{\mathcal{V}_2(t)} \leq e^{-\frac{\beta}{2}t}\sqrt{\mathcal{V}_2(0)} = \mathcal{L}_2(\mathcal{V}_2(0), t). \tag{4.5.23}$$

Since $e_{K,R}(t) = [I_3 \;\; 0_{3\times 3}]\, z_2(t)$, there is

$$\|e_{K,R}(t)\| \leq \left\| [\mathrm{I}_3 \quad 0_{3\times 3}]\, M_{2,1}^{-\frac{1}{2}} \right\| \sqrt{\mathcal{V}_2(t)}. \tag{4.5.24}$$

Next, we derive an upper bound for $\mathcal{V}_1(t)$. Using α_0 in (4.5.18), we have

$$z_1^\top(t) W_1 z_1(t) \geq \alpha_0 z_1^\top(t) M_1 z_1(t) = \alpha_0 \mathcal{V}_1(t). \tag{4.5.25}$$

Moreover, since $\mathcal{V}_1(t) = z_1^\top(t) M_1 z_1(t) = \|M_1^{\frac{1}{2}} z_1(t)\|^2$, we have

$$\begin{aligned} \left\| \begin{bmatrix} \frac{c_1}{m} \mathrm{I}_3 & \mathrm{I}_3 \end{bmatrix} z_1(t) \right\| &= \left\| \begin{bmatrix} \frac{c_1}{m} \mathrm{I}_3 & \mathrm{I}_3 \end{bmatrix} M_1^{-\frac{1}{2}} M_1^{\frac{1}{2}} z_1(t) \right\| \\ &\leq \left\| \begin{bmatrix} \frac{c_1}{m} \mathrm{I}_3 & \mathrm{I}_3 \end{bmatrix} M_1^{-\frac{1}{2}} \right\| \sqrt{\mathcal{V}_1(t)}. \end{aligned} \tag{4.5.26}$$

Using Lemma 4.3.4 together with (4.5.24), we obtain

$$\|\Delta_f(t)\| \left\| \begin{bmatrix} \frac{c_1}{m} \mathrm{I}_3 & \mathrm{I}_3 \end{bmatrix} z_1(t) \right\| \leq \alpha_1 \sqrt{\mathcal{V}_2(t)}\, \mathcal{V}_1(t) + \alpha_2 \sqrt{\mathcal{V}_2(t) \mathcal{V}_1(t)}. \tag{4.5.27}$$

Using the bound of $\sqrt{\mathcal{V}_2(t)}$ in (4.5.23), and substituting (4.5.25) and (4.5.27) into $\dot{\mathcal{V}}_1(t)$ in (4.5.4), we obtain

$$\dot{\mathcal{V}}_1(t) \leq -\big(\alpha_0 - \alpha_1 e^{-\frac{\beta}{2}t} \sqrt{\mathcal{V}_2(0)}\big) \mathcal{V}_1(t) + \alpha_2 e^{-\frac{\beta}{2}t} \sqrt{\mathcal{V}_2(0)} \sqrt{\mathcal{V}_1(t)}. \tag{4.5.28}$$

Applying the comparison principle in [62, Section 3.4] again gives

$$\sqrt{\mathcal{V}_1(t)} \leq e^{-\frac{\alpha_0}{2}t} e^{\frac{\alpha_1 \sqrt{\mathcal{V}_2(0)}}{\beta}} \left[\sqrt{\mathcal{V}_1(0)} + \frac{\alpha_2 \sqrt{\mathcal{V}_2(0)}}{2} \int_0^t e^{\frac{1}{2}(\alpha_0 - \beta)s}\, ds \right] = \mathcal{L}_1(\mathcal{V}_1(0), \mathcal{V}_2(0), t).$$

□

Both $\mathcal{L}_1(\mathcal{V}_1(0), \mathcal{V}_2(0), t)$ and $\mathcal{L}_2(\mathcal{V}_2(0), t)$ in (4.5.16) and (4.5.17) decay exponentially with respect to t, while their values depend on the initial quantities $\mathcal{V}_1(0)$ and $\mathcal{V}_2(0)$. To obtain explicit bounds depending only on prescribed initial-error limits, Proposition 4.5.2 introduces a set of admissible initial conditions. These conditions provide upper bounds on $\mathcal{V}_1(0)$ and $\mathcal{V}_2(0)$, which can be substituted into $\mathcal{L}_1$ and $\mathcal{L}_2$. In addition, they ensure that $\Psi_K(t) < \psi < h_1$ for all $t \geq 0$, so that Proposition 4.3.3 and Lemma 4.3.4 remain applicable along the trajectory.

Proposition 4.5.2 (Initial conditions). *Let $y_d : \mathbb{T} \to \mathbb{R}^3$ be a $\mathcal{C}^4$ trajectory satisfying (4.3.1). Define $\psi := \underline{\lambda}(K_R)\psi_K$ and $\psi_K \in \left(0, \frac{h_1}{\underline{\lambda}(K_R)}\right)$, so that $\psi \in (0, h_1)$. Assume that, for some $\alpha_\psi \in [0.5, 1)$ and $\overline{\mathcal{V}}_1 > 0$, the initial conditions satisfy*

$$\Psi_K(0) < \alpha_\psi\, \psi, \tag{4.5.29}$$

$$\frac{1}{2} e_\omega^\top(0) J e_\omega(0) \leq (1 - \alpha_\psi)\psi, \tag{4.5.30}$$

$$\mathcal{V}_1(0) \leq \overline{\mathcal{V}}_1. \tag{4.5.31}$$

Then, $\Psi_K(t) < \psi$ for all $t \in \mathbb{T}$, and hence the assumptions of Proposition 4.3.3 hold. Moreover, it follows that

$$\mathcal{V}_2(0) \leq \overline{\mathcal{V}}_2 := \left(1 + c_2\sqrt{\frac{2\alpha_\psi(1-\alpha_\psi)}{\underline{\lambda}(J)\, g_1}}\right)\psi. \tag{4.5.32}$$

Proof. Define $\mathcal{E}(t) := \frac{1}{2}e_\omega^\top(t)Je_\omega(t) + \Psi_K(t)$. Using (4.3.9) and (4.3.11), we obtain

$$\dot{\mathcal{E}}(t) = e_\omega^\top(t)J\dot{e}_\omega(t) + \dot{\Psi}_K(t) = -e_\omega^\top(t)K_\omega e_\omega(t) \leq 0. \tag{4.5.33}$$

Hence, $\mathcal{E}(t) \leq \mathcal{E}(0)$ for all $t \geq 0$. Moreover, since $\frac{1}{2}e_\omega^\top(t)Je_\omega(t) \geq 0$, we have $\Psi_K(t) \leq \mathcal{E}(t)$. With the initial conditions (4.5.29)–(4.5.30), it holds that

$$\mathcal{E}(0) = \frac{1}{2}e_\omega^\top(0)Je_\omega(0) + \Psi_K(0) < \psi < h_1. \tag{4.5.34}$$

Therefore,

$$\Psi_K(t) \leq \mathcal{E}(t) \leq \mathcal{E}(0) < \psi < h_1, \qquad \forall t \geq 0. \tag{4.5.35}$$

Thus, the condition $\Psi_K(t) < \psi < h_1$ required by Proposition 4.3.3 and Lemma 4.3.4 holds for all $t \geq 0$. Given $g_1\|e_{K,R}(0)\|^2 \leq \Psi_K(0)$ from Proposition 4.3.3, and by $\mathcal{V}_2(t)$ in (4.5.6), one has

$$\begin{aligned}
\mathcal{V}_2(0) &= \frac{1}{2}e_\omega^\top(0)Je_\omega(0) + \Psi_K(0) + c_2 e_{K,R}^\top(0)e_\omega(0) \\
&\leq (1-\alpha_\psi)\,\psi + \alpha_\psi\psi + c_2\,\|e_{K,R}(0)\|\,\|e_\omega(0)\| \\
&\leq \psi + c_2\sqrt{\frac{\psi}{g_1}}\,\left\|J^{-1/2}\right\|\,\left\|J^{1/2}e_\omega(0)\right\| \\
&\leq \left(1 + c_2\sqrt{\frac{2(1-\alpha_\psi)\alpha_\psi}{\underline{\lambda}(J)g_1}}\right)\psi = \overline{\mathcal{V}}_2.
\end{aligned} \tag{4.5.36}$$

□

By Proposition 4.5.2 and the definition of $\mathcal{L}_1$ in (4.5.16), the admissible initial conditions imply $\mathcal{L}_1(\mathcal{V}_1(0), \mathcal{V}_2(0), t) \leq \mathcal{L}_1(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t)$. We now derive explicit bounds on the translational tracking errors $e_p(t)$ and $e_v(t)$.

Corollary 4.5.3 (Time-varying tracking error bounds)**.** *Suppose that the conditions of Theorem 4.5.1 hold, and that the initial conditions in Proposition 4.5.2 are satisfied. Then, for all $t \in \mathbb{T}$, the translational tracking errors satisfy*

$$\|e_p(t)\| \leq \mathcal{L}_p(t), \tag{4.5.37}$$
$$\|e_v(t)\| \leq \mathcal{L}_v(t), \tag{4.5.38}$$

where

$$\mathcal{L}_p(t) := \left\| [\mathrm{I}_3 \quad 0_{3\times 3}]\, M_1^{-\frac{1}{2}} \right\| \mathcal{L}_1(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t), \tag{4.5.39}$$
$$\mathcal{L}_v(t) := \left\| [0_{3\times 3} \quad \mathrm{I}_3]\, M_1^{-\frac{1}{2}} \right\| \mathcal{L}_1(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t). \tag{4.5.40}$$

Bounds on $e_{K,R}(t)$ and $e_\omega(t)$ can be similarly obtained from $\mathcal{L}_2$. Since the translational error bounds are mainly used in the subsequent STL trajectory construction, we focus on $e_p(t)$ and $e_v(t)$ here. Compared with the error bounds in [127], the proposed bounds are less conservative. In particular, the translational error bounds are obtained directly from the decay estimate of $\mathcal{V}_1(t)$, while the effect of the rotational dynamics enters only through the attitude-induced perturbation term. This avoids embedding the rotational initial error directly into the translational tracking-error bound.

4.5.4 Optimization-based Control Gains

The high-dimensional nonlinear quadrotor dynamics make multidimensional control-gain selection complex, and unsuitable gains may lead to unnecessarily large upper bounds. According to (4.5.39) and (4.5.40), control gains can be chosen by minimizing the bound $\mathcal{L}_1(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t)$.

Let $\varkappa = \frac{\alpha_0 - \beta}{2}$. If $\varkappa \neq 0$, then $\dot{\mathcal{L}}_1(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t)$ is affine in $e^{\varkappa t}$, which is monotone in t. If $\varkappa = 0$, then $\dot{\mathcal{L}}_1\left(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t\right)$ is affine in t. In both cases, the equation $\dot{\mathcal{L}}_1\left(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t\right) = 0$ has at most one solution. If the equation admits a solution t_s, then it follows by direct differentiation that $\ddot{\mathcal{L}}_1\left(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t_s\right) < 0$. Hence, t_s is the unique stationary point and a local maximizer of $\mathcal{L}_1\left(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t\right)$. Define $t^* := \max\big(0, \min(t_s, T)\big) \in \mathbb{T}$, with the convention that $t_s := 0$ when no solution exists. Then we define the maximum value of $\mathcal{L}_1$ over $\mathbb{T}$ as

$$\mathcal{L}_{1,\max} := \max_{t\in\mathbb{T}} \mathcal{L}_1(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t) = \mathcal{L}_1(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t^*). \tag{4.5.41}$$

Furthermore, to satisfy the assumption in Proposition 4.3.3 that the diagonal entries of K_R are distinct, the condition $\Delta_{K_R} = \sum_{1\le i<j\le 3} \Delta_{ij} = 3$ is imposed, where

$$\Delta_{ij} = \begin{cases} 1, & \left|k_{R_i} - k_{R_j}\right| \ge \delta_r \\ 0, & \text{otherwise.} \end{cases} \tag{4.5.42}$$

In this thesis, $\delta_r = 1$. Problem 4.5.4 is formulated to represent *Objective 1* and can be solved by differential evolution [134], a parallelizable global optimizer for nonconvex problems that requires no initial guess.

Problem 4.5.4. *The control gains* $\{K_p, K_v, K_R, K_\omega\}$ *and tuning parameters* $\{\nu_1, \nu_2\}$ *are determined by solving the following optimization problem:*

$$\begin{aligned} \underset{K_p,K_v,K_R,K_\omega,\nu_1,\nu_2}{\arg\min} \quad & \mathcal{L}_{1,\max} \\ \textit{such that} \quad & 0 < \nu_1, \nu_2 < 1, \\ & \Delta_{K_R} = 3, \\ & \underline{k} \le k_{p_i}, k_{v_i}, k_{R_i}, k_{\omega_i} \le \bar{k},\ i \in \{1,2,3\}. \end{aligned} \tag{4.5.43}$$

Here, $0 < \underline{k} < \bar{k}$ *are constants and denote the lower and upper bounds of the control gains, respectively.*

4.6 STL-Constrained Trajectory Generation

Once the control gains are determined by solving Problem 4.5.4, the time-varying position error bound $\mathcal{L}_p\left(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t\right)$ is obtained. We next focus on *Objective 2*, namely the generation of an STL-constrained reference trajectory parameterized by piecewise Bézier curves, where a non-increasing envelope $\Gamma(t)$ is constructed from $\mathcal{L}_p$ and incorporated into the lower bound $\gamma(t)$ on the STL robustness margin. We build on the MICP formulation in (3.2.4) developed in Chapter 3, and augment it with the time-varying tracking-error envelope and the dynamic constraints required by the geometric controller.

According to (4.4.1), we choose $\Gamma(t) = \widetilde{\mathcal{L}}_p(t)$, where

$$\widetilde{\mathcal{L}}_p(t) := \begin{cases} \mathcal{L}_p(t^*), & t \le t^*, \\ \mathcal{L}_p(t), & t > t^*. \end{cases} \tag{4.6.1}$$

With this choice, $\Gamma(t)$ is non-increasing and satisfies $\|y_{\mathrm{tr}}(t) - y_d(t)\| \leq \Gamma(t)$ for all $t \in \mathbb{T}$. Therefore, enforcing an STL robustness margin larger than $\gamma(t) = \Gamma(t)+\gamma_c$ for the reference trajectory guarantees that the actual tracking trajectory satisfies the STL specification. The resulting STL-constrained trajectory generation problem is formulated as the following MICP.

Problem 4.6.1. *Given an STL specification φ, parameterize the desired position trajectory $y_d : \mathbb{T} \to \mathbb{R}^3$ using N piecewise Bézier curves of degree n, where $B_k(t)$ denotes the curve defined over $[t_k, t_{k+1}]$ for $k \in \{0, \ldots, N-1\}$. Let $\mathcal{C}_k := \{c_{k,i}\}_{i=0}^{n}$ denote the set of control points of B_k, and let $v_k, a_k \in \mathbb{R}^3$ denote the componentwise velocity and acceleration bounds for the k-th segment. For each segment, the STL robustness measure is required to satisfy $\rho_k^{\varphi}(r_k, \epsilon_k) := r_k - \epsilon_k \geq \gamma(t_k) = \Gamma(t_k) + \gamma_c$, and the satisfaction of φ is encoded using the binary variables z_k^{φ}. Satisfaction of the complete formula at the initial index is enforced by setting the top-level Boolean variable $z_0^{\varphi} = 1$.*

The resulting MICP maximizes the STL robustness while minimizing the motion effort of the reference trajectory:

$$\begin{aligned} \underset{\{\mathcal{C}_k, v_k, a_k, r_k, \epsilon_k, z_k^{\varphi}\}_{k=0}^{N-1}}{\arg\min} \quad & \sum_{k=0}^{N-1} \left(-\mathcal{W}\rho_k^{\varphi}(r_k, \epsilon_k) + \mathcal{Q}\|v_k\|_1 + \mathcal{R}\|a_k\|_1\right) \\ \textit{subject to} \quad & B_k \textit{ and } B_{k+1} \textit{ are joined with } C^4 \textit{ continuity}, \\ & B_k \textit{ satisfies the kinematic constraints}, \\ & \rho_k^{\varphi}(r_k, \epsilon_k) \geq \gamma(t_k), \\ & z_0^{\varphi} = 1. \end{aligned} \tag{4.6.2}$$

Here, $\mathcal{W}, \mathcal{Q}, \mathcal{R} \in \mathbb{R}_{>0}$ are user-defined weights. The continuity constraints are enforced for $k \in \{0, \ldots, N-2\}$, whereas the kinematic and robustness constraints are enforced for $k \in \{0, \ldots, N-1\}$.

The following subsections specify the continuity and dynamic constraints used in Problem 4.6.1. The STL robustness constraints follow the MICP encoding introduced in Chapter 3, with the lower bound updated to $\gamma(t_k)$ on each segment.

4.6.1 Continuity Constraints

To ensure $\mathcal{C}^4$ continuity at the junction time $t = t_{k+1}$ between consecutive Bézier segments B_k and B_{k+1}, the derivatives up to order four are matched at the junction.

Equivalently, the following finite-difference constraints are imposed on the control points for $\mathfrak{q} \in \{0, 1, 2, 3, 4\}$:

$$\sum_{\mathfrak{p}=0}^{\mathfrak{q}} (-1)^{\mathfrak{p}} \binom{\mathfrak{q}}{\mathfrak{p}} c_{k,n-\mathfrak{p}} = \sum_{\mathfrak{p}=0}^{\mathfrak{q}} (-1)^{\mathfrak{q}-\mathfrak{p}} \binom{\mathfrak{q}}{\mathfrak{p}} c_{k+1,\mathfrak{p}}. \tag{4.6.3}$$

These constraints enforce the equality between the backward finite differences of the end control points of B_k and the forward finite differences of the start control points of B_{k+1}, both evaluated at $t = t_{k+1}$.

4.6.2 Kinematic Constraints

Velocity Constraints

To ensure that the actual tracking velocity satisfies the componentwise limit $|\dot{y}_{\mathrm{tr}}(t)| \leq v_{\max}$, we use the non-increasing velocity-error envelope $\widetilde{\mathcal{L}}_v(t)$, defined analogously to $\widetilde{\mathcal{L}}_p(t)$. Specifically, $\widetilde{\mathcal{L}}_v(t) = \mathcal{L}_v(t^*)$ for $t \leq t^*$ and $\widetilde{\mathcal{L}}_v(t) = \mathcal{L}_v(t)$ for $t > t^*$. Let v_k be the componentwise velocity bound for the k-th Bézier segment. For all $i \in \{0, \ldots, n-1\}$, impose

$$|c_{k,i+1} - c_{k,i}| \leq \frac{v_k \Delta t}{n}, \tag{4.6.4}$$

$$0 < v_k \leq v_{\max} - \widetilde{\mathcal{L}}_v(t_k)\mathbf{1}_3. \tag{4.6.5}$$

Since $\dot{B}_k(t)$ is a Bézier curve of degree $n-1$ and its Bernstein polynomials satisfy the condition $\sum_{i=0}^{n-1} b_i^{n-1}(\tau) = 1$ for all $\tau \in [0, 1]$ [42, Section 4.2], the convex-hull property gives $|\dot{B}_k(t)| \leq v_k$. Therefore, by the triangle inequality,

$$|\dot{y}_{\mathrm{tr}}(t)| \leq |\dot{B}_k(t)| + \widetilde{\mathcal{L}}_v(t_k)\mathbf{1}_3 \leq v_k + \widetilde{\mathcal{L}}_v(t_k)\mathbf{1}_3 \leq v_{\max}. \tag{4.6.6}$$

Acceleration Constraints

Likewise, let a_k denote the componentwise acceleration bound for the k-th Bézier segment. Define $\Delta^2 c_{k,i} := c_{k,i+2} - 2c_{k,i+1} + c_{k,i}$ for all $i \in \{0, \ldots, n-2\}$. To bound the segment acceleration and enforce the condition in (4.3.1), impose

$$|\Delta^2 c_{k,i}| \leq \frac{a_k \Delta t^2}{n(n-1)}, \tag{4.6.7}$$

$$\left| g e_3 + \frac{n(n-1)}{\Delta t^2} \Delta^2 c_{k,i} \right| \leq b_a. \tag{4.6.8}$$

Since $\ddot{B}_k(t)$ is a Bézier curve of degree $n-2$ with control points $\frac{n(n-1)}{\Delta t^2}\Delta^2 c_{k,i}$, the convex-hull property implies $|\ddot{B}_k(t)| \le a_k$ and $|ge_3 + \ddot{B}_k(t)| \le b_a$ for all $t \in [t_k, t_{k+1}]$.

STL Robustness Constraints

The STL robustness constraints are encoded using the mixed-integer formulation introduced in Chapter 3. The only modification is that the required robustness lower bound is made time-varying: for each segment k, the robustness variable is constrained by

$$\rho_k^{\varphi}(\mathcal{C}_k, a_k) \ge \gamma(t_k).$$

This ensures that the reference trajectory satisfies the STL specification with sufficient margin to compensate for the tracking-error envelope and the physical safety margin.

4.7 Simulations

This section evaluates the proposed framework through a simulation scenario. The simulations are implemented in Python. Problem 4.5.4 is solved using differential evolution in SciPy [142], and Problem 4.6.1 is solved as an MICP using Gurobi [52]. All simulations are conducted on an 8-core Apple M1 machine with 16 GB of RAM. We first examine how the initial-condition parameters affect the control gain optimization in Problem 4.5.4. Then, using the reference trajectory generated by Problem 4.6.1, we evaluate the closed-loop tracking performance of the proposed framework.

4.7.1 Control Gain Optimization

The inertia matrix and mass of each quadrotor are $J = \mathrm{diag}(8.20, 8.45, 13.77) \times 10^{-2}\,\mathrm{kg}\cdot\mathrm{m}^2$ and $m = 4.34\,\mathrm{kg}$, respectively, as given in [127]. The remaining parameters are $v_{\max} = [2,2,2]^\top$, $b_a = [1,1,11]^\top$, and $\gamma_c = 0.35$.

To select the initial-condition parameters in (4.5.29)–(4.5.31) of Proposition 4.5.2, we conduct a sensitivity study with respect to ψ_K, α_ψ, and $\overline{\mathcal{V}}_1$. Table 4.2 reports how individual variations of these parameters affect the peak position and velocity bounds $\mathcal{L}_{\max,p} := \max_{t\in[0,T]} \mathcal{L}_p(t)$ and $\mathcal{L}_{\max,v} := \max_{t\in[0,T]} \mathcal{L}_v(t)$, as well as the initial-condition (IC) feasibility rate. The IC feasibility rate is computed from 5000 independent samples drawn from the baseline initial-condition distribution

$$x_0, v_0 \sim \mathrm{Unif}([-0.2, 0.2]^3), \qquad \omega_0, \mathfrak{r}_0 \sim \mathrm{Unif}([-0.1, 0.1]^3),$$

with the initial attitude constructed as $R_0 = \exp(\hat{\mathfrak{r}}_0)$. Here, $\mathrm{Unif}([a,b]^3)$ denotes a vector whose components are independently and uniformly distributed on $[a,b]$. It is defined as the percentage of samples satisfying the initial-condition constraints. For each tuning case, one of ψ_K, α_ψ, and $\overline{\mathcal{V}}_1$ is varied around the nominal setting

$$(\psi_K, \alpha_\psi, \overline{\mathcal{V}}_1) = (0.05, 0.7, 0.4),$$

while the other two parameters are kept fixed.

Consistent with (4.5.29) and (4.5.30), Table 4.2 shows that increasing ψ_K allows larger initial attitude and angular-velocity errors, thereby improving IC feasibility at the cost of looser peak bounds. By contrast, for a fixed ψ_K, α_ψ mainly affects the relative strictness of the initial attitude and angular-velocity bounds. Table 4.2 indicates that α_ψ has a relatively mild effect on both the peak bounds and the IC feasibility rate over the tested range. As for $\overline{\mathcal{V}}_1$, increasing $\overline{\mathcal{V}}_1$ relaxes the initial-condition constraint in (4.5.31), allowing larger initial position and velocity errors. This is reflected in the higher IC feasibility reported in Table 4.2 for both methods. The peak bounds increase for DMG-GC but remain nearly unchanged for GC.

Table 4.2: Effects of individual variations of ψ_K, α_ψ, and $\overline{\mathcal{V}}_1$ around the nominal setting $(0.05, 0.7, 0.4)$ on the peak bounds $\mathcal{L}_{\max,p}$, $\mathcal{L}_{\max,v}$, and the IC feasibility rate.

Param.	Value	$\mathcal{L}_{\max,p}$ (m)		$\mathcal{L}_{\max,v}$ (m/s)		IC Feasibility (%)	
		DMG-GC	GC	DMG-GC	GC	DMG-GC	GC
ψ_K	0.005[1]	0.37	**0.36**	0.77	**0.72**	12.20	**13.02**
	0.05	**0.61**	0.80	**1.46**	1.90	**31.64**	29.02
α_ψ	0.6	**0.61**	0.78	**1.46**	1.92	**30.98**	25.86
	0.8	**0.60**	0.80	**1.43**	1.92	**31.42**	27.56
$\overline{\mathcal{V}}_1$	0.3	**0.52**	0.79	**1.36**	1.90	**15.82**	15.62
	0.5	**0.69**	0.80	**1.54**	1.91	**52.28**	41.42

[1] $\psi_K = 0.005$ follows $\bar{\Psi}$ in GC [127], since both quantify the allowable initial attitude error.

Overall, choosing $(\psi_K, \alpha_\psi, \overline{\mathcal{V}}_1) = (0.05, 0.7, 0.4)$ provides a balance between IC feasibility and peak-bound conservatism. The optimized control gains are given by

$$K_p = \mathrm{diag}(25.2, 24.6, 25.3), \quad K_v = \mathrm{diag}(14.7, 14.7, 14.8),$$
$$K_R = \mathrm{diag}(28.9, 27.9, 29.9), \quad K_\omega = \mathrm{diag}(2.2, 1.8, 2.3),$$

and the auxiliary parameters are $\nu_1 = 0.75$ and $\nu_2 = 0.79$. Therefore, $\overline{\mathcal{V}}_2 = 1.46$ is obtained from (4.5.32), yielding the maximum position and velocity error bounds $\mathcal{L}_{\max,p} = 0.61$ m and $\mathcal{L}_{\max,v} = 1.46$ m/s, respectively. Furthermore, Table 4.2 shows that, compared with [127], the proposed method yields tighter error bounds in most cases, due to the additional tuning flexibility provided by multidimensional gain matrices.

4.7.2 Tracking Performance

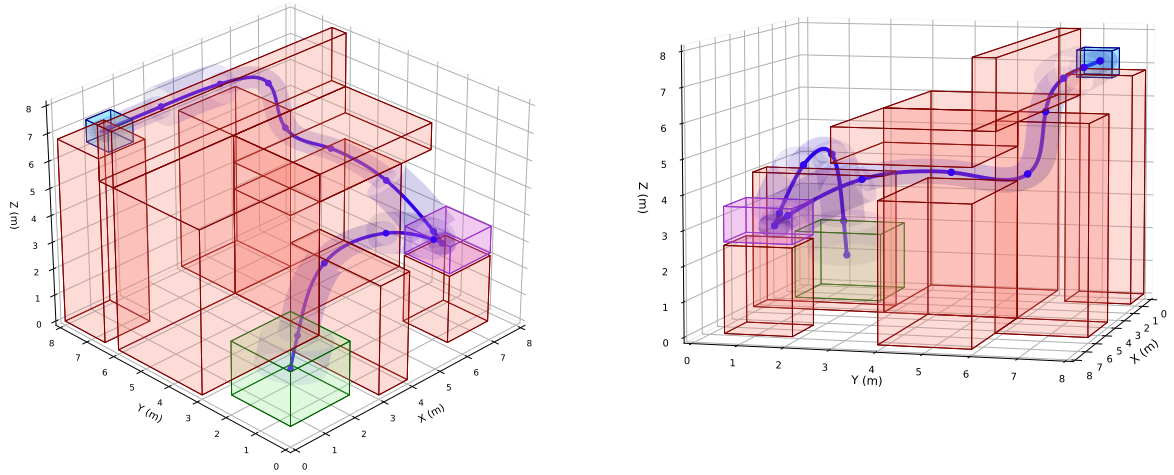
Planning Scenario

We consider a reach-avoid task in a constrained environment with a prescribed green initial region, a blue target region, a purple holding region, and seven red obstacle regions, as shown in Figure 4.3. The task is encoded by the following continuous-time STL specification:

$$\varphi = \left(\square_{[0,T]}\neg\mathbf{R}\right) \wedge \left(\lozenge_{[0,T]}\mathbf{B}\right) \wedge \left(\lozenge_{[0,T-2]}\square_{[0,2]}\mathbf{P}\right), \tag{4.7.1}$$

where $T = 20$. Here, $\mathbf{R}$ denotes the union of red obstacle regions, $\mathbf{B}$ denotes the blue target region, and $\mathbf{P}$ denotes the purple holding region. This specification requires the trajectory to avoid all obstacles throughout the entire time horizon, remain inside the holding region continuously for at least two seconds after entering it, and eventually reach the target region within $T = 20$ seconds.

Furthermore, the componentwise acceleration bounds are chosen consistently with b_a as $a_{\max} = [1, 1, 11]$, where the larger third component accounts for gravity, with $g = 9.81$. The desired trajectory, shown as the blue curves in Figure 4.3, consists of fourteen Bézier curve segments, with the corresponding time-varying robustness ρ^{φ} highlighted in blue tubes.



(a) Front-left perspective view.

(b) Rear-right perspective view.

Figure 4.3: Environment setup and desired trajectory: The blue curve denotes the desired trajectory parameterized by fourteen Bézier curve segments, with the dots marking the segment endpoints for clarity. The blue tube represents the robustness tube of the Bézier curves.

Tracking Evaluation Setup

To validate the tracking performance around the planned desired trajectory, we independently sample the initial perturbations and define the initial tracking states as follows:

$$p(0) = p_d(0) + \delta_p, \qquad v(0) = \dot{p}_d(0) + \delta_v,$$
$$R(0) = R_d(0)\exp(\hat{\mathfrak{r}}_0)\,, \qquad \omega(0) = \omega_d(0) + \delta_\omega.$$

The perturbations are sampled independently as

$$\delta_p \in [-0.5, 0.5]^3, \qquad \delta_v \in [-0.2, 0.2]^3, \qquad \mathfrak{r}_0 \in [-0.1, 0.1]^3, \qquad \delta_\omega \in [-0.1, 0.1]^3.$$

In the simulations, we choose $R_d(0) = I_3$ and $\omega_d(0) = 0$, so that $R(0) = \exp(\hat{\mathfrak{r}}_0)$, $\omega(0) = \delta_\omega$.

As presented in Figure 4.4, the tracking trajectories from these one hundred initial points satisfying the initial-condition constraints in (4.5.29)–(4.5.31) remain within the

safe tube and converge to the target set. The trajectories are computed in Python using `solve_ivp` with the `RK45` method and sampled at $dt = 0.01$.

Using the same one hundred initial states, we further compare the proposed geometric control approach with diagonal matrix gains (DMG-GC) against three baselines: standard geometric control with real gains (GC) [127], a CBF-QP [44], and nonlinear model predictive control (NMPC) on SE(3) [35, 121]. The CBF-QP and NMPC baselines are implemented in `CasADi` [5].

For the CBF-QP baseline, we consider two choices of the safety margin ϵ[2]:

i) $\epsilon_{i,p}(0)$, the initial position error of the i-th trajectory, which is at most 0.2 due to the initial sampling range;

ii) ϵ_φ, chosen according to the minimum robustness margin $\min_{t\in\mathbb{T}} \rho^\varphi(t)$ attained by the Bézier desired trajectory. Since Problem 4.6.1 requires the STL robustness margin to be no smaller than the tracking position-error bound, ϵ_φ satisfies $\epsilon_\varphi \geq \epsilon_{i,p}(0)$ for all $i \in \{1, \ldots, 100\}$.

According to the safety analysis in [44], choosing $\epsilon_{i,p}(0)$ ensures forward invariance of the safe tube. We also include $\epsilon = \epsilon_\varphi$ to provide a more comprehensive comparison, since the nominal trajectory exhibits both intermediate safety risks and strict terminal reachability requirements.

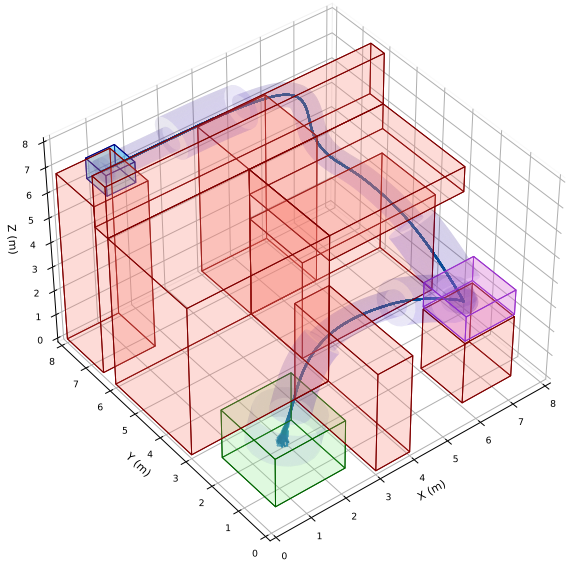
For the NMPC baseline, we evaluate prediction horizons $H_p \in \{2, 10, 15, 20, 25\}$ steps, where $H_p = 2$ is the smallest horizon that yields a feasible solution, while the larger horizons are included to examine the effect of increasing look-ahead on tracking performance and computational cost.

For the NMPC baseline, we evaluate prediction horizons $H_p \in \{2, 10, 15, 20, 25\}$ steps, where $H_p = 2$ is the minimum horizon for which the NMPC problem is feasible, and the larger horizons provide progressively longer horizons for evaluating the trade-off between tracking performance and online computational effort.

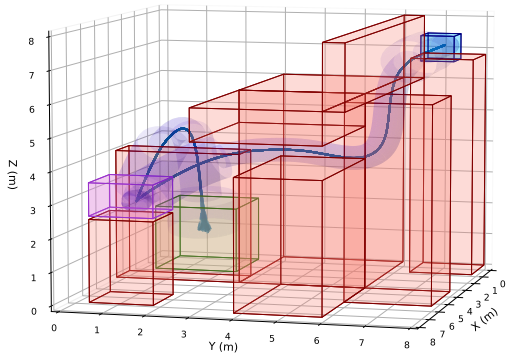
Runtime and Tracking Error Analysis

For the ith trajectory, let $e_{i,p}(t)$ and $e_{i,v}(t)$ denote the position and velocity tracking errors, respectively. To evaluate the worst-case tracking performance across the 100

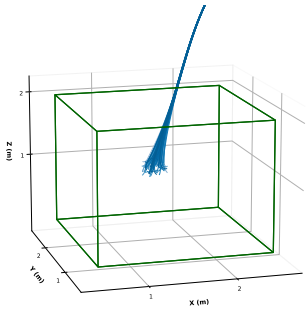
[2]Here ϵ refers to the parameter δ in [44], which denotes the radius of the safe tube enforced by the CBF. We use ϵ to avoid confusion with notation used elsewhere in this thesis. We also remark that the dynamics model in [44] is a reduced-order model and is slightly different from the full quadrotor dynamics considered here.



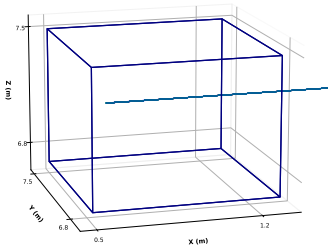
(a) Front-left perspective view.



(b) Rear-right perspective view.



(c) Initial set.



(d) Target set.

Figure 4.4: Environment and Tracking Trajectories. The curves denote tracking trajectories generated from perturbed initial conditions.

trajectories at each time t, define

$$e_p^{\max}(t) := \max_{i\in\{1,\dots,100\}} \|e_{i,p}(t)\|, \qquad e_v^{\max}(t) := \max_{i\in\{1,\dots,100\}} \|e_{i,v}(t)\|. \tag{4.7.2}$$

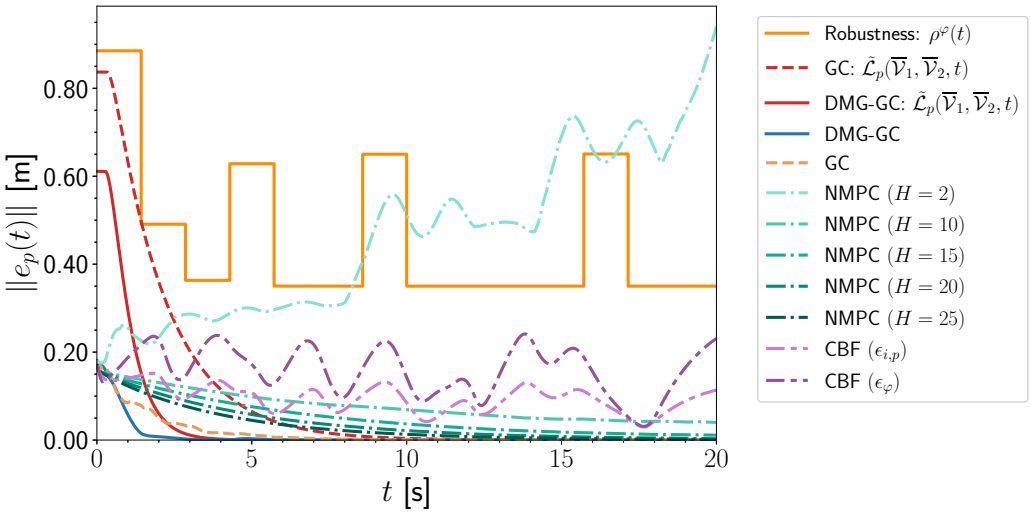
Figure 4.5 compares these pointwise maximum position and velocity tracking errors. The proposed DMG-GC approach keeps both errors within their theoretical bounds over the entire time horizon, confirming Corollary 4.5.3. Although the tracking errors under the standard GC approach also remain below the corresponding bounds, they decay more slowly than those under DMG-GC. Moreover, Figure 4.5(a) shows that the theoretical position-error bound $\tilde{\mathcal{L}}_p(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t)$ remains below the STL robustness margin of the desired trajectory. Thus, satisfaction of the STL specification is preserved despite the initial tracking errors. The position errors under NMPC and CBF-QP also remain below the robustness margin ρ^{φ}. For some parameter choices, such as a longer prediction horizon or a smaller safety margin, these position errors can be further reduced, but at the expense of sharper variations in the velocity errors, as shown in Figure 4.5(b).

In particular, the CBF-QP results require some additional interpretation. While using a smaller robustness margin ϵ in CBF-QP may reduce tracking errors, it introduces trade-offs. Since the initial conditions no longer lie in an ϵ-tube of the reference trajectory, the analysis in [44] provides no guarantee of remaining inside a safety tube. Achieving acceptable ultimate tracking error also requires tuning the gains (a_1 and a_2 in [44]). For the errors reported in Table 4.3, we had to select aggressive gains $a_1 = 36$ and $a_2 = 18$. These gains cause large initial velocity errors, as shown in Figure 4.5(b), and demand higher early-stage velocity and thrust, as shown in Figure 4.6. This behavior indicates that CBF-QP pushes the system toward much smaller admissible invariant sets, increasing the burden on the dynamics and hardware.

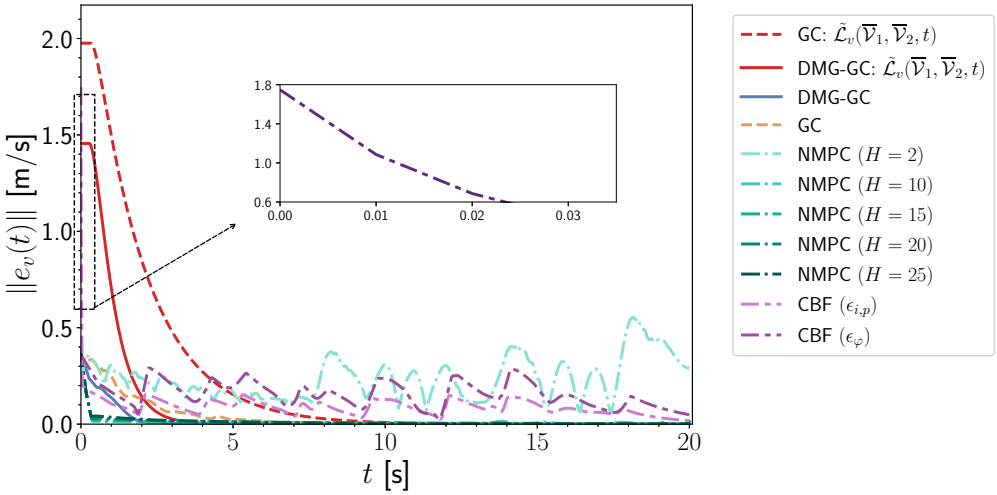
For the subsequent quantitative analysis, Table 4.3 reports the maximum tracking position and velocity errors for $t \geq 4\,\mathrm{s}$, after the errors for all approaches have largely settled, thereby avoiding bias from the initial error. Specifically, the reported maximum errors are defined as

$$\begin{aligned} e_p^{\max} &:= \max_{t\in[4,T]} \max_{i\in\{1,\dots,100\}} \|e_{i,p}(t)\|\,, \\ e_v^{\max} &:= \max_{t\in[4,T]} \max_{i\in\{1,\dots,100\}} \|e_{i,v}(t)\|\,. \end{aligned} \tag{4.7.3}$$

The STL robustness margin used for comparison is $\min_{t\in\mathbb{T}} \rho^{\varphi}(t) = 0.35$ m. DMG-GC achieves the lowest position and velocity errors among all evaluated controllers. Table 4.3 also compares the mean runtime among all controllers. The CBF-QP implementations are the fastest, partly because they are based on the simplified differential-flat quadrotor model used in [44]. GC is slightly faster than DMG-GC since it uses scalar gains rather than



(a) Position errors and bounds.



(b) Velocity errors and bounds.

Figure 4.5: Position and velocity tracking errors with theoretical bounds. Panel (a) presents the position tracking errors of DMG-GC, GC, NMPC, and CBF-QP, together with the theoretical position-error bounds for DMG-GC and GC and the STL robustness margin of the desired trajectory. Panel (b) presents the corresponding velocity tracking errors and the theoretical velocity-error bounds for DMG-GC and GC.

diagonal matrix gains, while the runtime difference between GC and DMG-GC remains small. In contrast, NMPC is slower, with its runtime increasing as the prediction horizon H_p increases. Although using the minimum feasible horizon $H_p = 2$ reduces the runtime relative to the longer horizons, it results in larger position and velocity tracking errors, illustrating the trade-off between tracking performance and computational effort.

Table 4.3: Comparison of tracking runtime and maximum position and velocity tracking errors across 100 trajectories. Tracking errors are evaluated for $t \geq 4\,\mathrm{s}$, and runtime is reported as mean $\pm$ standard deviation.

Controller		Mean Runtime (s)	$e_p^{\max}$ (mm)	$e_v^{\max}$ (mm/s)
DMG-GC		0.31 ± 0.01	**3.25**	**13.21**
GC		0.28 ± 0.04	17.52	45.54
CBF-QP	$\epsilon_i = e_{i,p}(0)$	0.25 ± 0.01	134.63	152.15
	$\epsilon_\varphi = \min_{t \in \mathbb{T}} \rho^\varphi(t)$	$\mathbf{0.24 \pm 0.01}$	241.22	286.62
NMPC	$H_p = 2$ (steps)	5.74 ± 0.12	939.73	553.89
	$H_p = 10$ (steps)	10.15 ± 0.07	108.09	14.81
	$H_p = 15$ (steps)	14.01 ± 0.34	87.99	19.48
	$H_p = 20$ (steps)	16.82 ± 0.14	71.06	20.83
	$H_p = 25$ (steps)	20.16 ± 0.13	57.74	19.79

Bound Satisfaction and Lyapunov Profiles

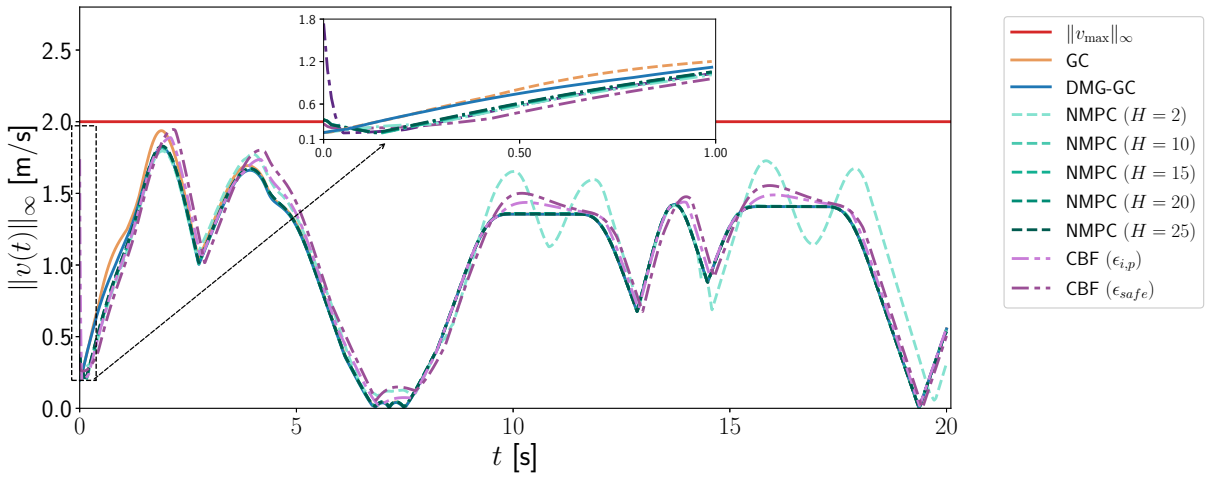
Figure 4.6 and Figure 4.7 verify the satisfaction of the velocity and thrust constraints, as well as the positivity condition on the third component of the desired force. As shown in Figure 4.6(a), all methods satisfy the prescribed velocity bound over the entire time horizon. The proposed DMG-GC approach maintains a smooth velocity response, with smaller velocity magnitudes than GC and CBF-QP, while its response remains relatively close to that of NMPC. In particular, GC produces relatively larger velocities in the later stage, while CBF-QP with a smaller safety margin $\epsilon_{i,p}(0)$ exhibits a sharper initial transient and stays closer to the velocity limit.

The thrust bound $\overline{\mathcal{F}}$, shown as the horizontal red line in Figure 4.6(b), is briefly derived as follows. From (4.3.19)–(4.3.20), together with (4.3.1) in Assumption 4.3.1, and since $R(t) \in \mathrm{SO}(3)$, we have $\|R(t)e_3\| = 1$. Hence, by the Cauchy–Schwarz and triangle inequalities,

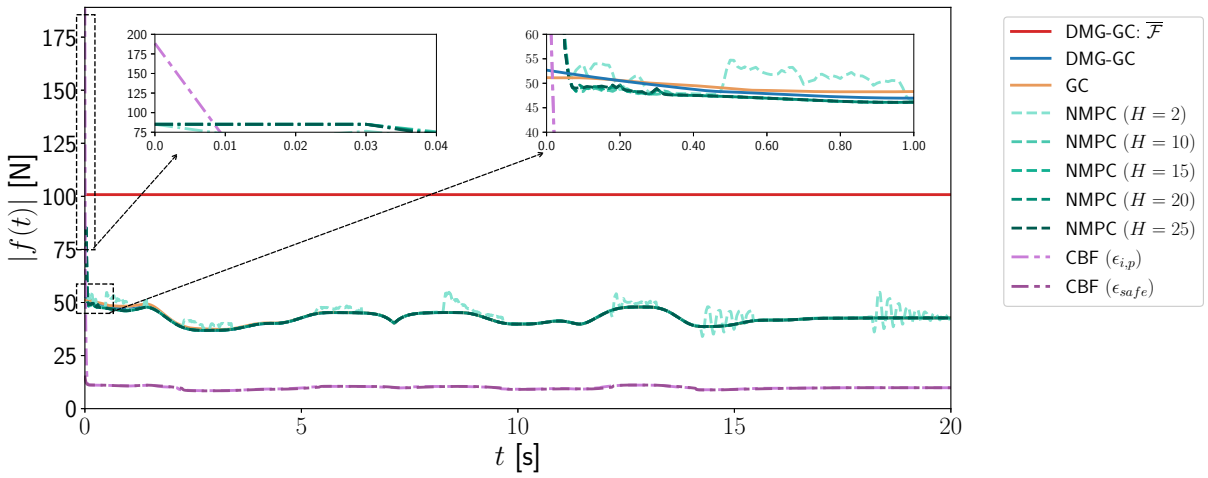
$$\begin{aligned} |f(t)| &\le \|F_d(t)\| \\ &\le \left\| [K_p \quad K_v]\, M_1^{-\frac{1}{2}} \right\| \mathcal{L}_1(\overline{\mathcal{V}}_1, \overline{\mathcal{V}}_2, t) + m\|b_a\| := \overline{\mathcal{F}}. \end{aligned} \tag{4.7.4}$$

As shown in Figure 4.6(b), the thrust generated by DMG-GC and GC remains strictly below $\overline{\mathcal{F}}$. In contrast, CBF-QP with a smaller safety margin produces a large initial thrust peak that exceeds $\overline{\mathcal{F}}$, indicating a loss of physical feasibility during the transient phase. This behavior is consistent with the sharper velocity-error variations observed in Figure 4.5(b), which lead to more aggressive control commands. The NMPC results remain below the thrust bound in this simulation, although their transient responses depend on the selected prediction horizon.

Finally, Figure 4.7(a) shows that $F_{d,3}(t)$ is positive throughout the simulations for all sampled trajectories over the time interval considered. Hence, along these sampled trajectories, the conditions (4.3.28)–(4.3.29) are satisfied in the simulations, and the desired force direction is well defined. Figure 4.7(b) shows that there exists a constant ψ such that $\Psi_K(t) < \psi$ for all sampled trajectories over $t \in [0, 20]$. Moreover, $\Psi_K(t)$ converges to a small neighborhood of zero after approximately $t = 0.4$ s. Together with the sampled initial condition $\Psi_K(0) \le \psi$, these numerical results indicate that the assumptions required by Proposition 4.3.3 are satisfied along the simulated trajectories.

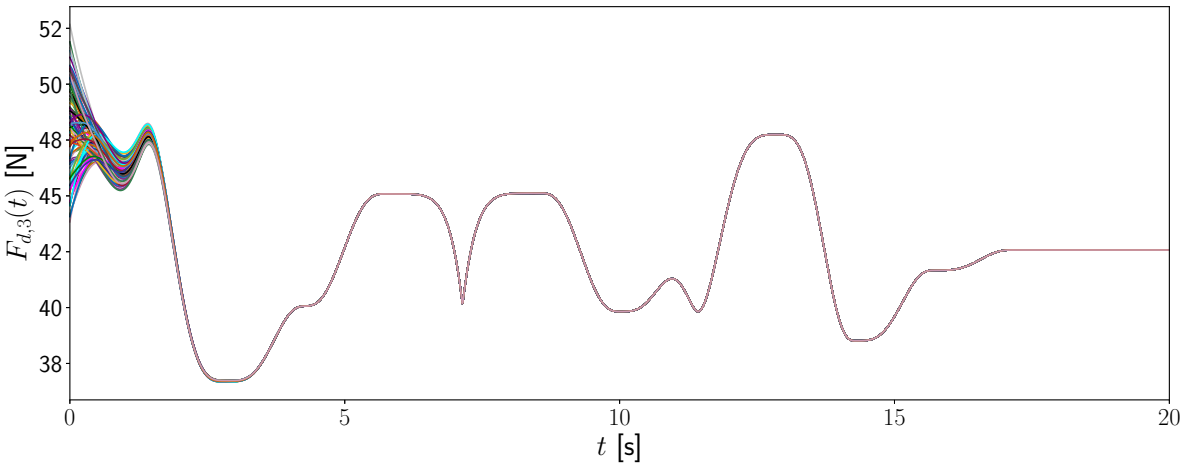


(a) Maximum velocity: $\max_{i\in\{1,\dots,100\}} \|v_i(t)\|_\infty$.

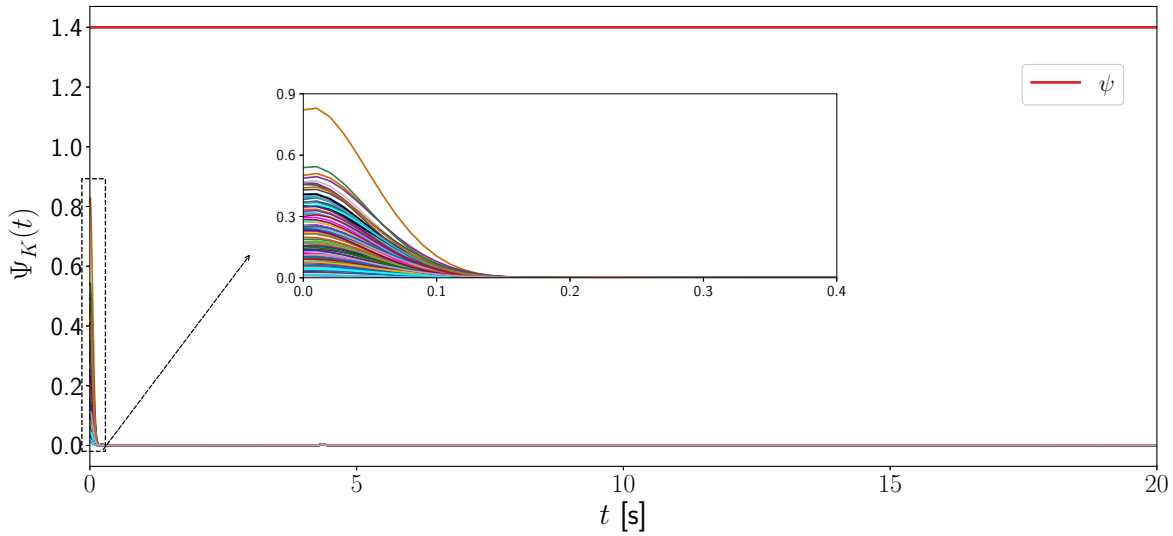


(b) Maximum thrust: $\max_{i\in\{1,\dots,100\}} |f_i(t)|$.

Figure 4.6: Constraint satisfaction across 100 trajectories: $\max_{j\in\{1,\dots,100\}} \|v_j(t)\|_\infty$ and $\max_{j\in\{1,\dots,100\}} |f_j(t)|$ for $t \in [0, T]$.



(a) $F_{d,3}(t)$.



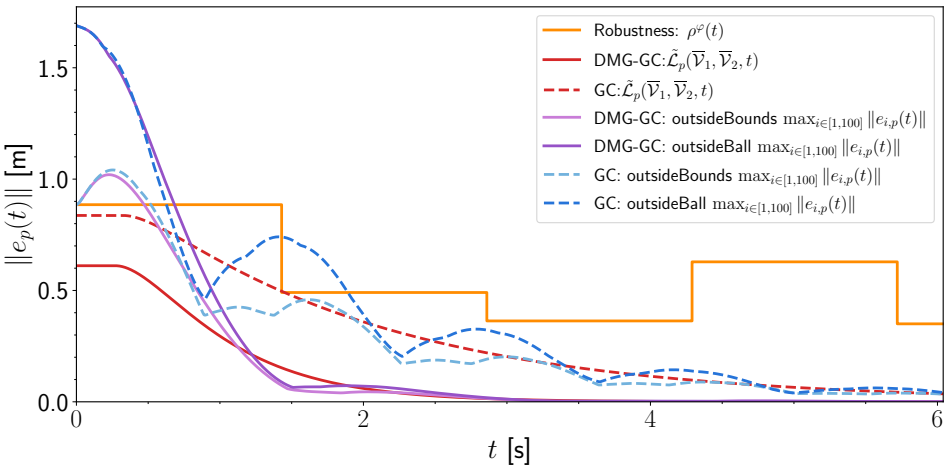
(b) $\Psi_K(t)$.

Figure 4.7: Condition satisfaction for 100 trajectories: $F_{d,3}(t)$ and $\Psi_K(t)$ over $t \in [0, 20]$.

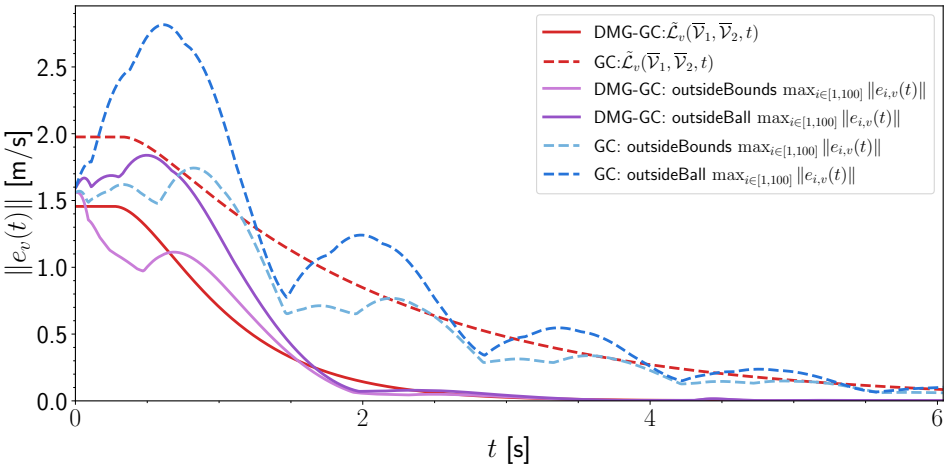
Adversarial Initial Conditions

Figure 4.8 examines two types of adversarial initial conditions to compare the robustness of the proposed DMG-GC controller with the standard GC controller. The first category consists of 100 initial points that violate the respective initial-condition requirements of both controllers. For DMG-GC, these requirements correspond to (4.5.29)–(4.5.31). Consequently, the resulting trajectories may exceed their respective theoretical error bounds, including the DMG-GC bounds in (4.5.37)–(4.5.38). The second category consists of 100 initial points that lie outside the STL robustness tube at the initial time $t = 0$, thereby also violating the respective initial-condition requirements of both controllers. The tracking trajectories generated by the proposed DMG-GC controller for the two categories are shown in Figure 4.9.

Since these adversarial initial conditions violate the prescribed sufficient conditions, the formal guarantees are not directly applicable in either case. Nevertheless, as shown in Figure 4.8, the DMG-GC errors decay rapidly and converge to zero, whereas the standard GC baseline exhibits larger transient peaks and slower decay. Although the theoretical initial-condition constraints are designed to provide formal certification within a prescribed region, the rapid and stable error decay of DMG-GC allows trajectories with large initial errors to return quickly to the STL robustness tube, as shown in Figure 4.9.

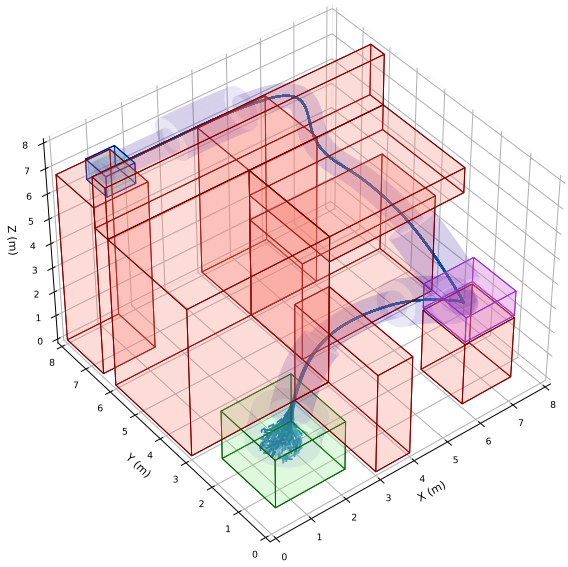


(a) Position errors: $\|e_{i,p}(t)\|$.

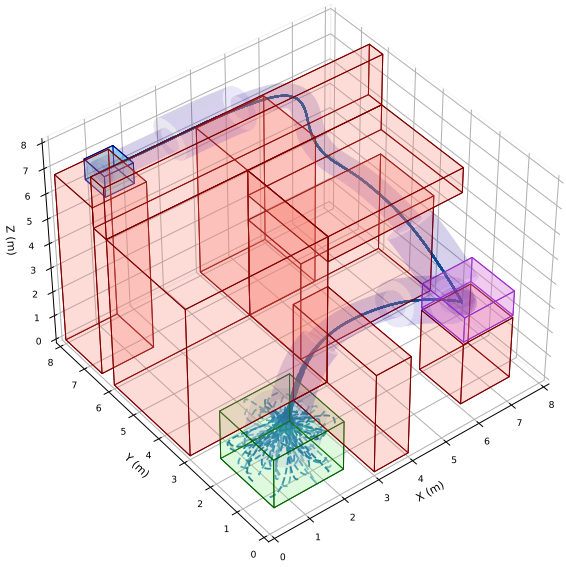


(b) Velocity errors: $\|e_{i,v}(t)\|$.

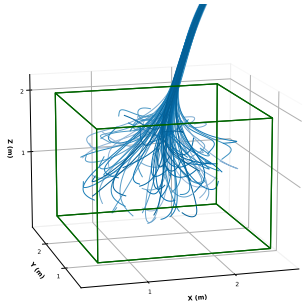
Figure 4.8: Robustness under adversarial initial conditions: Position and velocity tracking errors for initial points violating the theoretical conditions and for initial points outside the initial safe set. The plots are restricted to $t \leq 6$ to better show the transient behavior.



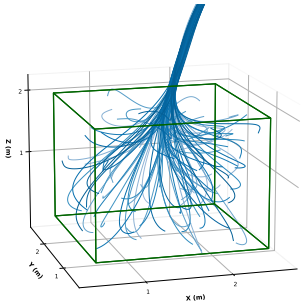
(a) Outside error bounds: trajectories.



(b) Outside STL tube: trajectories.



(c) Outside error bounds: initial points.



(d) Outside STL tube: initial points.

Figure 4.9: DMG-GC results under adversarial initial conditions: Tracking trajectories and initial-point distributions under two types of adversarial initial conditions. The two cases correspond to initial points outside the theoretical error bounds and outside the STL robustness tube at the initial time $t = 0$, respectively.

4.8 Summary

This chapter presented a continuous-time control framework for quadrotor systems under STL specifications by integrating MICP-based Bézier trajectory generation with geometric tracking control. The proposed approach enables safe and dynamically feasible motion planning while providing continuous-time tracking-error bounds. The theoretical conditions ensure that the position and velocity tracking errors remain within prescribed limits, and the resulting velocity, thrust, and desired-force positivity conditions support the corresponding feasibility constraints. The simulation results further demonstrate that the proposed DMG-GC method achieves better tracking performance and tighter tracking-error bounds than conventional scalar-gain geometric control and maintains the required physical bounds over the entire time horizon. Compared with CBF-QP and NMPC baselines, the proposed framework exhibits improved constraint satisfaction and less aggressive control behavior, thereby demonstrating the effectiveness of the matrix-gain design and the overall control framework.

Building on these results, the next chapter extends the continuous-time STL Bézier planning framework to multi-agent systems. In particular, the focus will shift from single-quadrotor tracking and feasibility analysis to scalable trajectory generation for multiple interacting agents under STL tasks. The multi-agent extension will investigate how to reduce the computational burden of centralized MICP formulations while preserving continuous-time specification satisfaction and inter-agent safety constraints. This provides a natural step toward applying the proposed framework to coordinated missions involving multiple quadrotors.

Chapter 5

Continuous-Time STL Control Synthesis for Multiple Quadrotors

5.1 Introduction

This chapter integrates continuous-time STL planning with tracking-error guarantees for multi-quadrotor systems. In contrast to the single-quadrotor setting considered in the previous chapter, multi-agent tasks require reasoning about both individual temporal requirements and inter-agent coordination.

To improve scalability in multi-agent settings, a backward-recursive structure is introduced into the STL encoding, allowing satisfaction and robustness information to be propagated across time intervals more efficiently. This is particularly important for multi-agent problems, where the number of binary variables and constraints grows with the number of agents, time intervals, and temporal operators.

The main contributions of this chapter are summarized as follows:

- A multi-agent continuous-time STL planning framework is formulated for quadrotor systems operating in constrained environments, incorporating prescribed tracking-error bounds so that the tracked trajectories satisfy the required safety and temporal-logic specifications.
- A backward-recursive structure is incorporated into the MICP-based STL encoding to improve the efficiency of multi-agent trajectory synthesis. The proposed framework is validated through numerical simulations involving obstacle avoidance, target reaching, and cooperative multi-agent tasks.

5.2 Problem Statement

Building on Problem 4.6.1 in Chapter 4, this chapter extends the STL-constrained trajectory generation framework from a single quadrotor to a multi-quadrotor system. Given an STL specification φ, the objective is to synthesize reference trajectories y_d^ℓ and geometric feedback controllers for N_ℓ quadrotors such that the actual tracking trajectory y_{tr}^ℓ of each quadrotor $\ell \in \{1, \ldots, N_\ell\}$ satisfies φ.

We apply the tracking-error envelope constructed in Chapter 4 to each quadrotor. The lower bound imposed on the STL robustness margin is defined as

$$\gamma(t) := \Gamma(t) + \gamma_c, \qquad \forall\, t \in \mathbb{T}, \tag{5.2.1}$$

where $\gamma_c > 0$ is the physical safety margin and $\Gamma(t)$ is the non-increasing position tracking-error envelope. For clarity of exposition, we assume that all quadrotors have the same physical parameters and use the same geometric controller gains. Therefore, the same envelope $\Gamma(t)$ is used for all quadrotors, and for each $\ell \in \{1, \ldots, N_\ell\}$,

$$\left\| y_{\text{tr}}^\ell(t) - y_d^\ell(t) \right\| \le \Gamma(t), \qquad \forall t \in \mathbb{T}. \tag{5.2.2}$$

If the desired trajectory y_d^ℓ satisfies the STL specification φ with robustness margin $\rho^{\ell,\varphi}(t) \ge \gamma(t)$, then $\Gamma(t) < \gamma(t) \le \rho^{\ell,\varphi}(t)$ since $\gamma_c > 0$. Hence,

$$\left\| y_{\text{tr}}^\ell(t) - y_d^\ell(t) \right\| \le \Gamma(t) < \rho^{\ell,\varphi}(t), \qquad \forall t \in \mathbb{T}, \tag{5.2.3}$$

which implies $y_{\text{tr}}^\ell \models \varphi$. Therefore, the main task in this chapter is to generate Bézier-parameterized reference trajectories for all quadrotors that satisfy the required STL robustness lower bound while additionally enforcing inter-agent safety, smoothness, and the kinematic constraints required by the geometric controller. If the quadrotors have different physical parameters or controller gains, Problem 4.5.4 in Chapter 4 can be solved separately for each quadrotor to obtain agent-specific tracking-error envelopes $\Gamma^\ell(t)$ and the corresponding lower bounds $\gamma^\ell(t) = \Gamma^\ell(t) + \gamma_c$ for all $t \in \mathbb{T}$.

The following multi-quadrotor STL-constrained trajectory generation problem extends the formulation in Problem 4.6.1.

Problem 5.2.1. *Let $z_k^\varphi \in \{0, 1\}$ denote the top-level Boolean variable indicating whether the Bézier reference trajectories jointly satisfy the multi-agent STL formula φ at index k. The MICP is formulated to maximize the STL robustness while minimizing the motion*

effort of the reference trajectories:

$$
\begin{aligned}
\underset{\mathcal{C}_k^\ell, v_k^\ell, a_k^\ell, r_k^\ell, \epsilon_k^\ell, z_k^{\ell,\varphi}}{\arg\min} \quad & \sum_{\ell=1}^{N_\ell} \sum_{k=0}^{N-1} \left(-\mathcal{W} \rho_k^{\ell,\varphi}(\mathcal{C}_k^\ell, a_k^\ell) + \mathcal{Q}\|v_k^\ell\|_1 + \mathcal{R}\|a_k^\ell\|_1 \right) \\
\textit{subject to} \quad & B_k^\ell \textit{ and } B_{k+1}^\ell \textit{ are joined with } \mathcal{C}^4\textit{-continuity}, \\
& B_k^\ell \textit{ satisfies the kinematic constraints}, \\
& B_k^i \textit{ and } B_k^j \textit{ are collision-free}, \quad 1 \le i < j \le N_\ell, \\
& \rho_k^{\ell,\varphi}\left(\mathcal{C}_k^\ell, a_k^\ell\right) \ge \gamma(t_k), \\
& z_0^\varphi = 1,
\end{aligned}
\tag{5.2.4}
$$

where $\mathcal{W}, \mathcal{Q}, \mathcal{R} \in \mathbb{R}_{++}$ are user-defined weights. For all $\ell \in \{1, \dots, N_\ell\}$, the continuity constraints are enforced for $k \in \{0, \dots, N-2\}$, whereas the kinematic, robustness, and STL encoding constraints are enforced for $k \in \{0, \dots, N-1\}$.

5.3 Encoding STL Satisfaction for Multi-Agent Systems

This section presents an MICP encoding of the multi-agent STL specification φ over piecewise Bézier reference trajectories.

Recursive STL Encoding over Bézier Curves

The STL encoding of Bézier-curve robustness from [155] is adopted and extended to the multi-agent setting by introducing an agent index ℓ. As stated in Problem 5.2.1, each segment B_k^ℓ has the robustness measure $\rho_k^{\ell,\varphi}\left(r_k^\ell, \epsilon_k^\ell\right) = r_k^\ell - \epsilon_k^\ell \ge \gamma(t_k)$. Here, $r_k^\ell \in \mathbb{R}_+$ specifies the margin ensuring that the endpoints of the curve lie strictly within the required set, while $\epsilon_k^\ell \in \mathbb{R}_+$ is the margin used to constrain the intermediate control points within the same set under the acceleration constraints introduced in Chapter 3.

Throughout this chapter, $t_k = k\Delta t$ denotes the k-th sampling instant. The Boolean variable associated with an STL formula at index k denotes satisfaction of the formula at t_k under the discrete-time STL abstraction. Although the atomic-predicate constraints below are imposed over the entire k-th Bézier segment, they provide sufficient conditions for satisfaction of the corresponding predicates at t_k.

Given $\mathcal{I} = [a, b]$ and the finite interval $\widetilde{\mathcal{I}} = (t_k + \mathcal{I}) \cap \mathbb{T}$, define $k_l = k + \left\lfloor \frac{a}{\Delta t} \right\rfloor$ and $= \min\left(k + \left\lfloor \frac{b}{\Delta t} \right\rfloor, N-1\right)$ with $k_l \le k_r$. The indices k_l and k_r are the left and right discrete

indices associated with the finite temporal window, respectively. The encoding presented in Chapter 3 is extended to the multi-agent formulation as follows.

For an STL formula φ, let $\mathsf{A}_\varphi \subseteq \{1, \ldots, N_\ell\}$ denote the set of agent indices appearing in φ. The agent-index set is defined recursively. For an atomic predicate π associated with quadrotor ℓ and its negation $\neg\pi$, define $\mathsf{A}_\pi = \mathsf{A}_{\neg\pi} = \{\ell\}$. For Boolean and Until formulas, define $\mathsf{A}_{\varphi_1 \wedge \varphi_2} = \mathsf{A}_{\varphi_1 \vee \varphi_2} = \mathsf{A}_{\varphi_1 \mathcal{U}_\mathcal{I} \varphi_2} = \mathsf{A}_{\varphi_1} \cup \mathsf{A}_{\varphi_2}$. For the unary temporal operators, $\mathsf{A}_{\Box_\mathcal{I}\varphi} = \mathsf{A}_{\Diamond_\mathcal{I}\varphi} = \mathsf{A}_\varphi$. For the atomic cases, the single-agent index ℓ is retained in the encoding variables for clarity. At the top level, the Boolean variable associated with the complete formula φ is abbreviated as $z_k^\varphi := z_k^{\mathsf{A}_\varphi, \varphi}$.

- **Predicate** π:

$$z_k^{\ell,\pi} = \left(\bigwedge_{i=1}^{\mathrm{Row}(H)} \left(\bigwedge_{j \in \{0,n\}} \frac{b_i - H_i c_{k,j}^\ell}{\|H_i\|} \geq r_k^\ell \right) \right) \wedge z_{\epsilon_k^\ell}. \tag{5.3.1}$$

- **Negation** $\neg\pi$:

$$z_k^{\ell,\neg\pi} = \left(\bigvee_{i=1}^{\mathrm{Row}(H)} \left(\bigwedge_{j \in \{0,n\}} \frac{H_i c_{k,j}^\ell - b_i}{\|H_i\|} \geq r_k^\ell \right) \right) \wedge z_{\epsilon_k^\ell}. \tag{5.3.2}$$

Here,

$$z_{\epsilon_k^\ell} := \left(\bigwedge_{j \in \{1,\ldots,n\}} \left| c_{k,j}^\ell - c_{k,j-1}^\ell \right| \leq \frac{a_k^\ell \Delta t^2}{2n} \right) \wedge \left(\bigwedge_{j=1}^{d} \frac{8\epsilon_k^\ell}{3\sqrt{d}\Delta t^2} - \left| a_{k,j}^\ell \right| \geq 0 \right). \tag{5.3.3}$$

The Boolean satisfaction conditions for conjunction, disjunction, Always, and Eventually are stated first. The Until operator is implemented using an equivalent backward-recursive mixed-integer encoding.

- **Conjunction** $\varphi = \varphi_1 \wedge \varphi_2$:

$$z_k^{\mathsf{A}_\varphi, \varphi} = z_k^{\mathsf{A}_{\varphi_1}, \varphi_1} \wedge z_k^{\mathsf{A}_{\varphi_2}, \varphi_2}, \qquad \mathsf{A}_\varphi = \mathsf{A}_{\varphi_1} \cup \mathsf{A}_{\varphi_2}. \tag{5.3.4}$$

- **Disjunction** $\varphi = \varphi_1 \vee \varphi_2$:

$$z_k^{\mathsf{A}_\varphi, \varphi} = z_k^{\mathsf{A}_{\varphi_1}, \varphi_1} \vee z_k^{\mathsf{A}_{\varphi_2}, \varphi_2}, \qquad \mathsf{A}_\varphi = \mathsf{A}_{\varphi_1} \cup \mathsf{A}_{\varphi_2}. \tag{5.3.5}$$

- **Always** $\varphi = \square_{\mathcal{I}}\varphi_1$:

$$z_k^{\mathsf{A}_\varphi,\varphi} = \bigwedge_{k'=k_l}^{k_r} z_{k'}^{\mathsf{A}_{\varphi_1},\varphi_1}, \qquad \mathsf{A}_\varphi = \mathsf{A}_{\varphi_1}. \tag{5.3.6}$$

- **Eventually** $\varphi = \Diamond_{\mathcal{I}}\varphi_1$:

$$z_k^{\mathsf{A}_\varphi,\varphi} = \bigvee_{k'=k_l}^{k_r} z_{k'}^{\mathsf{A}_{\varphi_1},\varphi_1}, \qquad \mathsf{A}_\varphi = \mathsf{A}_{\varphi_1}. \tag{5.3.7}$$

- **Until** $\psi = \varphi_1 \mathcal{U}_{\mathcal{I}} \varphi_2$: For a fixed evaluation index k, let $q \in \{k, \dots, k_r\}$ denote the backward-recursion index, and define $\alpha_q := z_q^{\mathsf{A}_{\varphi_1},\varphi_1}$ and $\beta_q := z_q^{\mathsf{A}_{\varphi_2},\varphi_2}$. Introducing $u_{k,q}^{\psi} \in \{0,1\}$, the Until operator is encoded as

$$\begin{aligned}
u_{k,k_r}^{\psi} &= \alpha_{k_r} \wedge \beta_{k_r}, \\
u_{k,q}^{\psi} &= \alpha_q \wedge \left(\beta_q \vee u_{k,q+1}^{\psi}\right), && \text{if } q = k_r - 1, \dots, k_l, \\
u_{k,q}^{\psi} &= \alpha_q \wedge u_{k,q+1}^{\psi}, && \text{if } q = k_l - 1, \dots, k, \\
z_k^{\mathsf{A}_\psi,\psi} &= u_{k,k}^{\psi}, \qquad \mathsf{A}_\psi = \mathsf{A}_{\varphi_1} \cup \mathsf{A}_{\varphi_2}.
\end{aligned} \tag{5.3.8}$$

Finally, the constraint $z_0^{\mathsf{A}_\varphi,\varphi} = 1$ is imposed to enforce satisfaction of the complete STL specification φ from initial time $t = 0$.

Lemma 5.3.1 (Soundness of the Multi-Agent STL Encoding)**.** *For any $k \in \{0, \dots, N-1\}$ and STL formula φ, if $z_k^{\mathsf{A}_\varphi,\varphi}$ is true, then the Bézier reference trajectories $\{y_d^\ell\}_{\ell \in \mathsf{A}_\varphi}$ jointly satisfy φ.*

Proof. The proof proceeds by structural induction on φ. For an atomic predicate π associated with quadrotor ℓ, or its negation $\neg\pi$, the corresponding constraints enforce the tightened predicate conditions on the k-th Bézier segment. Together with $z_{\epsilon_k^\ell}$ in (5.3.3), the convex-hull property guarantees that these conditions hold continuously for $t \in [t_k, t_{k+1}]$, as established in Chapter 3. Hence, the result holds for the atomic formulas.

Suppose that the result holds for the immediate subformulas. For $\varphi = \varphi_1 \wedge \varphi_2$, the encoding requires both subformulas to hold over the k-th Bézier segment. Together with $\mathsf{A}_\varphi = \mathsf{A}_{\varphi_1} \cup \mathsf{A}_{\varphi_2}$, the induction hypothesis establishes satisfaction of $\varphi_1 \wedge \varphi_2$. For $\varphi = \varphi_1 \vee \varphi_2$, at least one subformula holds over the k-th Bézier segment, and the induction hypothesis establishes satisfaction of $\varphi_1 \vee \varphi_2$.

For $\varphi = \Box_{\mathcal{I}}\varphi_1$, the encoding requires φ_1 to hold over every Bézier segment indexed by $k' \in \{k_l, \ldots, k_r\}$. The induction hypothesis therefore establishes the always condition. Similarly, for $\varphi = \Diamond_{\mathcal{I}}\varphi_1$, the encoding requires φ_1 to hold over at least one Bézier segment indexed by $k' \in \{k_l, \ldots, k_r\}$, which establishes the eventually condition.

Finally, let $\psi = \varphi_1 \mathcal{U}_{\mathcal{I}} \varphi_2$ and consider the backward recursion in (5.3.8). We first show by backward induction that, for every $q \in \{k_l, \ldots, k_r\}$, $u^{\psi}_{k,q} = 1$ implies that there exists $k' \in \{q, \ldots, k_r\}$ such that, for every $k'' \in \{q, \ldots, k'\}$, $\beta_{k'} = 1$ and $\alpha_{k''} = 1$.

At $q = k_r$, the claim follows from $u^{\psi}_{k,k_r} = \alpha_{k_r} \wedge \beta_{k_r}$. Suppose that it holds at $q+1$. If $u^{\psi}_{k,q} = 1$, then $\alpha_q = 1$ and either $\beta_q = 1$ or $u^{\psi}_{k,q+1} = 1$. In the first case, $k' = q$ is a valid witness. In the second case, the induction hypothesis provides a witness in $\{q+1, \ldots, k_r\}$, while $\alpha_q = 1$ extends the required satisfaction of φ_1 to the segment indexed by q.

For $q = k_l - 1, \ldots, k$, the prefix recursion $u^{\psi}_{k,q} = \alpha_q \wedge u^{\psi}_{k,q+1}$ requires φ_1 to hold over every segment preceding the admissible witness window. Consequently, $u^{\psi}_{k,k} = 1$ implies that there exists a witness segment $k' \in \{k_l, \ldots, k_r\}$ over which φ_2 holds, while φ_1 holds over every segment indexed from k through k', inclusive. Since $z_k^{\mathsf{A}_\psi, \psi} = u^{\psi}_{k,k}$, the Until condition is satisfied.

Therefore, by structural induction, if $z_k^{\mathsf{A}_\varphi, \varphi}$ is true, then the Bézier reference trajectories $\{y_d^\ell\}_{\ell \in \mathsf{A}_\varphi}$ jointly satisfy φ when evaluated from the k-th Bézier segment. □

Encoding Multi-Agent Collision Avoidance

To ensure safety guarantees between any two agents i and j (where $1 \le i < j \le N_\ell$),the distance between the Bézier curves $B_k^i(t)$ and $B_k^j(t)$ is required to remain at least $\epsilon_{\text{inter},k} = 2\Gamma(t_k) + \gamma_c$ for all $t \in [t_k, t_{k+1}]$. For simplicity, $\mathcal{CH}_k^i$ and ϵ_k^i are grouped into the tuple $A_k^i = \left(\mathcal{CH}_k^i, \epsilon_k^i\right)$. Then, the multi-agent safety specification is defined as follows:

$$z_{\text{inter}} = \bigwedge_{1 \le i < j \le N_\ell} \text{safe}\left(A_k^i, A_k^j, \epsilon_{\text{inter},k}\right), \tag{5.3.9}$$

where

$$\begin{aligned} \text{safe}\left(A_k^i, A_k^j, \epsilon_{\text{inter},k}\right) &:= \left\| \text{cent}(\mathcal{CH}_k^i) - \text{cent}(\mathcal{CH}_k^j) \right\|_1 \\ &\ge \|c_{k,0}^i - c_{k,n}^i\|_1 + \|c_{k,0}^j - c_{k,n}^j\|_1 \\ &\quad + 2\sqrt{d}\left(\epsilon_k^i + \epsilon_k^j\right) + \epsilon_{\text{inter},k}\sqrt{d}. \end{aligned} \tag{5.3.10}$$

Here, $\text{cent}(\mathcal{CH}_k^i) = \frac{1}{n+1}\sum_{l=0}^{n} c_{k,l}^i$ is the centroid of the control points associated with the convex hull $\mathcal{CH}_k^i$.

The lower-bound constraint on the ℓ_1-norm in (5.3.10) defines a nonconvex feasible set and is therefore encoded using an exact mixed-integer sign disjunction in this work. Define

$$p_k^{ij} := \mathrm{cent}(\mathcal{CH}_k^i) - \mathrm{cent}(\mathcal{CH}_k^j), \qquad \mathcal{S} := \{-1, 1\}^d. \tag{5.3.11}$$

For each $s \in \mathcal{S}$, introduce a binary variable $\beta_{k,s}^{ij} \in \{0, 1\}$. Since $\|p_k^{ij}\|_1 = \max_{s\in\mathcal{S}} s^\top p_k^{ij}$, condition (5.3.10) is equivalently encoded by

$$\sum_{s\in\mathcal{S}} \beta_{k,s}^{ij} = 1, \qquad \beta_{k,s}^{ij} \in \{0, 1\}, \quad s \in \mathcal{S}, \tag{5.3.12}$$

and $\beta_{k,s}^{ij} = 1$ enforces

$$\beta_{k,s}^{ij} = 1 \quad \Longrightarrow \quad s^\top p_k^{ij} \geq D_k^{ij}, \qquad s \in \mathcal{S}, \tag{5.3.13}$$

where

$$D_k^{ij} := \|c_{k,0}^i - c_{k,n}^i\|_1 + \|c_{k,0}^j - c_{k,n}^j\|_1 + 2\sqrt{d}\left(\epsilon_k^i + \epsilon_k^j\right) + \epsilon_{\mathrm{inter},k}\sqrt{d}. \tag{5.3.14}$$

The endpoint norms in (5.3.14) can be represented using continuous auxiliary variables. For each $\ell \in \{i, j\}$ and $r \in \{1, \ldots, d\}$, introduce $u_{k,r}^\ell \geq 0$ satisfying

$$u_{k,r}^\ell \geq c_{k,0,r}^\ell - c_{k,n,r}^\ell, \qquad u_{k,r}^\ell \geq -\left(c_{k,0,r}^\ell - c_{k,n,r}^\ell\right). \tag{5.3.15}$$

Accordingly, the two endpoint norms in (5.3.14) are replaced by $\sum_{r=1}^d u_{k,r}^i$ and $\sum_{r=1}^d u_{k,r}^j$, respectively. Because these auxiliary variables appear on the right-hand side of a lower-bound separation constraint, this epigraph representation is exact in the projected feasible set.

For $d = 3$, the set $\mathcal{S}$ contains eight sign patterns; hence, eight sign-selection binary variables are introduced for each agent pair and each Bézier segment.

Lemma 5.3.2. *Given an STL specification φ, if the n-degree Bézier curves are generated by solving Problem 5.2.1 such that conditions (5.3.3) and (5.3.9) hold for all agent pairs (i, j) with $1 \leq i < j \leq N_\ell$, then for each segment $k \in \{0, \ldots, N-1\}$, the distance between any two Bézier curves satisfies*

$$\left\|B_k^i(t) - B_k^j(t)\right\| \geq \epsilon_{\mathrm{inter},k}, \quad \forall t \in [t_k, t_{k+1}], \quad 1 \leq i < j \leq N_\ell. \tag{5.3.16}$$

Proof. The 1-norm span of the k-th Bézier segment B_k^i of the i-th agent is defined as

$$\operatorname{span}_1(B_k^i) = \max_{0\le l\le n} \left\| c_{k,l}^i - \operatorname{cent}(\mathcal{CH}_k^i) \right\|_1, \tag{5.3.17}$$

where

$$\operatorname{cent}(\mathcal{CH}_k^i) = \frac{1}{n+1}\sum_{l=0}^{n} c_{k,l}^i \tag{5.3.18}$$

is the centroid of the control points associated with the convex hull $\mathcal{CH}_k^i$.

Under constraints (5.3.3), Proposition 3.4.1 establishes the following Euclidean tube bound for all $l \in \{1,\dots,n\}$:

$$\min\left\{ \left\| c_{k,l}^i - c_{k,0}^i \right\|, \left\| c_{k,l}^i - c_{k,n}^i \right\| \right\} \le \epsilon_k^i. \tag{5.3.19}$$

Using $\|x\|_1 \le \sqrt{d}\|x\|$ for every $x \in \mathbb{R}^d$, we obtain

$$\min\left\{ \left\| c_{k,l}^i - c_{k,0}^i \right\|_1, \left\| c_{k,l}^i - c_{k,n}^i \right\|_1 \right\} \le \sqrt{d}\,\epsilon_k^i. \tag{5.3.20}$$

Define

$$m_{k,i}^* = \frac{c_{k,0}^i + c_{k,n}^i}{2} \tag{5.3.21}$$

as the midpoint of the segment connecting the endpoints. Without loss of generality, suppose that the first term realizes the minimum in (5.3.19). Then,

$$\begin{aligned} \left\| c_{k,l}^i - m_{k,i}^* \right\|_1 &\le \left\| c_{k,l}^i - c_{k,0}^i \right\|_1 + \left\| c_{k,0}^i - m_{k,i}^* \right\|_1 \\ &\le \sqrt{d}\,\epsilon_k^i + \frac{1}{2}\left\| c_{k,0}^i - c_{k,n}^i \right\|_1. \end{aligned} \tag{5.3.22}$$

The same bound holds when the second term realizes the minimum in (5.3.19). Therefore,

$$\begin{aligned} \left\| \operatorname{cent}(\mathcal{CH}_k^i) - m_{k,i}^* \right\|_1 &= \left\| \frac{1}{n+1}\sum_{l=0}^{n} c_{k,l}^i - m_{k,i}^* \right\|_1 \\ &\le \frac{1}{n+1}\sum_{l=0}^{n} \left\| c_{k,l}^i - m_{k,i}^* \right\|_1 \\ &\le \sqrt{d}\,\epsilon_k^i + \frac{1}{2}\left\| c_{k,0}^i - c_{k,n}^i \right\|_1. \end{aligned} \tag{5.3.23}$$

Consequently, for every control point $c_{k,l}^i$,

$$\begin{aligned}\left\|c_{k,l}^i - \mathrm{cent}(\mathcal{CH}_k^i)\right\|_1 &\le \left\|c_{k,l}^i - m_{k,i}^*\right\|_1 + \left\|\mathrm{cent}(\mathcal{CH}_k^i) - m_{k,i}^*\right\|_1 \\ &\le 2\sqrt{d}\,\epsilon_k^i + \left\|c_{k,0}^i - c_{k,n}^i\right\|_1. \end{aligned} \tag{5.3.24}$$

It follows that

$$\mathrm{span}_1(B_k^i) \le 2\sqrt{d}\,\epsilon_k^i + \left\|c_{k,0}^i - c_{k,n}^i\right\|_1. \tag{5.3.25}$$

Similarly,

$$\mathrm{span}_1(B_k^j) \le 2\sqrt{d}\,\epsilon_k^j + \left\|c_{k,0}^j - c_{k,n}^j\right\|_1. \tag{5.3.26}$$

By the convex-hull property of Bézier curves, $B_k^i(t) \in \mathcal{CH}_k^i$ for all $t \in [t_k, t_{k+1}]$. Hence,

$$\left\|B_k^i(t) - \mathrm{cent}(\mathcal{CH}_k^i)\right\|_1 \le \mathrm{span}_1(B_k^i), \qquad \forall t \in [t_k, t_{k+1}], \tag{5.3.27}$$

and similarly,

$$\left\|B_k^j(t) - \mathrm{cent}(\mathcal{CH}_k^j)\right\|_1 \le \mathrm{span}_1(B_k^j), \qquad \forall t \in [t_k, t_{k+1}]. \tag{5.3.28}$$

Now consider any two Bézier curves B_k^i and B_k^j over $t \in [t_k, t_{k+1}]$. The mixed-integer constraints (5.3.12)–(5.3.13) ensure that there exists an active sign pattern $s \in \{-1, 1\}^d$ satisfying $s^\top p_k^{ij} \ge D_k^{ij}$. Since $s^\top p_k^{ij} \le \|p_k^{ij}\|_1$, the centroid-separation condition (5.3.10) follows. Using (5.3.25) and (5.3.26), we obtain

$$\begin{aligned}\left\|\mathrm{cent}(\mathcal{CH}_k^i) - \mathrm{cent}(\mathcal{CH}_k^j)\right\|_1 &\ge 2\sqrt{d}\,\epsilon_k^i + \left\|c_{k,0}^i - c_{k,n}^i\right\|_1 \\ &\quad + 2\sqrt{d}\,\epsilon_k^j + \left\|c_{k,0}^j - c_{k,n}^j\right\|_1 + \epsilon_{\mathrm{inter},k}\sqrt{d} \\ &\ge \mathrm{span}_1(B_k^i) + \mathrm{span}_1(B_k^j) + \epsilon_{\mathrm{inter},k}\sqrt{d}. \end{aligned} \tag{5.3.29}$$

Therefore, for every $t \in [t_k, t_{k+1}]$,

$$\begin{aligned}\left\|B_k^i(t) - B_k^j(t)\right\|_1 &\ge \left\|\mathrm{cent}(\mathcal{CH}_k^i) - \mathrm{cent}(\mathcal{CH}_k^j)\right\|_1 \\ &\quad - \left\|B_k^i(t) - \mathrm{cent}(\mathcal{CH}_k^i)\right\|_1 \\ &\quad - \left\|B_k^j(t) - \mathrm{cent}(\mathcal{CH}_k^j)\right\|_1 \\ &\ge \epsilon_{\mathrm{inter},k}\sqrt{d}. \end{aligned} \tag{5.3.30}$$

Using $\|x\|_1 \le \sqrt{d}\|x\|$ for every $x \in \mathbb{R}^d$, it follows that $\left\|B_k^i(t) - B_k^j(t)\right\| \ge \epsilon_{\mathrm{inter},k}$. This completes the proof.

□

Proposition 5.3.3 (STL Satisfaction for Multiple Quadrotors)**.** *Given an STL specification φ, consider N_ℓ quadrotors indexed by $\ell \in \{1, \dots, N_\ell\}$. Suppose that the multi-quadrotor STL-constrained trajectory generation problem defined in Problem 5.2.1 is feasible and returns the Bézier reference trajectories $y_d^\ell : \mathbb{T} \to \mathbb{R}^d$, $\ell \in \{1, \dots, N_\ell\}$. Suppose further that condition (5.3.10) holds for all agent pairs on each segment. Then, the tracking trajectories $\{y_{\text{tr}}^\ell\}_{\ell=1}^{N_\ell}$ satisfy the STL specification φ. Moreover, the distance between any two tracking trajectories satisfies*

$$\|y_{\text{tr}}^i(t) - y_{\text{tr}}^j(t)\| \geq \gamma_c, \qquad \forall t \in \mathbb{T},\ 1 \leq i < j \leq N_\ell. \tag{5.3.31}$$

Proof. The proof follows from the STL satisfaction result in Chapter 4 and the inter-agent separation guarantee established in Lemma 5.3.2. By Lemma 5.3.1, the returned Bézier reference trajectories $\{y_d^\ell\}_{\ell=1}^{N_\ell}$ jointly satisfy the STL specification φ. From Chapter 4, the tracking controller guarantees $\|y_{\text{tr}}^\ell(t) - y_d^\ell(t)\| \leq \Gamma(t)$, for all $t \in \mathbb{T}$, $\ell \in \{1, \dots, N_\ell\}$. Since the STL robustness margin satisfies $\Gamma(t) \leq \rho^{\ell,\varphi}(t), \forall t \in \mathbb{T}$, the tracking error remains within the robustness margin of the reference trajectory. Therefore, by the tracking-robust STL satisfaction result in Chapter 4, the tracking trajectories $\{y_{\text{tr}}^\ell\}_{\ell=1}^{N_\ell}$ satisfy the STL specification φ.

By Lemma 5.3.2, for any $1 \leq i < j \leq N_\ell$ and any $t \in [t_k, t_{k+1}]$, $\|y_d^i(t) - y_d^j(t)\| \geq \epsilon_{\text{inter},k} = 2\Gamma(t_k) + \gamma_c$. Using the triangle inequality and the tracking-error bound, we obtain

$$\begin{aligned} \|y_{\text{tr}}^i(t) - y_{\text{tr}}^j(t)\| &\geq \|y_d^i(t) - y_d^j(t)\| - \|y_{\text{tr}}^i(t) - y_d^i(t)\| - \|y_{\text{tr}}^j(t) - y_d^j(t)\| \\ &\geq 2\Gamma(t_k) + \gamma_c - 2\Gamma(t). \end{aligned} \tag{5.3.32}$$

Since Γ is nonincreasing, $\Gamma(t) \leq \Gamma(t_k)$ for all $t \in [t_k, t_{k+1}]$. It follows that $\|y_{\text{tr}}^i(t) - y_{\text{tr}}^j(t)\| \geq \gamma_c$. Thus, any two tracking trajectories remain separated by at least γ_c, and the proof is complete. □

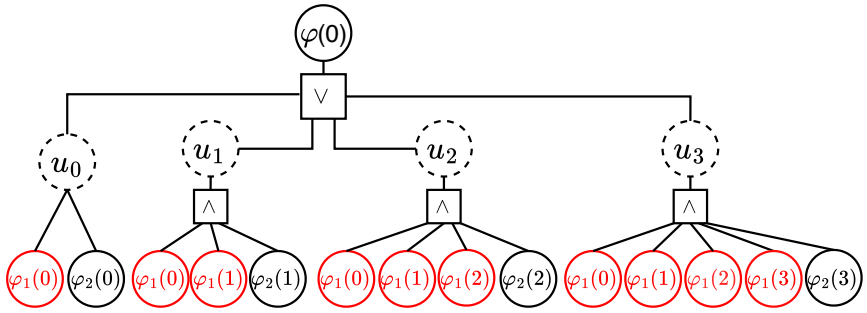
Remark 5.3.4. *The multi-agent safety encoding is conservative because it separates the full convex hulls of the Bézier curves. The ℓ_1-norm is used because its superlevel constraint admits an exact finite sign-disjunction representation. However, the set defined by a lower bound on the ℓ_1-norm is nonconvex. Therefore, the exact encoding requires binary variables that select one of the 2^d sign patterns, together with linear or convex constraints within the selected branch. The resulting collision-avoidance formulation is compatible with MICP solvers, but it does not consist solely of continuous constraints.*

Complexity Analysis

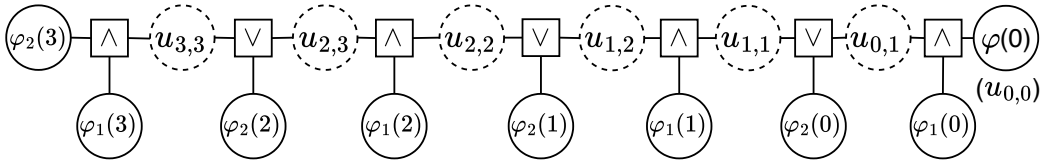
Let $|\varphi|$ denote the number of disjunction occurrences in the STL formula φ, and let N_{win} denote the maximum number of sampling indices involved in the encoding of

any bounded temporal operator. For continuous-time STL specifications over piecewise Bézier trajectories with N segments, the STL encoding introduces binary variables for the disjunctive operators. As shown in Figure 5.1, the backward-recursion encoding reuses subformula encodings, thereby avoiding the repeated encoding. Consequently, the number of binary variables introduced by the STL encoding scales as $\mathcal{O}\left(N|\varphi|N_{\text{win}}\right)$, whereas the direct semantic expansion scales as $\mathcal{O}\left(N|\varphi|N_{\text{win}}^2\right)$. Furthermore, for other nested formulas, the backward recursion reuses each subformula encoding, making the costs of the temporal windows additive rather than multiplicative.

The exact mixed-integer sign-disjunction encoding of the pairwise collision-avoidance constraint introduces 2^d binary variables for each agent pair and each Bézier segment. Hence, the collision-avoidance encoding introduces $2^d N\binom{N_\ell}{2}$ additional binary variables and the same order of indicator constraints. The total number of binary variables therefore scales as $\mathcal{O}\left(N|\varphi|N_{\text{win}} + 2^d N N_\ell^2\right)$. For the three-dimensional quadrotor position space, $d = 3$, so eight sign-selection binary variables are required for each agent pair and each segment. The continuous auxiliary constraints used to represent the endpoint ℓ_1-norms scale as $\mathcal{O}\left(dN_\ell^2 N\right)$.



(a) Semantic expansion encoding. $\varphi(0) = \bigvee_{j=0}^{3} u_j$ where $u_j = \varphi_2(j) \wedge (\bigwedge_{i=0}^{j}) \varphi_1(i),\ j = \{0, 1, 2, 3\}$.



(b) Backward-recursion encoding. With the terminal term $u_{3,3} := \varphi_1(3) \wedge \varphi_2(3)$, the auxiliary binary variables are defined recursively, for $i = 2, 1, 0$, as $u_{i,i+1} := \varphi_2(i) \vee u_{i+1,i+1}$ and $u_{i,i} := \varphi_1(i) \wedge u_{i,i+1}$. Then $\varphi(0) = u_{0,0} = \varphi_1(0) \wedge u_{0,1}$.

Figure 5.1: Two MICP encodings of $\varphi = \left(\varphi_1 \mathcal{U}_{[0,3]} \varphi_2\right)$ at time $t = 0$. Dashed circles denote auxiliary binary variables. The bounded-until window contains $N\text{win} = 4$ sampled time instants with sampling interval $\Delta t = 1$. (a) The direct semantic expansion requires $\mathcal{O}\left(N_{\text{win}}^2\right)$ logical-variable occurrences and constraints because the prefix conditions $\varphi_1(\cdot)$ (red circles) are repeated for each possible witness time. (b) The backward-recursion encoding introduces auxiliary binary variables for intermediate truth values and requires only $\mathcal{O}(N_{\text{win}})$ auxiliary variables and logical constraints.

5.4 Simulations

This section evaluates the proposed framework in terms of runtime performance, safety, and stability. The simulations are implemented in Python and Problem 5.2.1 is solved as an MICP using Gurobi [52]. All simulations are conducted on an 8-core Apple M1 machine with 16 GB of RAM.

The following subsections present four scenarios to validate the proposed method. Since multi-agent safety is already encoded in z_{inter}, it is omitted from the subsequent

specifications. For readability, the statement that robot r avoids a region $\mathbf{O}$ is denoted by $r \notin \mathbf{O}$, which is logically equivalent to the negation $\neg(r \in \mathbf{O})$.

5.4.1 Runtime Analysis

To evaluate how runtime scales with the number of agents, we consider the simple scenario shown in Figure 5.2. Because this scenario is simple and can accommodate different numbers of quadrotors, it better isolates the effect of N_ℓ on computation time. The specification is

$$\varphi = \bigwedge_{\ell=1}^{N_\ell} \left(\square_{[0,T]}(r_\ell \notin \mathbf{Y}) \wedge \Diamond_{[0,T]}(r_\ell \in \mathbf{B}) \right), \tag{5.4.1}$$

which requires each robot r_ℓ to avoid the yellow obstacle $\mathbf{Y}$ and reach the blue region $\mathbf{B}$. Table 5.1 shows that runtime increases with the number of agents N_ℓ, consistent with the $\mathcal{O}(N_\ell^2 N)$ scaling of pairwise collision-avoidance constraints discussed in Section 5.3. The runtime is compared against the MICP baseline proposed in [155], which uses direct semantic expansions and Bézier-curve trajectory parameterization. To enable a fair comparison in the multi-agent setting, the same collision-avoidance encoding is also incorporated into the baseline. As the number of robots N_ℓ increases, the proposed method shows lower runtime than the baseline. Although the direct and backward-recursive encodings of the Always and Eventually operators both scale linearly with N_{win}, the backward recursion decomposes each temporal operator into local Boolean relations involving only consecutive indices. This produces a sparser and more structured mixed-integer formulation, which can be handled more efficiently by the solver.

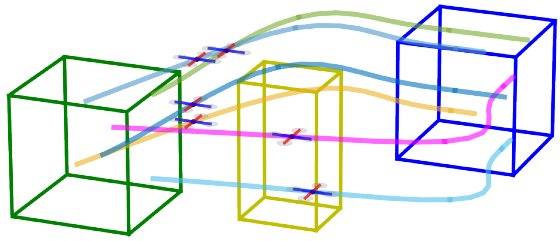


Figure 5.2: Case 0 environment with six quadrotors ($N_\ell = 6$).

Table 5.1: Comparison of runtime (s) for different numbers of robots.

#Robots	Proposed	MICP [155]
2	**0.16**	0.91
3	**1.17**	5.56
4	**8.79**	27.20
5	**17.89**	73.80
6	**54.42**	171.98

5.4.2 Safety Analysis

Case 1: Narrow Corridors. In Case 1, the quadrotors are required to fly through a gradually narrowing passage while avoiding the five yellow obstacle regions $\{\mathbf{Y}_i\}_{i=1}^{5}$, and eventually reach a compact blue region $\mathbf{B}$, as shown in Figure 5.3(a)–(c). The navigation constraints become progressively tighter. Given $T = 17$ s, the specification is

$$\varphi = \bigwedge_{\ell=1}^{2} \Big(\big(\wedge_{i=1}^{5} \Box_{[0,T]}(r_\ell \notin \mathbf{Y}_i)\big) \wedge (\Diamond_{[0,T]} r_\ell \in \mathbf{B}) \Big). \tag{5.4.2}$$

Case 2: Reach and Avoid. Case 2 requires each quadrotor r_ℓ to stay away from the yellow obstacles $\mathbf{Y}$ and remain in the green region $\mathbf{G}$ for at least 2 seconds. Finally, all quadrotors need to reach the diagonally opposite blue region, as depicted in Figure 5.3(d)–(f), resulting in an interlaced routing challenge. Given $T = 20$ s, the specification is

$$\varphi = \bigwedge_{\ell=1}^{4} \Big(\big(\Box_{[0,T]}(r_\ell \notin \mathbf{Y})\big) \wedge \big(\Diamond_{[0,T-2]} \Box_{[0,2]} r_\ell \in \mathbf{G}\big) \wedge \big(\Diamond_{[0,T]} r_\ell \in \mathbf{B}_{\ell'}\big) \Big), \tag{5.4.3}$$

where the pairing $(\ell, \ell') \in \{(1,3),(2,4),(3,1),(4,2)\}$ encodes this diagonal goal assignment.

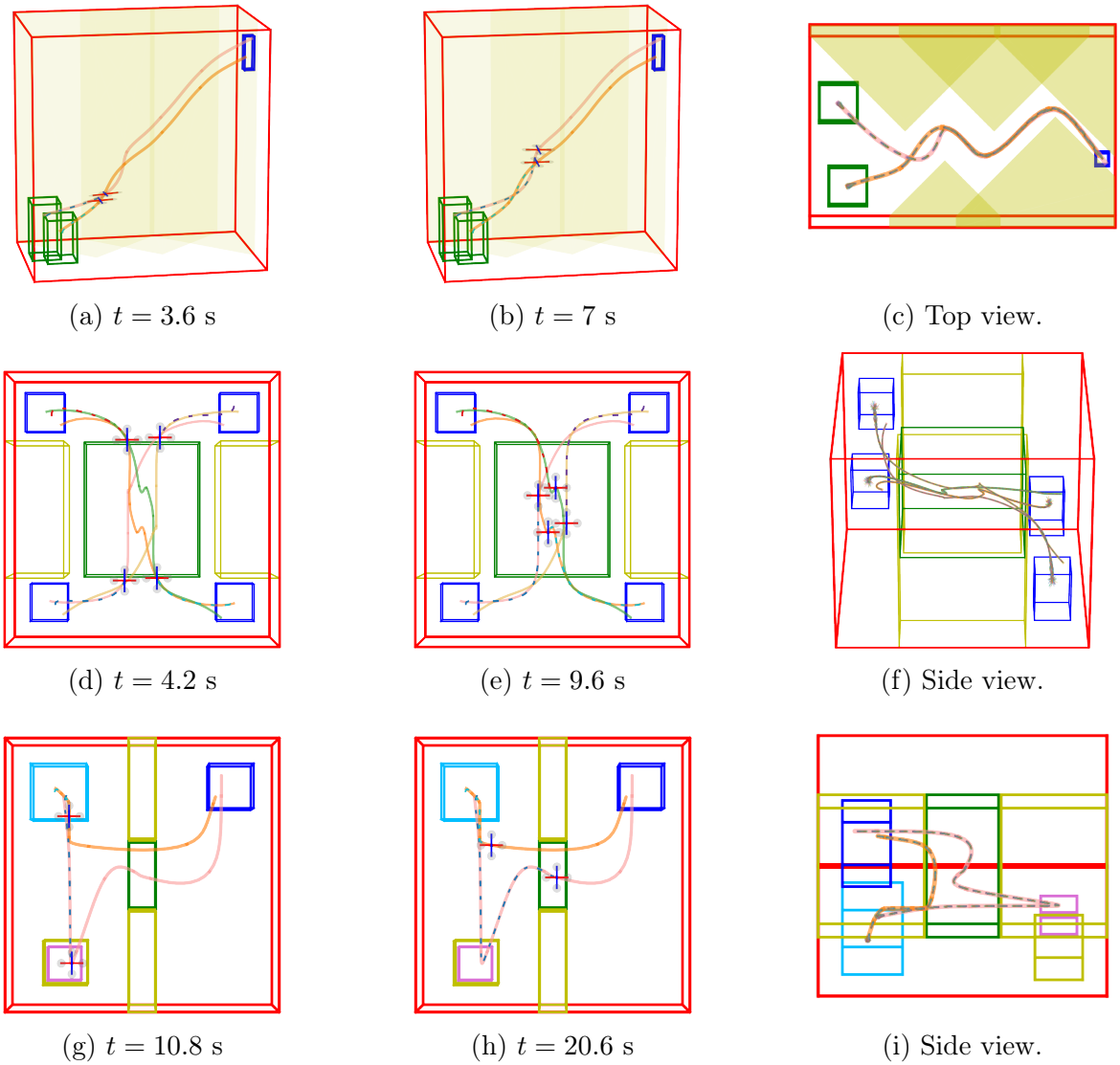
Case 3: Key–Door Puzzle. In Case 3, the green gate $\mathbf{G}$ is the only passage to the blue target region $\mathbf{B}$. Robot r_1 must first retrieve the key from the purple region $\mathbf{K}$, then unlock the gate and begin passing through $\mathbf{G}$, while robot r_2 must remain outside $\mathbf{G}$. After the gate is unlocked, both robots must pass through $\mathbf{G}$ to reach $\mathbf{B}$ while avoiding

the yellow obstacle region $\mathbf{Y}$ at all times, as shown in Figure 5.3(g)–(i). With $T = 30$ s, the specification is expressed as

$$\begin{aligned}\varphi_{\text{door}} = &((r_1 \notin \mathbf{G}) \wedge (r_2 \notin \mathbf{G}))\, \mathcal{U}_{[0,T]} \Big((r_1 \in \mathbf{K}) \\ &\wedge \left[(r_2 \notin \mathbf{G}) \mathcal{U}_{[0,T]} \left((r_1 \in \mathbf{G}) \wedge (r_2 \notin \mathbf{G}) \right) \right] \Big) \\ &\wedge \bigwedge\nolimits_{\ell=1}^{2} \Big(\left(\Diamond_{[0,T]} r_\ell \in \mathbf{B} \right) \wedge \left(\Box_{[0,T]} r_\ell \notin \mathbf{Y} \right) \Big).\end{aligned} \tag{5.4.4}$$

This specification is complex due to the nested structure of the two *Until* operators.

In all cases mentioned above, the Bézier trajectories are parameterized by eight control points ($n = 7$), and the tracking integration time step is 0.2 s. For each case, the planned trajectories y_d^ℓ and the corresponding tracking trajectories y_{tr}^ℓ are generated for all N_ℓ agents. For each approach, 100 tracking trials per agent are simulated under identical initial error conditions. The simulation results are analyzed in terms of stability, safety, and robustness.

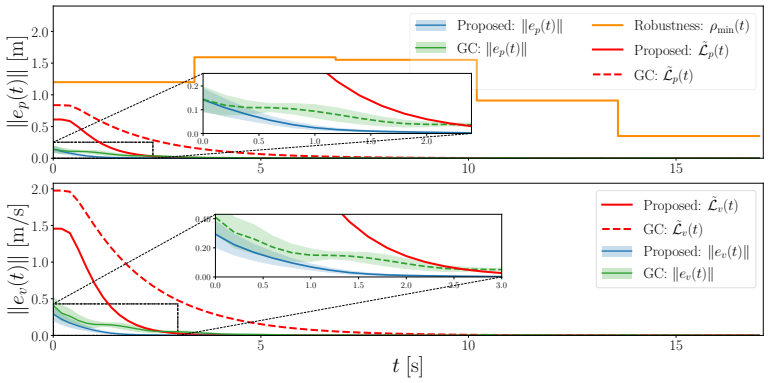


(a) $t = 3.6$ s (b) $t = 7$ s (c) Top view.

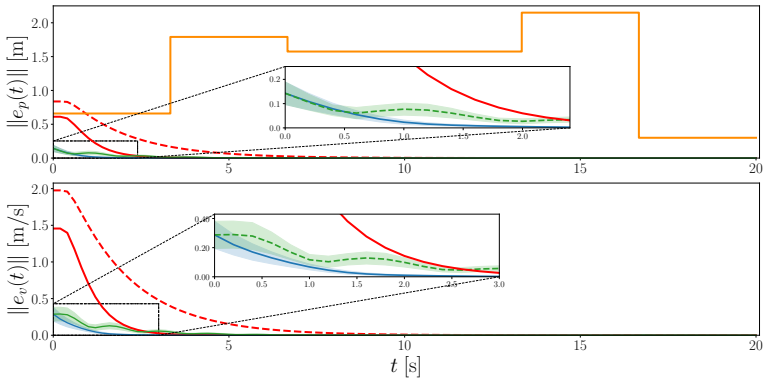
(d) $t = 4.2$ s (e) $t = 9.6$ s (f) Side view.

(g) $t = 10.8$ s (h) $t = 20.6$ s (i) Side view.

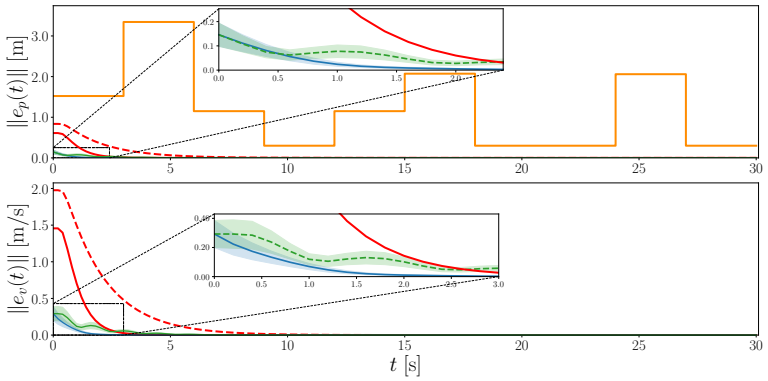
Figure 5.3: Benchmarks and trajectory snapshots for Cases 1–3, shown from top to bottom. For each case, the first two columns present two representative time snapshots, where only the trajectory with the largest $\|e_p\|$ is plotted for clarity. The third column shows an additional view together with the tracking trajectories. In all environment plots, solid lines denote planned trajectories and dashed lines denote tracking trajectories.



(a) Case 1: Narrow Corridors.



(b) Case 2: Reach and Avoid.



(c) Case 3: Key–Door Puzzle.

Figure 5.4: Comparison of the position and velocity tracking errors and their corresponding bounds for Cases 1–3, comparing the proposed method with conventional geometric control (GC) [127]. To avoid repetition and improve clarity, the legend in panel (a) also applies to panels (b) and (c).

Safety is assessed by examining both the planning and tracking trajectories. A safety threshold $\gamma_c = 0.2$ m is enforced on the minimum inter-trajectory distance. As reported in Table 5.2, the minimum inter-trajectory distances are evaluated over the discrete time $\mathcal{T} = \{t_k = kdt \mid 0 \leq kdt \leq T\}$. The minimum distance between planning trajectories is defined as

$$\text{dist}_{\text{plan}} = \min_{1 \leq i < j \leq N_\ell} \min_{t_k \in \mathcal{T}} \left\| y_d^i(t_k) - y_d^j(t_k) \right\|. \tag{5.4.5}$$

The tracking distance $\text{dist}_{\text{track}}$ is the overall minimum inter-trajectory distance over all time steps in $\mathcal{T}$, across all agents and all tracking trials (100 per agent). Table 5.2 shows that, in all cases, the minimum inter-trajectory distances exceed the safety threshold γ_c, thereby validating the collision-avoidance behavior of both the planned and tracked trajectories. Moreover, the worst-case robustness across all robots is $\rho^{\varphi}_{\min} = 0.3 > 0$, confirming that all Bézier trajectories satisfy the STL specifications. Meanwhile, Figure 5.4 shows that $\rho^{\varphi}_{\min}(t) \geq \tilde{L}_p(t) = \Gamma(t)$ for all $t \in [0, T]$, implying that the tracked trajectories also satisfy the STL specifications according to (5.2.3).

Table 5.2: Safety summary across cases, including the number of robots (N_ℓ), number of segments (#S), runtime, worst-case robustness margin $\rho^{\varphi}_{\min}$, and the minimum inter-trajectory distances of the planned and tracking trajectories.

Case	N_ℓ	#S	Runtime (s)	$\rho^{\varphi}_{\min}$	$\text{dist}_{\text{plan}}$	$\text{dist}_{\text{track}}$
Narrow Corridors	2	5	0.07	0.3	0.615	0.616
Reach-Avoid	4	6	11.92	0.3	0.462	0.461
Key-Door	2	10	1.48	0.3	0.681	0.676

5.4.3 Stability Analysis

Table 5.3 illustrates the comparison of the stability performance of the proposed method and that of [127]. The position error convergence time $t_{c,p}$ is defined as the earliest time such that $\|e_p(t)\| \leq 10^{-2}$ m for all $t \geq t_{c,p}$. Similarly, the velocity error convergence time $t_{c,v}$ is defined as the earliest time such that $\|e_v(t)\| \leq 10^{-2}$ m/s for all $t \geq t_{c,v}$. The post-convergence errors $\bar{e}_p$ and $\bar{e}_v$ are defined as the means of all finite samples of $\|e_p(t)\|$ for $t \geq t_{c,p}$ and $\|e_v(t)\|$ for $t \geq t_{c,v}$, respectively, pooled over all agents and trajectories in each case.

As shown in Figure 5.4, the proposed controller yields lower time-varying mean errors $\|e_p(t)\|$ and $\|e_v(t)\|$, together with narrower shaded bands, indicating smaller dispersion across trials over time. Because the post-convergence behavior is not directly evident in Figure 5.4, the convergence times and post-convergence errors in Table 5.3 show that the proposed method achieves faster convergence, tighter steady-state tracking, and reduced residual oscillations relative to the scalar-gain baseline [127].

Table 5.3: Tracking performance comparison between the proposed method and [127], reported as mean ± standard deviation over pooled trajectories from all agents in each case. Here, $t_{c,p}$ and $t_{c,v}$ denote the convergence times for position and velocity errors, respectively.

Case	Convergence time $t_{c,p}$ (s)		$\bar{e}_p \pm \sigma_p$ ($\times 10^{-3}$ m)		Convergence time $t_{c,v}$ (s)		$\bar{e}_v \pm \sigma_v$ ($\times 10^{-3}$ m/s)	
	Proposed	GC [127]	Proposed	GC [127]	Proposed	GC [127]	Proposed	GC [127]
Narrow Corridors	$\mathbf{1.57 \pm 0.24}$	4.41 ± 0.36	$\mathbf{0.60 \pm 0.07}$	1.73 ± 0.10	$\mathbf{2.10 \pm 0.24}$	5.38 ± 0.32	$\mathbf{0.76 \pm 0.14}$	2.29 ± 0.01
Reach-Avoid	$\mathbf{1.57 \pm 0.23}$	4.33 ± 0.39	$\mathbf{0.38 \pm 0.04}$	1.02 ± 0.08	$\mathbf{2.10 \pm 0.23}$	5.89 ± 0.49	$\mathbf{0.40 \pm 0.10}$	1.2 ± 0.21
Key-Door	$\mathbf{1.59 \pm 0.25}$	4.34 ± 0.42	$\mathbf{0.22 \pm 0.02}$	0.63 ± 0.05	$\mathbf{2.12 \pm 0.24}$	5.89 ± 0.51	$\mathbf{0.24 \pm 0.04}$	0.69 ± 0.13

5.5 Summary

This chapter presents a continuous-time control framework for multi-quadrotor systems under STL specifications by integrating geometric tracking control with MICP-based Bézier trajectory generation. The proposed approach enables safe and dynamically feasible motion planning with formal tracking guarantees. Compared with conventional geometric control with real gains, the proposed matrix-gain geometric controller yields tighter tracking bounds and improved tracking performance. In addition, the multi-agent extension of the continuous-time STL Bézier planning framework improves computational efficiency while ensuring specification satisfaction. While this chapter focuses on bounded-time STL specifications, long-horizon robotic objectives, such as repeated visitation, sequencing, and persistent safety, motivate the use of LTL specifications and the abstraction-free controller synthesis framework developed in the next chapter.

Chapter 6

Control Synthesis for Continuous-Time Systems under Linear Temporal Logic

6.1 Introduction

In this chapter, we focus on LTL, a rigorous formal language widely used in model checking [23] and increasingly adopted in robotics for specifying temporally structured mission requirements, such as sequencing and long-horizon task execution. One of the central challenges in realizing such specifications in robotic systems is to design controllers for long-horizon tasks while accounting for system dynamics and safety constraints [66].

A common approach is to synthesize controllers for the resulting subtasks using CBF-QP methods [113]. However, such methods still have technical restrictions. Notably, CBF-QP-based methods typically require online optimization to compute control inputs, and conflicts among barrier-function constraints may lead to infeasible quadratic programs (QPs). To address this issue, methods for the composition of barrier functions and a prioritization-based control method were proposed to guarantee the feasibility of the controller [132]. Nevertheless, these approaches still typically rely on online optimization to compute the control input. Alternatively, hybrid barrier certificates establish satisfaction of syntactically co-safe LTL specifications through a hybrid-system eventuality argument [14]. This approach avoids the standard CBF-QP formulation, but it does not explicitly incorporate bounded-control constraints. Motivated by recent progress on the compatibility of CBFs and control Lyapunov functions (CLFs), we instead exploit CLBFs as certificates for safe stabilization [24, 93, 92]. Since the level sets of a CLBF characterize regions where

both safety and convergence can be guaranteed, they provide a natural way to represent feasible transitions between subtasks.

Building on the formally certified CLBFs proposed in [92], this chapter develops an abstraction-free framework for controller synthesis under decomposed LTL specifications. By approximating and patching the winning sets of the decomposed subtasks, the proposed framework constructs feasible trajectories that satisfy the specification while potentially avoiding repeated online QP computations. The main contributions are threefold, as follows.

- To the best of the author's knowledge, this is the first work connecting the approximation of the winning set of LTL specifications and the level sets of CLBFs under bounded control.
- By formulating the local specification tasks as a sequence of safe stabilization problems, the proposed method yields formally verified switching feedback controllers under bounded control constraints. With the offline-computed CLBFs, this formulation enables efficient online planning under bounded control constraints and allows for dynamic, on-the-fly re-planning in the presence of state perturbations while tracking reference trajectories in noisy environments or with perturbed dynamics.
- A set of case studies, including numerical simulations and a real-world experiment in which a Crazyflie quadrotor tracks a trajectory generated by the proposed algorithm, is provided to demonstrate the effectiveness of the proposed method.

6.2 Preliminaries and Problem Statement

Throughout this chapter, we consider a control-affine system of the form:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \qquad t \in \mathbb{R}_+, \tag{6.2.1}$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$, and the functions $f : \mathcal{X} \to \mathbb{R}^n$ and $g : \mathcal{X} \to \mathbb{R}^{n\times p}$ are locally Lipschitz. The control input $u : [0, \infty) \to \mathbb{R}^p$ satisfies

$$u(t) \in \mathcal{U}_{\text{box}} := \left\{ v \in \mathbb{R}^p \mid \underline{u}_j \leq v_j \leq \overline{u}_j,\ j = 1, \ldots, p \right\}, \tag{6.2.2}$$

where $\underline{u}_j, \overline{u}_j \in \mathbb{R}$, with $\underline{u}_j \leq \overline{u}_j$, denote the lower and upper bounds on each control component, respectively. Such hyperbox constraints model physical input limitations commonly

encountered in practice, such as motor torque limits. We denote by $\phi(t, x_0, u)$ the solution of system (6.2.1) with initial condition $x(0) = x_0$ under an input signal $u(\cdot)$ satisfying (6.2.2). The solution is defined on its maximal interval of existence $I(x_0, u) \subseteq [0, \infty)$. We assume that $I(x_0, u) = [0, \infty)$ for all $x_0 \in \mathcal{X}$ and all input signals $u(\cdot)$ considered in this chapter.

6.2.1 State Constraints and Smooth Approximation

Consider N state constraints defined by a finite set $\{h_i\}_{i=1}^N$ of continuously differentiable functions $h_i : \mathcal{X} \to \mathbb{R}$. For each $i = 1, \ldots, N$, the safe set $\mathcal{C}_i$ associated with h_i is defined as $\mathcal{C}_i := \{x \in \mathcal{X} \mid h_i(x) \leq 1\}$. Accordingly, the overall safe region $\mathcal{C}_{\text{safe}}$ is the intersection of all $\mathcal{C}_i$, i.e., $\mathcal{C}_{\text{safe}} = \bigcap_{i=1}^N \mathcal{C}_i$, which can also be written as the pointwise maximum of all h_i as follows:

$$\mathcal{C}_{\text{safe}} = \{x \in \mathcal{X} \mid h_{\max}(x) \leq 1\}, \tag{6.2.3}$$

where $h_{\max}(x) := \max_{i=1,\ldots,N} h_i(x)$ is not a continuously differentiable function in general. Thus, we approximate $h_{\max}$ by the smooth log-sum-exp function [93] as follows:

$$h_{\text{soft}}(x; \tau) := \frac{1}{\tau} \log \left(\sum_{i=1}^{N} e^{\tau h_i(x)} \right), \tag{6.2.4}$$

where $\tau > 0$ is a tunable parameter. For all $x \in \mathcal{X}$ and $\tau > 0$,

$$h_{\max}(x) \leq h_{\text{soft}}(x; \tau) \leq h_{\max}(x) + \frac{\log N}{\tau}, \tag{6.2.5}$$

and hence $h_{\text{soft}}(\cdot; \tau)$ converges pointwise to $h_{\max}$ as $\tau \to \infty$. To simplify the subsequent notation, for a fixed $\tau > 0$, let $h(x) := h_{\text{soft}}(x; \tau)$ and define the approximate safe set as

$$\mathcal{C} := \{x \in \mathcal{X} \mid h(x) \leq 1\} \subseteq \mathcal{C}_{\text{safe}}. \tag{6.2.6}$$

Denote the boundary of $\mathcal{C}$ by $\partial\mathcal{C} := \{x \in \mathcal{X} \mid h(x) = 1\}$.

The system is said to be safe with respect to the constraint set $\{h_i\}_{i=1}^N$ if for any initial state $x_0 \in \mathcal{C}$, there exists a control input $u \in \mathcal{U}_{\text{box}}$ such that the solution $\phi(t, x_0, u)$ remains in $\mathcal{C}$ for all $t \geq 0$. Let the interior of $\mathcal{C}$ be denoted by $\text{Int}(\mathcal{C})$, which is assumed to be nonempty. Let $x_m^\star \in \text{Int}(\mathcal{C})$ be an equilibrium point of system (6.2.1) under zero input, i.e., $f(x_m^\star) = 0$. We then define the safe region of attraction (ROA) under a given state-feedback control policy $u = \kappa(x) \in \mathcal{U}_{\text{box}}$ as

$$\begin{aligned} \mathcal{R}_{\text{safe}} := \Big\{ x_0 \in \mathcal{C} \mid \ & \phi(t, x_0, \kappa) \in \mathcal{C}, \ \forall t \geq 0, \\ & \text{and} \quad \lim_{t \to \infty} \phi(t, x_0, \kappa) = x_m^\star \Big\}. \end{aligned} \tag{6.2.7}$$

6.2.2 LTL Satisfaction of Continuous-Time Trajectories

As described in Section 2.6, an LTL formula over the finite set of atomic propositions AP is interpreted over an infinite word in $(2^{AP})^\omega$. Throughout this chapter, $\mathrm{LTL}_{\setminus\bigcirc}$ denotes the next-free fragment of LTL, obtained by excluding the next operator $\bigcirc$ [120]. Formulas in $\mathrm{LTL}_{\setminus\bigcirc}$ are stutter invariant under finite repetitions of consecutive identical letters. For example, for distinct atomic propositions $\pi_1, \pi_2, \pi_3 \in AP$, the words $\sigma = \{\pi_1\}\{\pi_2\}(\{\pi_3\})^\omega$ and $\sigma' = \{\pi_1\}\{\pi_1\}\{\pi_2\}(\{\pi_3\})^\omega$ satisfy the same formulas in $\mathrm{LTL}_{\setminus\bigcirc}$. This property allows satisfaction to be independent of the number of observations recorded within an interval where the label remains unchanged.

Recall the labeling function $L : \mathcal{X} \to 2^{AP}$ introduced in Section 2.6, where $L(x) \subseteq AP$ denotes the set of atomic propositions satisfied at state $x \in \mathcal{X}$. Following [145], for each atomic proposition $\pi \in AP$, define $[\![\pi]\!] := \{x \in \mathcal{X} \mid \pi \in L(x)\}$. For each subset of atomic propositions $P \in 2^{AP}$, define $[\![P]\!] := \{x \in \mathcal{X} \mid L(x) = P\}$. Hence, $[\![P]\!]$ is the set of states that satisfy all and only the propositions in P. Note that for any two distinct $P, Q \in 2^{AP}$, $[\![P]\!] \cap [\![Q]\!] = \emptyset$.

Definition 6.2.1 (Finite variability [97]). *Let $\xi : \mathbb{R}_+ \to \mathcal{X}$ be a continuous-time signal. The signal ξ is said to be of finite variability under L if there exists a sequence $\mathcal{I} = \{I_m\}_{m=1}^{\infty}$ of non-overlapping intervals covering $\mathbb{R}_+$ such that*

$$L(\xi(t)) = L(\xi(t')), \quad \forall\, t, t' \in I_m, \quad m \geq 1, \tag{6.2.8}$$

and their left endpoints $\inf(I_m)$ satisfy $\inf(I_m) \to \infty$ as $m \to \infty$.

The finite-variability condition excludes infinitely many label changes over a bounded time interval [97, 145].

Definition 6.2.2 (Trace of a continuous-time trajectory, adapted from [97]). *Let $\xi : \mathbb{R}_+ \to \mathcal{X}$ be a continuous-time trajectory of finite variability under L. Suppose that there exists $\varepsilon_0 > 0$ such that $L(\xi(t)) = L(\xi(0))$ for all $t \in [0, \varepsilon_0)$. For every $t \geq 0$, let $L_\xi^+(t) \in 2^{AP}$ denote the unique right-hand label for which there exists $\varepsilon_t > 0$ such that $L(\xi(s)) = L_\xi^+(t)$ for all $s \in (t, t + \varepsilon_t)$. The trace of ξ is the infinite word $\sigma = \sigma_0\sigma_1 \cdots \in (2^{AP})^\omega$, defined recursively together with a sequence of observation times $\{\tau_m\}_{m \in \mathbb{N}_0} \subseteq \mathbb{R}_+ \cup \{\infty\}$ as follows:*

i) $\tau_0 := 0$ and $\sigma_0 := L(\xi(0))$;

ii) for $m \geq 0$, let $\tau_{m+1} := \inf\{t > \tau_m \mid L_\xi^+(t) \neq \sigma_m\}$, where $\inf \varnothing := \infty$. If $\tau_{m+1} < \infty$, then $\sigma_{m+1} := L_\xi^+(\tau_{m+1})$;

iii) if $\tau_{m+1} = \infty$, define $\tau_k := \infty$ and $\sigma_k := \sigma_m$ for all $k \geq m+1$.

Remark 6.2.3 (Well-definedness of the continuous-time trace)**.** *The initial-label condition in Definition 6.2.2 ensures that $L_\xi^+(0) = \sigma_0$ and rules out an instantaneous label change at $t = 0$. Finite variability guarantees that the right-hand label $L_\xi^+(t)$ exists uniquely at every $t \geq 0$ and that the finite label-change times have no finite accumulation point. Consequently, if infinitely many label changes occur, then $\tau_m \to \infty$ as $m \to \infty$. If only finitely many label changes occur, the final symbol is repeated indefinitely according to Definition 6.2.2. Hence, the recursive construction in Definition 6.2.2 produces a well-defined infinite word $\sigma \in (2^{AP})^\omega$.*

Definition 6.2.4 (Held trace of a continuous-time trajectory)**.** *Let $\sigma = \sigma_0\sigma_1 \cdots \in (2^{AP})^\omega$ be the trace of ξ given by Definition 6.2.2, and suppose that $\sigma_0 \neq \emptyset$. The held trace of ξ is the infinite word $\sigma_{\mathrm{h}} = \sigma_{\mathrm{h},0}\sigma_{\mathrm{h},1} \cdots \in \left(2^{AP} \setminus \{\emptyset\}\right)^\omega$, defined recursively by $\sigma_{\mathrm{h},0} := \sigma_0$ and, for every $n \in \mathbb{N}_0$,*

$$\sigma_{\mathrm{h},n+1} := \begin{cases} \sigma_{n+1}, & \sigma_{n+1} \neq \emptyset, \\ \sigma_{\mathrm{h},n}, & \sigma_{n+1} = \emptyset. \end{cases} \tag{6.2.9}$$

Since σ_{h} is defined recursively from the well-defined infinite word σ, and since $\sigma_0 \neq \emptyset$, the held trace is also a well-defined infinite word in $\left(2^{AP} \setminus \{\emptyset\}\right)^\omega$.

Definition 6.2.5 (LTL satisfaction of a continuous-time trajectory under held-trace semantics)**.** *Let $\xi : \mathbb{R}_+ \to \mathcal{X}$ be a continuous-time trajectory with a well-defined held trace $\sigma_{\mathrm{h}} = \sigma_{\mathrm{h},0}\sigma_{\mathrm{h},1} \cdots$ as given in Definition 6.2.4. The trajectory ξ satisfies an $\mathrm{LTL}_{\setminus\bigcirc}$ formula φ under the held-trace semantics, denoted by $\xi \models \varphi$, if $\sigma_{\mathrm{h}} \models \varphi$.*

For an initial state $x_0 \in \mathcal{X}$, a control input $u : \mathbb{R}_+ \to \mathcal{U}_{\mathrm{box}}$ is called admissible from x_0 if system (6.2.1) admits a unique trajectory $\phi(\cdot, x_0, u)$ that is defined for all $t \geq 0$ and remains in $\mathcal{X}$.

Definition 6.2.6 (Winning set for an LTL specification)**.** *Given the control-affine system (6.2.1) and an $\mathrm{LTL}_{\setminus\bigcirc}$ formula φ, the winning set with respect to φ is defined as*

$$\mathrm{Win}(\varphi) := \{x_0 \in \mathcal{X} \mid \exists\, u \text{ admissible from } x_0 \text{ such that } \phi(\cdot, x_0, u) \models \varphi\}. \tag{6.2.10}$$

where trajectory satisfaction is defined in Definition 6.2.5.

Thus, $\mathrm{Win}(\varphi)$ is the set of all initial states from which there exists an admissible control input whose corresponding trajectory admits a well-defined held trace satisfying φ.

6.2.3 Problem Formulation

Consider the continuous-time system (6.2.1) and an $\mathrm{LTL}_{\setminus\bigcirc}$ specification φ defined over a finite set of atomic propositions AP. Since computing the exact winning set $\mathrm{Win}(\varphi)$ is generally intractable, the objective of this chapter is to compute a certified inner approximation

$$\widetilde{\mathrm{Win}}(\varphi) \subseteq \mathrm{Win}(\varphi), \tag{6.2.11}$$

and to synthesize a switching state-feedback control policy such that, for every initial condition $x_0 \in \widetilde{\mathrm{Win}}(\varphi)$, the policy generates an admissible control input u whose corresponding trajectory satisfies $\phi(\cdot, x_0, u) \models \varphi$.

As defined in Definition 2.6.5, a lasso-form word consists of a finite prefix followed by a finite suffix that repeats indefinitely. In this chapter, given a lasso-form word σ such that $\sigma \models \varphi$ and the labeling function L associated with AP, the task of generating a closed-loop trajectory whose held trace is stutter-equivalent to σ is decomposed into the finite sequence of transitions associated with the prefix and one cycle of the suffix. By stutter invariance, such a trajectory satisfies φ. Each transition is formulated as a safe-stabilization subtask, and the subtasks associated with the suffix are repeated indefinitely. The overall synthesis problem is then reduced to computing CLBF-based inner approximations of the controller-specific local winning sets and patching the corresponding local feedback controllers.

Together, the certified set $\widetilde{\mathrm{Win}}(\varphi)$ and the switching policy obtained by patching the local feedback controllers provide a correct-by-construction solution to the $\mathrm{LTL}_{\setminus\bigcirc}$ control synthesis problem: for every initial condition $x_0 \in \widetilde{\mathrm{Win}}(\varphi)$, the resulting closed-loop trajectory satisfies the prescribed $\mathrm{LTL}_{\setminus\bigcirc}$ specification.

6.3 LTL Specification Decomposition

Consider an $\mathrm{LTL}_{\setminus\bigcirc}$ formula φ over AP. Let $\sigma^{\mathrm{ref}} \in \left(2^{AP} \setminus \{\emptyset\}\right)^{\omega}$ be a reference word satisfying $\sigma^{\mathrm{ref}} \models \varphi$. Since $\mathrm{LTL}_{\setminus\bigcirc}$ is next-free, it suffices to synthesize a feedback control policy whose closed-loop trajectory induces a trace σ that is stutter-equivalent to σ^{ref}. Following the sequential structure of σ^{ref}, we decompose the synthesis problem into safe-stabilization subtasks, each associated with a transition between two consecutive letters of σ^{ref}.

Specifically, for each consecutive pair $(\sigma^{\mathrm{ref}}_m, \sigma^{\mathrm{ref}}_{m+1})$ in the word σ^{ref}, with $m \in \mathbb{N}_0$, let $\mathcal{X}_m \subseteq \mathcal{X}$ denote the local state-space region associated with the transition from σ^{ref}_m to

$\sigma^{\mathrm{ref}}_{m+1}$. We formulate such a transition as a local reach-avoid control problem and define its transition tuple as

$$\mathcal{T}_m := (\mathcal{S}_m, \mathcal{G}_m, \mathcal{A}_m), \tag{6.3.1}$$

where $\mathcal{S}_m, \mathcal{G}_m, \mathcal{A}_m \subseteq \mathcal{X}_m$ denote the source set, the goal set, and the forbidden set, respectively. Note that the forbidden set $\mathcal{A}_m$ includes the proposition regions that must not be visited while realizing the transition.

Since the reference word σ^{ref} can be written in lasso form as $\sigma^{\mathrm{ref}} = \sigma^{\mathrm{ref}}_{\mathrm{pre}} \left(\sigma^{\mathrm{ref}}_{\mathrm{suf}}\right)^{\omega}$, the induced infinite transition-task sequence $\{\mathcal{T}_m\}_{m\in\mathbb{N}_0}$ admits a finite prefix–suffix representation. Accordingly, let $M_{\mathrm{pre}} = \mathrm{len}(\sigma^{\mathrm{ref}}_{\mathrm{pre}})$ denote the number of prefix transitions, including the transition from the last prefix symbol to the first suffix symbol when the prefix is nonempty, and let $M_{\mathrm{suf}} = \mathrm{len}(\sigma^{\mathrm{ref}}_{\mathrm{suf}}) \geq 1$ denote the number of transitions in one suffix cycle, including the wrap-around transition from the last suffix symbol to the first one. Define the finite transition-index set $\mathcal{M} := \{0, \ldots, M-1\}$ where $M = M_{\mathrm{pre}} + M_{\mathrm{suf}}$. We write the corresponding finite pattern as $\{\mathcal{T}_m\}_{m\in\mathcal{M}}$ and define a lasso index map $\zeta : \mathbb{N}_0 \to \mathcal{M}$ by

$$\zeta(n) = \begin{cases} n, & 0 \leq n < M_{\mathrm{pre}}, \\ M_{\mathrm{pre}} + \big((n - M_{\mathrm{pre}}) \bmod M_{\mathrm{suf}}\big), & n \geq M_{\mathrm{pre}}. \end{cases} \tag{6.3.2}$$

Thus, the transition active at stage $n \in \mathbb{N}_0$ is $\mathcal{T}_{\zeta(n)}$.

Therefore, realizing the reference word σ^{ref} is reduced to a sequence of transitions $\{\mathcal{T}_m\}_{m\in\mathcal{M}}$, each requiring the system to move from $\mathcal{S}_m$ to $\mathcal{G}_m$ while avoiding $\mathcal{A}_m$. The following assumptions are imposed.

Assumption 6.3.1. *For each $m \in \mathcal{M}$, consider the finite transition $\mathcal{T}_m = (\mathcal{S}_m, \mathcal{G}_m, \mathcal{A}_m)$ associated with a local state space $\mathcal{X}_m \subseteq \mathcal{X}$, where $\mathcal{S}_m, \mathcal{G}_m, \mathcal{A}_m \subseteq \mathcal{X}_m$. We assume that*

- *$\mathcal{S}_m \subseteq [\![\sigma^{\mathrm{ref}}_m]\!]$ is nonempty and satisfies $\mathcal{S}_m \cap \mathcal{A}_m = \emptyset$;*
- *$\mathcal{G}_m$ is nonempty, connected, and closed, and satisfies $\mathcal{G}_m \subseteq \mathrm{Int}\big([\![\sigma^{\mathrm{ref}}_{m+1}]\!]\big)$ and $\mathcal{G}_m \cap \mathcal{A}_m = \emptyset$. Here, $\mathrm{Int}\big([\![\sigma^{\mathrm{ref}}_{m+1}]\!]\big)$ denotes the interior of $[\![\sigma^{\mathrm{ref}}_{m+1}]\!]$.*
- *$\mathcal{A}_m$ is closed.*
- *$\mathcal{X}_m \setminus (\mathcal{A}_m \cup \mathcal{G}_m) \subseteq [\![\sigma^{\mathrm{ref}}_m]\!] \cup [\![\emptyset]\!]$.*

Definition 6.3.2 (Finite-memory control strategy)**.** *Consider the finite transition $\{\mathcal{T}_m\}_{m\in\mathcal{M}}$ induced by the reference word σ^{ref} and the lasso index map $\zeta : \mathbb{N}_0 \to \mathcal{M}$. A finite-memory control strategy for system (6.2.1) is a tuple*

$$\boldsymbol{\kappa} = (\mathcal{M}, m_0, \mu, \kappa), \tag{6.3.3}$$

where

i) The set of finite-memory states is given by the finite transition-index set $\mathcal{M}$, where each $m \in \mathcal{M}$ indicates the currently active local transition task $\mathcal{T}_m$;

ii) $m_0 \in \mathcal{M}$ is the initial memory state, and in this work we set $m_0 = \zeta(0)$;

iii) $\mu : \mathcal{X} \times \mathcal{M} \to \mathcal{M}$ is the memory-update function defined by

$$\mu(x, \zeta(n)) = \begin{cases} \zeta(n+1), & x \in \mathcal{G}_{\zeta(n)}, \\ \zeta(n), & \text{otherwise}, \end{cases} \qquad n \in \mathbb{N}_0. \tag{6.3.4}$$

iv) $\kappa : \mathcal{X} \times \mathcal{M} \to \mathcal{U}_{box}$ is the control law. For each $m \in \mathcal{M}$, the local control law associated with memory state m is denoted by $\kappa_m : \mathcal{X}_m \to \mathcal{U}_{box}$ and defined as $\kappa_m(x) := \kappa(x, m)$, for $x \in \mathcal{X}_m$.

According to Definition 6.3.2, at stage $n \in \mathbb{N}_0$, the active memory state is $\zeta(n) \in \mathcal{M}$. The control input $u(t) = \kappa(x(t), \zeta(n))$ is applied until the trajectory first enters the goal set $\mathcal{G}_{\zeta(n)}$ during this stage, at a time denoted by τ_{n+1}. The memory state is then updated according to $\mu\big(x(\tau_{n+1}), \zeta(n)\big) = \zeta(n+1)$. The resulting closed-loop trajectory of system (6.2.1) from x_0 under the finite-memory strategy $\boldsymbol{\kappa}$ is denoted by $\phi_{\boldsymbol{\kappa}}(\cdot; x_0)$.

Definition 6.3.3 (Winning set for a word-induced transition)**.** *Consider a transition $\mathcal{T}_m = (\mathcal{S}_m, \mathcal{G}_m, \mathcal{A}_m)$, as defined in (6.3.1), induced by the pair of consecutive symbols $(\sigma^{\mathrm{ref}}_m, \sigma^{\mathrm{ref}}_{m+1})$ in the reference word σ^{ref}. Let $\kappa_m : \mathcal{X}_m \to \mathcal{U}_{box}$ be the local control law associated with memory state m. The winning set of $\mathcal{T}_m$ under κ_m is defined as*

$$\mathrm{Win}_\sigma(\mathcal{T}_m) := \left\{ x_0 \in \mathcal{S}_m \;\middle|\; \begin{array}{l} \exists\, \tau_m \geq 0 \text{ such that} \\ \phi(t, x_0, \kappa_m) \in \mathcal{X}_m \setminus \mathcal{A}_m, \quad \forall t \in [0, \tau_m], \\ \phi(\tau_m, x_0, \kappa_m) \in \mathcal{G}_m \end{array} \right\}. \tag{6.3.5}$$

Here, τ_m denotes the duration of the transition from σ^{ref}_m to $\sigma^{\mathrm{ref}}_{m+1}$, $\kappa_m : \mathcal{X}_m \to \mathcal{U}_{box}$ is a local feedback control strategy, and $\phi(\cdot, x_0, \kappa_m)$ is the corresponding trajectory of system (6.2.1) under κ_m.

Assumption 6.3.4 (Local finite variability)**.** *For each local transition $\mathcal{T}_m$ and each closed-loop trajectory generated by κ_m during a finite transition interval, the labeling signal $t \mapsto L(x(t))$ changes value only finitely many times on that interval.*

Assumption 6.3.4 imposes finite variability on each finite transition interval. Together with the absence of finite accumulation of task-switching times, it ensures finite variability of the complete closed-loop trajectory under L.

Proposition 6.3.5. *Consider an* $\mathrm{LTL}_{\setminus\bigcirc}$ *formula* φ *over* AP *and a reference word* $\sigma^{\mathrm{ref}} = \sigma^{\mathrm{ref}}_{\mathrm{pre}}\left(\sigma^{\mathrm{ref}}_{\mathrm{suf}}\right)^{\omega} \in \left(2^{AP}\setminus\{\emptyset\}\right)^{\omega}$ *satisfying* $\sigma^{\mathrm{ref}} \models \varphi$. *Let* $\{\mathcal{T}_m\}_{m=0}^{M-1}$ *be the corresponding finite collection of word-induced transitions, with fixed local control laws* $\{\kappa_m\}_{m=0}^{M-1}$ *and associated winning sets* $\{\mathrm{Win}_\sigma(\mathcal{T}_m)\}_{m=0}^{M-1}$. *Let* $\zeta : \mathbb{N}_0 \to \mathcal{M}$ *be the lasso index map. Under Assumption 6.3.4, suppose that*

$$x_0 \in \mathrm{Win}_\sigma\big(\mathcal{T}_{\zeta(0)}\big), \qquad \mathcal{G}_{\zeta(n)} \subseteq \mathrm{Win}_\sigma\big(\mathcal{T}_{\zeta(n+1)}\big), \quad n \in \mathbb{N}_0, \tag{6.3.6}$$

hold and that sequential first-entry switching generates switching times $0 = \tau_0 < \tau_1 < \tau_2 < \cdots$ *satisfying* $\tau_n \to \infty$. *Then there exists a finite-memory control strategy* $\boldsymbol{\kappa} = (\mathcal{M}, \zeta(0), \mu, \kappa)$ *for system (6.2.1) such that the corresponding closed-loop trajectory is of finite variability under* L *and satisfies* $\phi_{\boldsymbol{\kappa}}(\cdot; x_0) \models \varphi$.

Proof. Define the finite-memory control law by $\kappa(x, m) := \kappa_m(x), \qquad x \in \mathcal{X}_m, \quad m \in \mathcal{M}$, and assign arbitrary admissible values in $\mathcal{U}_{\mathrm{box}}$ outside the corresponding local regions. Let $x(t) := \phi_{\boldsymbol{\kappa}}(t; x_0)$. On each interval $[\tau_n, \tau_{n+1})$, the active memory state is $\zeta(n)$ and is updated to $\zeta(n+1)$ when the trajectory first enters $\mathcal{G}_{\zeta(n)}$.

We first show inductively that $x(\tau_n) \in \mathrm{Win}_\sigma\big(\mathcal{T}_{\zeta(n)}\big)$, for $n \in \mathbb{N}_0$. The claim holds for $n = 0$ by (6.3.6). Suppose that it holds for some n. By the definition of $\mathrm{Win}_\sigma(\mathcal{T}_{\zeta(n)})$ under the fixed local control law $\kappa_{\zeta(n)}$, the trajectory remains in $\mathcal{X}_{\zeta(n)} \setminus \mathcal{A}_{\zeta(n)}$ and reaches $\mathcal{G}_{\zeta(n)}$ at the finite first-entry time τ_{n+1}. Hence, $x(\tau_{n+1}) \in \mathcal{G}_{\zeta(n)} \subseteq \mathrm{Win}_\sigma\big(\mathcal{T}_{\zeta(n+1)}\big)$, which completes the induction.

By Assumption 6.3.4, the trajectory undergoes only finitely many label changes on each transition interval $[\tau_n, \tau_{n+1}]$. Moreover, since $\tau_n \to \infty$, every bounded time interval intersects only finitely many transition intervals. It follows that the complete closed-loop trajectory is of finite variability under L.

By the label-consistency conditions of the word-induced transitions, $L(x(t)) \in \left\{\sigma^{\mathrm{ref}}_{\zeta(n)}, \emptyset\right\}$, for $t \in [\tau_n, \tau_{n+1})$, and $L(x(\tau_{n+1})) = \sigma^{\mathrm{ref}}_{\zeta(n+1)}$. Thus, the held observation remains $\sigma^{\mathrm{ref}}_{\zeta(n)}$ during the nth transition and is updated to $\sigma^{\mathrm{ref}}_{\zeta(n+1)}$ upon reaching $\mathcal{G}_{\zeta(n)}$. Applying this argument for every $n \in \mathbb{N}_0$ shows that the held trace σ_{h} of the closed-loop trajectory is stutter-equivalent to σ^{ref}.

Since the $\mathrm{LTL}_{\setminus\bigcirc}$ formula φ is stutter-invariant, $\sigma_{\mathrm{h}} \models \varphi$. By Definition 6.2.5, the closed-loop trajectory satisfies $\phi_{\boldsymbol{\kappa}}(\cdot; x_0) \models \varphi$. □

Proposition 6.3.5 motivates formulating each word-induced transition $\mathcal{T}_m$ as a bounded-input safe-stabilization subproblem whose solution provides the corresponding local control law κ_m.

6.4 LTL Control Synthesis with Patching CLBFs

This section constructs patching CLBFs for the transitions $\{\mathcal{T}_m\}_{m=0}^{M-1}$ to obtain bounded local controllers and winning sets, which are used to synthesize a finite-memory control strategy for the LTL specification φ.

6.4.1 Patching CLBFs Construction

For each transition $\mathcal{T}_m$, let $\{h_i^m\}_{i=1}^{N_m}$ denote its N_m state-constraint functions and let $\mathcal{C}^m \subseteq \mathcal{X}_m$ denote the corresponding approximate safe set constructed as in (6.2.6). We assume that $\mathcal{C}^m \cap \mathcal{A}_m = \emptyset$ and $\mathcal{G}_m \subseteq \mathcal{C}^m$. Let $x_m^\star \in \mathrm{Int}(\mathcal{G}_m)$ be an equilibrium of system (6.2.1) under zero input, i.e., $f(x_m^\star) = 0$. To construct the patching CLBFs, we first present the definitions of CLFs and CBFs for system (6.2.1) under bounded control inputs.

Definition 6.4.1. *A continuously differentiable function $V : \mathcal{X}_m \to \mathbb{R}$ is a CLF for system (6.2.1) on $\mathcal{C}^m$ with respect to $x_m^\star$ if $V(x_m^\star) = 0$ and for every $x \in \mathcal{C}^m \setminus \{x_m^\star\}$, there exists $u \in \mathcal{U}_{box}$ such that*

$$\begin{aligned} V(x) > 0, \quad &\text{and} \\ L_f V(x) + L_g V(x) u < 0, \quad & \end{aligned} \tag{6.4.1}$$

where $L_f V(x) := \nabla V(x)^\top f(x)$ and $L_g V(x) := \nabla V(x)^\top g(x)$ are the Lie derivatives of V along f and g, respectively.

Definition 6.4.2. *A continuously differentiable function $h : \mathcal{X}_m \to \mathbb{R}$ is a CBF for system (6.2.1) with respect to $\mathcal{C}^m$ if, for every $x \in \mathcal{C}^m$, there exists $u \in \mathcal{U}_{box}$ such that*

$$\begin{aligned} h(x) \leq 1, \quad &\text{and} \\ L_f h(x) + L_g h(x) u < 0. \quad & \end{aligned} \tag{6.4.2}$$

Following [93, 92], we have the following proposition about the strict compatibility condition between the CLF $\mathcal{V}$ and the CBF h.

Proposition 6.4.3 ([92]). *For system (6.2.1), let $V : \mathcal{X}_m \to \mathbb{R}$ be a candidate CLF on $\mathcal{C}^m$ with respect to $x_m^\star$, and let $h : \mathcal{X}_m \to \mathbb{R}$ be a candidate CBF with respect to $\mathcal{C}^m$. If the following conditions hold:*

i) For every $x \in \mathcal{C}^m \setminus \{x_m^\star\}$, there exists $u \in \mathcal{U}_{box}$ such that

$$L_f\mathcal{V}(x) + L_g\mathcal{V}(x)u < 0. \tag{6.4.3}$$

Note that this condition is equivalent to

$$\begin{aligned} &(h(x) \leq 1 \wedge x \neq x_m^\star) \\ \implies &L_f\mathcal{V}(x) + \sum_{j=1}^{p} \big(m_j \left(L_g\mathcal{V}(x)\right)_j - r_j \left| \left(L_g\mathcal{V}(x)\right)_j \right| \big) < 0, \end{aligned} \tag{6.4.4}$$

where $m_j = \frac{\underline{u}_j + \overline{u}_j}{2}$ and $r_j = \frac{\overline{u}_j - \underline{u}_j}{2}$.

ii) For every $x \in \partial\mathcal{C}^m$, there exists $u \in \mathcal{U}_{box}$ such that

$$L_f\mathcal{V}(x) + L_g\mathcal{V}(x)u < 0, \tag{6.4.5}$$

and

$$L_fh(x) + L_gh(x)u < 0. \tag{6.4.6}$$

Given the hyperbox constraint on the control input, the above two conditions (6.4.5) and (6.4.6) are equivalent to

$$\begin{aligned} \Big(&h(x) = 1 \\ &\wedge \begin{cases} \lambda_1, \lambda_2, \ \lambda_j^{\pm} \geq 0 \quad (j = 1, \ldots, p), \\ \lambda_1 + \lambda_2 + \sum_{j=1}^{p} (\lambda_j^+ + \lambda_j^-) = 1 \end{cases} \\ &\wedge \ \lambda_1 L_g\mathcal{V}(x) + \lambda_2 L_gh(x) + \sum_{j=1}^{p} (\lambda_j^+ - \lambda_j^-)\, e_j = 0 \Big) \\ \implies &\lambda_1 L_f\mathcal{V}(x) + \lambda_2 L_fh(x) < \sum_{j=1}^{p} (\lambda_j^+ \overline{u}_j - \lambda_j^- \underline{u}_j). \end{aligned} \tag{6.4.7}$$

Here $e_j \in \mathbb{R}^{1\times p}$ denotes the j-th standard basis row vector.

Then, $\mathcal{V}$ and h are said to be strictly compatible.

We refer the readers to [92] for the detailed proof of Proposition 6.4.3. It is worth noting that conditions (6.4.4) for the CLF and (6.4.7) for compatibility are first-order formulas, which can be verified by off-the-shelf SMT solvers, such as dReal [48], providing the formal guarantees. Suppose that $\mathcal{C}^m$ is compact. We then have the following theorem.

Theorem 6.4.4 ([93, 92])**.** *Let $\mathcal{V}$ be a CLF and h be a CBF for system (6.2.1) with respect to $\mathcal{C}^m$. If $\mathcal{V}$ and h are strictly compatible as defined in Proposition 6.4.3, then there exists a continuously differentiable CLBF $W : \mathcal{X}_m \to \mathbb{R}$ for system (6.2.1), satisfying:*

i) W is positive definite, and $W(x_m^\star) = 0$.

ii) $\mathcal{C}^m = \{x \in \mathcal{X}_m \mid W(x) \leq 1\}$.

iii) For every $x \in \mathcal{C}^m \setminus \{x_m^\star\}$, there exists $u \in \mathcal{U}_{box}$ such that

$$L_f W(x) + L_g W(x) u < 0. \tag{6.4.8}$$

We refer the reader to [92] for the detailed proof of this theorem. To construct such a CLBF W, for some $\varepsilon \in (0, 1)$, we define the bump function

$$b(x) = \begin{cases} 0, & h(x) \leq 1 - \varepsilon, \\ 3\left(\dfrac{h(x) - (1-\varepsilon)}{\varepsilon}\right)^2 - 2\left(\dfrac{h(x) - (1-\varepsilon)}{\varepsilon}\right)^3, & 1 - \varepsilon < h(x) < 1, \\ 1, & h(x) \geq 1. \end{cases} \tag{6.4.9}$$

Compared with the exponential bump function used in [92], the cubic bump preserves C^1 continuity and improves numerical stability, since both b and its derivative with respect to h are bounded polynomials for $1 - \varepsilon < h(x) < 1$. This polynomial form also avoids the near-singular division encountered in the exponential design near $h(x) = 1 - \varepsilon$.

As discussed in [92], the compatibility also holds on the inner boundary of $\mathcal{C}^m$, i.e., $\partial\mathcal{C}_\varepsilon^- := \{x \in \mathcal{X}_m \mid 1 - \varepsilon \leq h(x) \leq 1\}$. By rescaling $\mathcal{V}$ if necessary, we can assume that $\mathcal{V}(x) \leq 1$, $\forall x \in \mathcal{C}^m$, and $\mathcal{V}(x) \leq h(x)$ $\forall x \in \partial\mathcal{C}_\varepsilon^-$. Then, W can be constructed as follows [93]:

$$W(x) := (1 - b(x))\mathcal{V}(x) + b(x)h(x), \tag{6.4.10}$$

Using W, we next synthesize a bounded local controller and obtain an inner approximation of the corresponding winning set, which is used to certify the set inclusion in (6.3.6).

6.4.2 Finite-Memory Control Synthesis

Considering a transition $\mathcal{T}_m$ and its patching CLBF W constructed in (6.4.10), we synthesize the bounded controller proposed in [80] for hyperrectangle input constraints. For clarity, we suppress the transition index m throughout the following local controller construction.

For system (6.2.1) with $u \in \mathcal{U}_{\text{box}} \subset \mathbb{R}^p$, define

$$a_W(x) := L_f W(x) = \nabla W(x) \cdot f(x), \tag{6.4.11}$$

$$b_{W,j}(x) := -L_{g_j} W(x) = -\nabla W(x) \cdot g_j(x), \quad j \in \{1, \dots, p\}, \tag{6.4.12}$$

where $\nabla W(x) = b\nabla h + (1-b)\nabla V + (h-V)\nabla b$ and $g_j(x)$ denotes the j-th column of $g(x)$. The bounded controller $u : \mathcal{X}_m \to \mathcal{U}_{\text{box}}$ is then defined componentwise by

$$u_j(x) := \rho_j(x)\omega_j(x), \quad j = 1, \dots, p. \tag{6.4.13}$$

Here, the vertex-control component ω_j and the continuous rescaling factor ρ_j are given by

$$\omega_j(x) := \begin{cases} \overline{u}_j, & b_{W,j}(x) > 0, \\ 0, & b_{W,j}(x) = 0, \\ \underline{u}_j, & b_{W,j}(x) < 0, \end{cases} \tag{6.4.14}$$

$$\rho_j(x) := \begin{cases} 1 - \big(1 - \Gamma(x)\upsilon_j(x)\big)e^{\tau_j(x)\upsilon_j(x)}, & \Pi_j(x) > 0, \\ 0, & \Pi_j(x) = 0, \end{cases} \tag{6.4.15}$$

where $\Pi_j(x) := b_{W,j}(x)\omega_j(x) \geq 0$, and, for $\sum_{k=1}^p \Pi_k(x) > 0$,

$$\Gamma(x) := \frac{|a_W(x)| + a_W(x)}{2\sum_{k=1}^p \Pi_k(x)}, \tag{6.4.16}$$

$$\upsilon_j(x) := \frac{\Pi_j(x)}{\sum_{k=1}^p \Pi_k(x)}, \tag{6.4.17}$$

$$\tau_j(x) := p\frac{\ln(1-\Gamma(x))}{1-\Gamma(x)} - \eta\Pi_j(x), \tag{6.4.18}$$

Here, $\eta > 0$ is a prescribed constant. Note that the strict CLBF decrease condition in (6.4.8) ensures that $0 \leq \Gamma(x) < 1$.

Restoring the transition index m, the preceding construction yields a patching CLBF W_m in (6.4.10) and a bounded local controller u^m in (6.4.13) for each transition $\mathcal{T}_m$, $m \in \mathcal{M}$. The corresponding 1-sublevel set is

$$\mathcal{W}_m^1 := \{x \in \mathcal{X}_m \mid W_m(x) \leq 1\}. \tag{6.4.19}$$

By Theorem 6.4.4, the 1-sublevel set satisfies $\mathcal{W}_m^1 = \mathcal{C}^m$. Under u^m, the set $\mathcal{W}_m^1$ is forward invariant and strictly safe, namely, $\mathcal{W}_m^1 \subseteq \mathcal{X}_m \setminus \mathcal{A}_m$. Furthermore, because the closed-loop trajectory asymptotically converges to $x_m^\star \in \mathrm{Int}(\mathcal{G}_m)$, every trajectory initialized in $(\mathcal{S}_m \cap \mathcal{W}_m^1) \setminus \mathcal{G}_m$ reaches $\mathcal{G}_m$ in finite time. Identifying the local control law κ_m in Definition 6.3.3 with u^m, we therefore define

$$\widetilde{\mathrm{Win}}_\sigma(\mathcal{T}_m) := (\mathcal{S}_m \cap \mathcal{W}_m^1) \setminus \mathcal{G}_m \subseteq \mathrm{Win}_\sigma(\mathcal{T}_m), \qquad m \in \mathcal{M}. \tag{6.4.20}$$

Based on Proposition 6.3.5, we establish the following proposition for the local controllers synthesized from the patching CLBFs.

Proposition 6.4.5. *Consider system (6.2.1), an $\mathrm{LTL}_{\setminus\bigcirc}$ formula φ over AP, and a reference word $\sigma^{\mathrm{ref}} = \sigma^{\mathrm{ref}}_{\mathrm{pre}} \left(\sigma^{\mathrm{ref}}_{\mathrm{suf}}\right)^\omega \in \left(2^{AP} \setminus \{\emptyset\}\right)^\omega$ satisfying $\sigma^{\mathrm{ref}} \models \varphi$, with lasso index map $\zeta : \mathbb{N}_0 \to \mathcal{M}$. For each $m \in \mathcal{M}$, let u^m be the patching CLBF controller associated with $\mathcal{T}_m$. Under Assumptions 6.3.1 and 6.3.4, with $\kappa_m = u^m$, suppose that*

$$\mathcal{G}_{\zeta(n)} \subseteq \widetilde{\mathrm{Win}}_\sigma\big(\mathcal{T}_{\zeta(n+1)}\big), \qquad n \in \mathbb{N}_0. \tag{6.4.21}$$

Consider the finite-memory control strategy $\boldsymbol{\kappa} = (\mathcal{M}, \zeta(0), \mu, \kappa)$ in Definition 6.3.2, with $\kappa_m(x) = u^m(x)$ for all $x \in \mathcal{X}_m$ and $m \in \mathcal{M}$. Then, for every $x_0 \in \widetilde{\mathrm{Win}}_\sigma(\mathcal{T}_{\zeta(0)})$, the corresponding closed-loop trajectory is of finite variability under L and satisfies $\phi_{\boldsymbol{\kappa}}(\cdot; x_0) \models \varphi$. Consequently,

$$\widetilde{\mathrm{Win}}_\sigma(\varphi) := \widetilde{\mathrm{Win}}_\sigma\big(\mathcal{T}_{\zeta(0)}\big) \tag{6.4.22}$$

is a certified inner approximation of the global winning set, i.e., $\widetilde{\mathrm{Win}}_\sigma(\varphi) \subseteq \mathrm{Win}(\varphi)$.

Proof. Let $x_0 \in \widetilde{\mathrm{Win}}_\sigma(\varphi)$. By (6.4.22), (6.4.20), and (6.4.21),

$$\begin{aligned} x_0 &\in \widetilde{\mathrm{Win}}_\sigma\big(\mathcal{T}_{\zeta(0)}\big) \subseteq \mathrm{Win}_\sigma\big(\mathcal{T}_{\zeta(0)}\big), \\ \mathcal{G}_{\zeta(n)} &\subseteq \widetilde{\mathrm{Win}}_\sigma\big(\mathcal{T}_{\zeta(n+1)}\big) \subseteq \mathrm{Win}_\sigma\big(\mathcal{T}_{\zeta(n+1)}\big), \qquad n \in \mathbb{N}_0. \end{aligned} \tag{6.4.23}$$

Thus, by induction, sequential first-entry switching generates an infinite sequence of switching times such that, for every $n \in \mathbb{N}$, $0 = \tau_0 < \tau_1 < \cdots < \tau_n < \infty$.

We next show that these switching times have no finite accumulation point, namely, that $\tau_n \to \infty$. Since $\widetilde{\mathrm{Win}}_\sigma(\mathcal{T}_m) = (\mathcal{S}_m \cap \mathcal{W}_m^1) \setminus \mathcal{G}_m$, condition (6.4.21) implies $\mathcal{G}_{\zeta(n)} \cap \mathcal{G}_{\zeta(n+1)} = \emptyset$, for $n \in \mathbb{N}_0$. Each $\mathcal{G}_m$ is closed and contained in the compact set $\mathcal{W}_m^1 = \mathcal{C}^m$; hence, each $\mathcal{G}_m$ is compact. Because the lasso sequence contains only finitely many distinct pairs of consecutive goal sets, there exists $d_{\min} > 0$ such that

$$\mathrm{dist}\big(\mathcal{G}_{\zeta(n)}, \mathcal{G}_{\zeta(n+1)}\big) \geq d_{\min}, \qquad n \in \mathbb{N}_0. \tag{6.4.24}$$

Since f and g are continuous in system (6.2.1), $\mathcal{M}$ is finite, and $\mathcal{W}_m^1$ and $\mathcal{U}_{\text{box}}$ are compact, there exists a common finite velocity bound $\bar{v} > 0$ satisfying

$$\|f(x) + g(x)u\| \leq \bar{v}, \qquad x \in \mathcal{W}_m^1, \quad u \in \mathcal{U}_{\text{box}}, \quad m \in \mathcal{M}. \tag{6.4.25}$$

For every $n \geq 1$, $x(\tau_n) \in \mathcal{G}_{\zeta(n-1)}$, and $x(\tau_{n+1}) \in \mathcal{G}_{\zeta(n)}$. Therefore,

$$\begin{aligned} d_{\min} &\leq \|x(\tau_{n+1}) - x(\tau_n)\| \\ &\leq \int_{\tau_n}^{\tau_{n+1}} \|\dot{x}(t)\|\, dt \\ &\leq \bar{v}(\tau_{n+1} - \tau_n). \end{aligned} \tag{6.4.26}$$

It follows that $\tau_{n+1} - \tau_n \geq \frac{d_{\min}}{\bar{v}} > 0$, for $n \geq 1$, and hence $\tau_n \to \infty$.

Assumption 6.3.4 ensures finite variability on every finite transition interval. Together with $\tau_n \to \infty$, it follows that the complete closed-loop trajectory is of finite variability under L.

Finally, Assumption 6.3.1 gives $\mathcal{X}_{\zeta(n)} \setminus \left(\mathcal{A}_{\zeta(n)} \cup \mathcal{G}_{\zeta(n)}\right) \subseteq [\![\sigma^{\text{ref}}_{\zeta(n)}]\!] \cup [\![\emptyset]\!]$ and $\mathcal{G}_{\zeta(n)} \subseteq [\![\sigma^{\text{ref}}_{\zeta(n+1)}]\!]$, for $n \in \mathbb{N}_0$. Therefore, all the hypotheses of Proposition 6.3.5 are satisfied. Consequently, $\phi_{\boldsymbol{\kappa}}(\cdot; x_0) \models \varphi$. Since this holds for every $x_0 \in \widetilde{\text{Win}}_\sigma(\varphi)$, Definition 6.2.6 yields $\widetilde{\text{Win}}_\sigma(\varphi) \subseteq \text{Win}(\varphi)$. □

We summarize the proposed framework in Algorithm 1.

Remark 6.4.6. *We would like to emphasize that the proposed motion planning framework can accommodate trajectory-tracking deviations, including those caused by moderate exogenous noise or disturbances, through rapid online replanning. When the current state deviates from the planned trajectory during tracking, the current state can be used as a new initial condition, and the control synthesis can be re-executed to generate a new valid path. The formal satisfaction guarantee is restricted to the nominal model and holds provided that the new initial state remains inside the certified local winning set $\widetilde{\text{Win}}_\sigma(\mathcal{T}_m)$ associated with the current local specification $\mathcal{T}_m$. Under this condition, the replanned trajectory is guaranteed to satisfy φ.*

Algorithm 1: Patching CLBF-Based Control under $\mathrm{LTL}_{\setminus\bigcirc}$ Specification φ

Input: System (6.2.1) and reference word $\sigma^{\mathrm{ref}} = \sigma^{\mathrm{ref}}_{\mathrm{pre}}(\sigma^{\mathrm{ref}}_{\mathrm{suf}})^{\omega} \models \varphi$.

Output: Closed-loop trajectory and finite-memory control strategy.

$M_{\mathrm{pre}} \leftarrow 0$ if $\sigma^{\mathrm{ref}}_{\mathrm{pre}}$ is empty, else $\mathrm{len}(\sigma_{\mathrm{pre}})$;

$M_{\mathrm{suf}} \leftarrow \mathrm{len}(\sigma^{\mathrm{ref}}_{\mathrm{suf}})$; // $M_{\mathrm{suf}} \geq 1$ as the suffix is nonempty in lasso form

$M \leftarrow M_{\mathrm{pre}} + M_{\mathrm{suf}}$;

// Define the memory-index map ζ.

Function `MemoryIndex`(n)**:**

  if $0 \leq n < M_{\mathrm{pre}}$ **then**

    return n;

  else

    return $M_{\mathrm{pre}} + ((n - M_{\mathrm{pre}}) \bmod M_{\mathrm{suf}})$;

// Offline synthesis phase:

Construct transitions $\{\mathcal{T}_m\}_{m=0}^{M-1}$ that satisfy Assumption 6.3.1[1];

for $m = 0$ **to** $M - 1$ **do**

  Construct the safe ROA estimate $\mathcal{W}^1_m$ for $\mathcal{T}_m$ using (6.4.10) ;

  if $m \geq 1$ **then**

    Verify that $\mathcal{G}_{m-1}$ satisfies (6.4.21);

  if $m = M - 1$ **then**

    // Suffix wrap-around inclusion check

    Verify that $\mathcal{G}_{M-1} \subseteq \widetilde{\mathrm{Win}}(\mathcal{T}_{M_{\mathrm{pre}}})$; // $\mathcal{T}_{M_{\mathrm{pre}}} = \mathcal{T}_0$ if $M_{\mathrm{pre}} = 0$

  Store $\mathcal{W}^1_m$ and controller $u^m(\cdot)$;

// Online control phase:

$n \leftarrow 0$; // stage counter

$x(0) \leftarrow x_0 \in \widetilde{\mathrm{Win}}(\varphi)$;

while *execution continues* **do**

  $m \leftarrow$ `MemoryIndex`(n) ; // $m = \zeta(n)$

  Apply $u(t) = u^m(x(t))$;

  Evolve $\dot{x}(t) = f(x(t)) + g(x(t))u(t)$;

  if $x(t) \in \mathcal{G}_m$ **then**

    $n \leftarrow n + 1$;

[1] $\mathcal{T}_{M-1}$ is already the suffix wrap-around transition from the last suffix symbol to the first suffix symbol.

6.5 Case Studies

In this section, we evaluate the efficacy of the proposed approach on two simulations and one real experiment with the Crazyflie quadrotor. The patching CLBF step is done with the LyZNet [94] toolbox on a server node equipped with an Intel(R) Xeon(R) Gold 6326 CPU @ 2.90 GHz (32 cores). Comparisons were conducted on a server with 2× Intel Xeon Silver 4114 CPUs and 128 GB RAM, with each run allocated 2 CPU threads.

Case 1: Rigid Spacecraft Attitude Control. First, we consider the attitude tracking problem for a rigid spacecraft. The system dynamics are governed by Euler's rotational equations [67]:

$$J\dot{x} = -x \times Jx + u$$

where $J \in \mathbb{R}^{3\times 3}$ is the positive-definite, symmetric inertia matrix. For an asymmetric rigid body, we assume $J = \text{diag}(10, 15, 20)$. $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top \in [-1, 1]^3$ denotes the angular velocities of the spacecraft in the body-fixed frame, and $u = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^\top \in [-3, 3]^3$ denotes the input control torques. We consider the following LTL specification:

$$\varphi = \Diamond(\text{Green}) \wedge \Box(\neg \text{Red}_1) \wedge \Box(\neg \text{Red}_2), \tag{6.5.1}$$

which requires the system to eventually reach the goal set Green and always avoid the two avoidance sets, Red_1 and Red_2 in the state space.

A local quadratic CLF for this system can be obtained in the form $V(x) = x^\top Px$ by linearizing the system at the equilibrium point and solving the continuous-time algebraic Riccati equation with Q and R as identity matrices of appropriate dimensions. This identical CLF configuration is maintained throughout the subsequent numerical examples. For such systems with known CLFs, we only need to clearly define the barrier functions with respect to the red avoidance sets, and then the patching CLBF can be constructed easily by the method in Section 6.4.1. As shown in Fig. 6.1, we have $h_1 = 1 - ((x_1 - 1)^2 + (x_2 - 0.6)^2 + x_3^2 - 0.5^2)$ for Red_1 (upper right) and $h_2 = 1 - ((x_1 + 1)^2 + (x_2 + 1)^2 + x_3^2 - 0.5^2)$ for Red_2 (lower left), respectively. The goal set Green is defined as a ball region centered at the origin with a radius of 0.1. Then, we construct a smooth barrier function with $\tau = 5.2$, and consequently patch it with the CLF with $\varepsilon = 0.5$. All 20 trajectories, initialized from samples within the winning set, successfully realize the LTL specification with the computed controller.

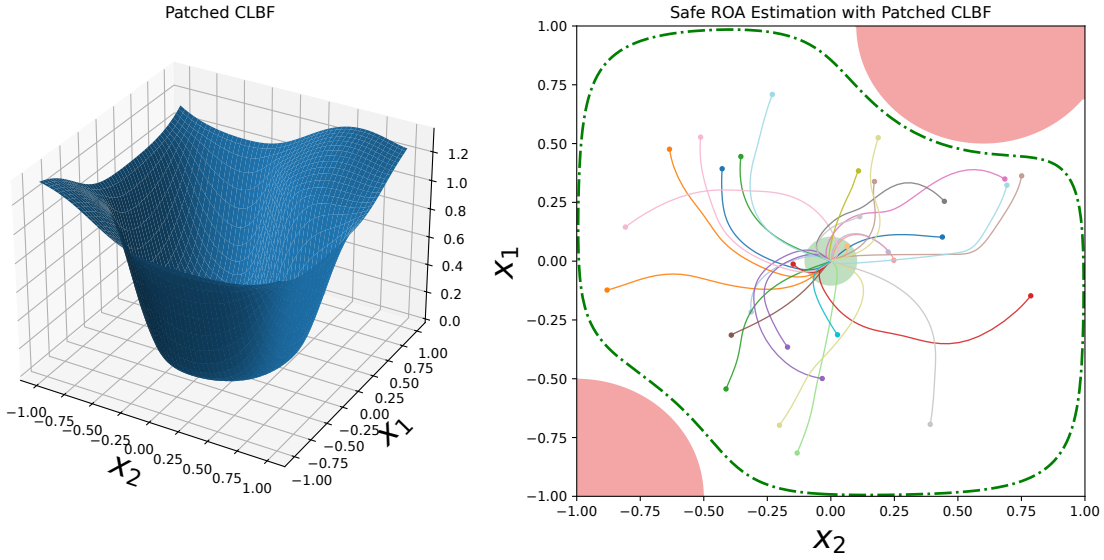


Figure 6.1: Visualization on the $x_3 = 0$ slice over the (x_1, x_2)-plane. Left: the certified CLBF for the LTL specification (6.5.1). Right: the projected winning-set approximation, indicated by the dashed green curve, and sampled state trajectories under $u \in [-3, 3]^3$. All state trajectories initialized in this approximation satisfy φ.

Case 2: Omnidirectional Robot Control. Second, we consider an omnidirectional robot [27] with state $x = [x_1, x_2, x_3]^\top$ and control input $u = [v_1, v_2, \omega]^\top \in [-3, 3]^3$. The dynamics are given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \cos x_3 & -\sin x_3 & 0 \\ \sin x_3 & \cos x_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \omega \end{bmatrix}.$$

As shown in Figure 6.2, the robot needs to complete a cyclic transfer task in a gated workspace, where gate_3 and gate_4 open automatically upon robot proximity and are thus omitted from the LTL specification. Starting from the processing station r_4, the robot alternates between two routes: one through r_5, r_2, and gate_1 to reach r_1, and the other through r_3, r_7, and gate_2 to reach r_6. In each route, the robot picks up a key before passing through the corresponding gate and then returns to r_4. The outbound and return routes

are denoted by $\mathfrak{y}_1, \mathfrak{y}_2$ and $\mathfrak{y}_1', \mathfrak{y}_2'$, respectively.

$$
\begin{aligned}
\mathfrak{y}_1 &= r_4 \wedge \Diamond\big(r_5 \wedge \Diamond(r_2 \wedge \Diamond(\text{gate}_1 \wedge \Diamond r_1))\big),\\
\mathfrak{y}_1' &= r_1 \wedge \Diamond\big(\text{gate}_1 \wedge \Diamond(r_2 \wedge \Diamond(r_5 \wedge \Diamond r_4))\big),\\
\mathfrak{y}_2 &= r_4 \wedge \Diamond\big(r_3 \wedge \Diamond(r_7 \wedge \Diamond(\text{gate}_2 \wedge \Diamond r_6))\big),\\
\mathfrak{y}_2' &= r_6 \wedge \Diamond\big(\text{gate}_2 \wedge \Diamond(r_7 \wedge \Diamond(r_3 \wedge \Diamond r_4))\big).
\end{aligned}
$$

The overall LTL specification requires the two transfer cycles to be repeatedly executed in an alternating manner:

$$
\varphi = \Box\left(\neg \text{obs}\right) \wedge \Box\Diamond\left(\mathfrak{y}_1 \wedge \Diamond\left(\mathfrak{y}_1' \wedge \Diamond\left(\mathfrak{y}_2 \wedge \Diamond\ \mathfrak{y}_2'\right)\right)\right),
$$

where $\text{obs} = \bigvee_{i=1}^{9} \text{obs}_i$. In this task, we need to decompose the overall LTL specification into 16 transitions as described in Section 6.3. Regarding the barrier functions, we use a *log-sum-exp* smooth approximation again, and then construct the barrier functions for each $\mathcal{T}_m$ with $\tau = 4$ as well as a 0.5 safety margin for the obstacles to further ensure the robot's safety. As illustrated in Figure 6.2, for the first subtask $\varphi_1 = r_4 \wedge \Diamond(r_5)$, we can compute the approximate winning set $\widetilde{\text{Win}}(\varphi)$, shown as the purple hatched region. As an example, the orange hatched region represents the approximate winning set $\widetilde{\text{Win}}(\mathcal{T}_{10})$ for the subtask $\varphi_{10} = r_3 \wedge \Diamond(r_7)$, which captures almost the entire winning set of this local specification. Consequently, for each subtask, we can immediately synthesize a controller for any states within the approximated winning set, which provides robustness and flexibility without solving QPs on the fly.

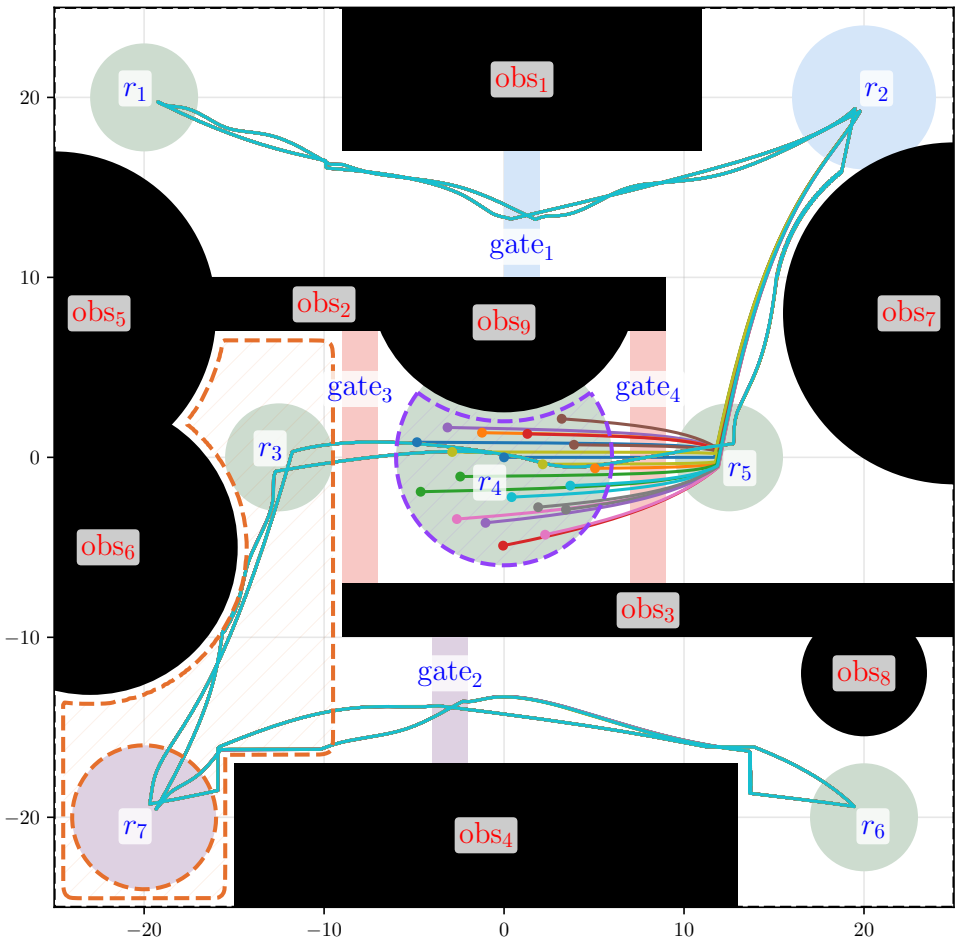


Figure 6.2: Twenty trajectories of the omnidirectional robot initialized from different states in a complex planar environment under the LTL specification using the proposed method. The purple and orange hatched regions denote $\widetilde{\text{Win}}(\varphi)$ and $\widetilde{\text{Win}}(\mathcal{T}_{10})$, respectively. The controller is executed at 50 Hz.

Table 6.1: Comparison of trajectory performance and computational cost. Values are reported as mean ± sample standard deviation over 20 trajectories.

Method	$\bar{T}$ (s)	$\bar{T}_{\text{step}}$ (ms)	$\bar{E}_{\text{step}}$
Patched CLBF	$\mathbf{15.47 \pm 0.08}$	$\mathbf{0.41 \pm 0.00}$	$\mathbf{0.37 \pm 0.01}$
QP-CLF-CBF	26.51 ± 0.43	1.67 ± 0.02	0.89 ± 0.00

With 20 different initial conditions sampled in the approximated winning set, we further compare the proposed controller against the standard CBF-QP baseline with the CLF-based controller via Sontag's formula. For each trajectory $i \in \{1, \ldots, 20\}$, let T_i, M_i, and $u_i(k)$ denote the CPU runtime, the number of control steps, and the control input at the k-th step, respectively. Since M_i varies with the initial condition, we average the per-trajectory metrics so that each trajectory contributes equally. Specifically, we compute the CPU runtime as $\bar{T} = \frac{1}{20}\sum_{i=1}^{20} T_i$, the CPU runtime per control step as $\bar{T}_{\text{step}} = \frac{1}{20}\sum_{i=1}^{20} \frac{T_i}{M_i}$, and the control-energy cost per control step as $\bar{E}_{\text{step}} = \frac{1}{20}\sum_{i=1}^{20} \frac{1}{M_i}\sum_{k=1}^{M_i} \|u_i(k)\|_2^2$. As shown in Table 6.1, the proposed controller achieves higher computational efficiency by avoiding QP solving at every control step, thereby reducing the overall computation time. The proposed controller also yields a lower control-energy cost.

Additionally, Figure 6.3 illustrates that the maximum absolute values over all 20 trajectories,

$$|v_{1,\max}(t)| = \max_{i=1,\ldots,20} |v_{1,i}(t)|, \quad |v_{2,\max}(t)| = \max_{i=1,\ldots,20} |v_{2,i}(t)|, \quad |\omega_{\max}(t)| = \max_{i=1,\ldots,20} |\omega_i(t)|,$$

stay below the bound throughout the execution, confirming that the synthesized CLBF-based controller satisfies the input box constraint $u = [v_1, v_2, \omega]^\top \in [-3, 3]^3$.

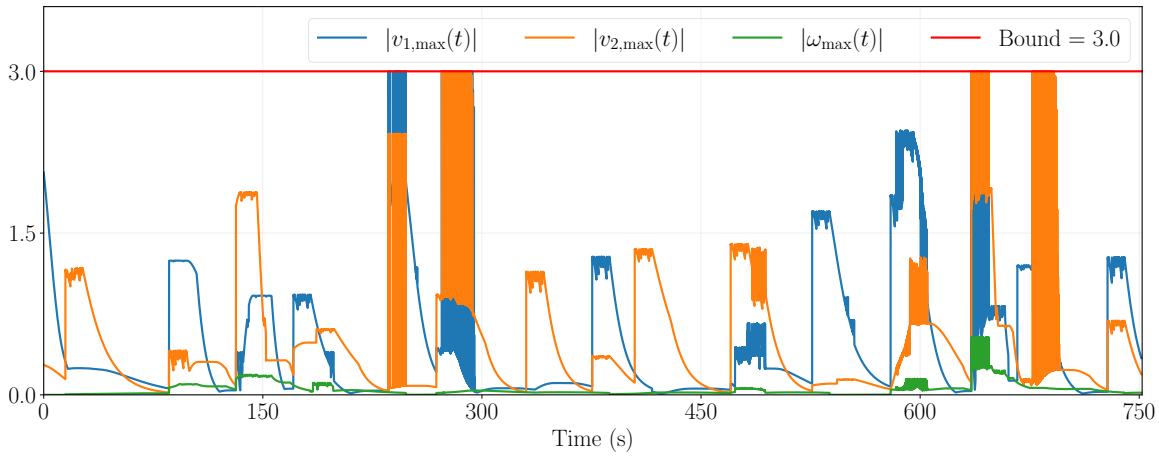


Figure 6.3: Maximum absolute control inputs over 20 trajectories for Case 2. The plotted quantities $|v_{1,\max}(t)|$, $|v_{2,\max}(t)|$, and $|\omega_{\max}(t)|$ remain within the prescribed input bound 3.0, showing that the synthesized controller satisfies the box constraint $u \in [-3, 3]^3$.

Case 3: Crazyflie Quadrotor Experiment. Finally, we test our method on a Crazyflie quadrotor for a planar task with the following LTL specification:

$$\varphi = a_0 \wedge \Diamond a_1 \wedge \Box\Diamond a_2 \wedge \Box\Diamond a_3 \wedge \Box(\neg \text{obs}) \tag{6.5.2}$$

where $\text{obs} = \vee_{i=1}^{3} \text{obs}_i$. We use the single integrator with $u \in [-0.3, 0.3]^2$ for planning with the proposed method, while the tracking is done with its default controller. This specification requires the trajectory to start from region a_0, avoid the three obstacle regions, eventually visit region a_1, and repeatedly visit regions a_2 and a_3. The simulation scenario in Webots[2] and resulting trajectory are shown in Figure 6.4, and the corresponding real experiment scenario and trajectory are shown in Figure 6.5. The results demonstrate the practical executability of the generated temporal-logic task sequence while respecting control constraints. Although the simplified planning model used in the real experiment may introduce deviations, the recorded real-world trajectory follows the planned trajectory and completes the specified task sequence, demonstrating the practical trackability of the generated trajectory.

[2] https://www.cyberbotics.com/

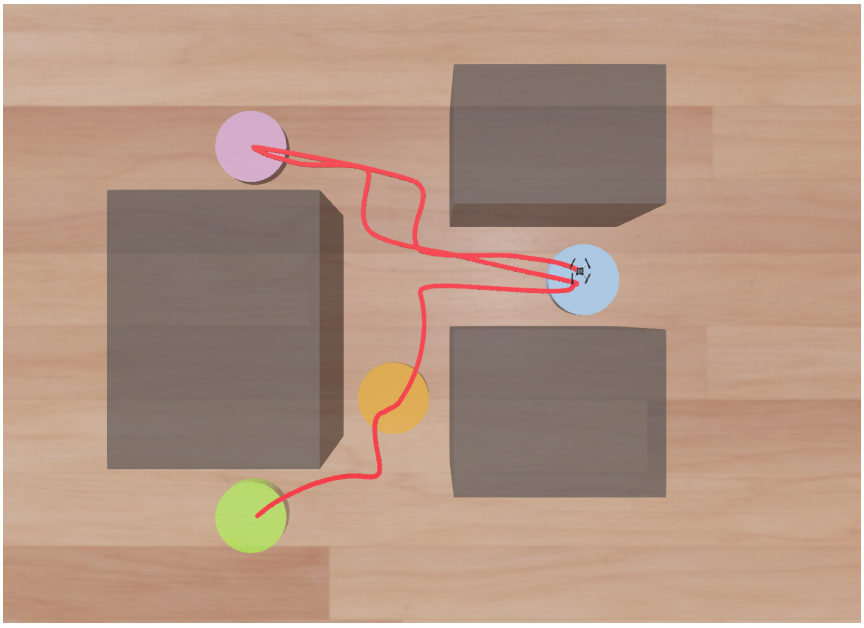


Figure 6.4: Webots simulation scenario and executed trajectory for the Crazyflie quadrotor. The black, green, orange, blue, and pink regions correspond to obstacles, a_0, a_1, a_2, and a_3, respectively. The red curve denotes the trajectory executed by the quadrotor.

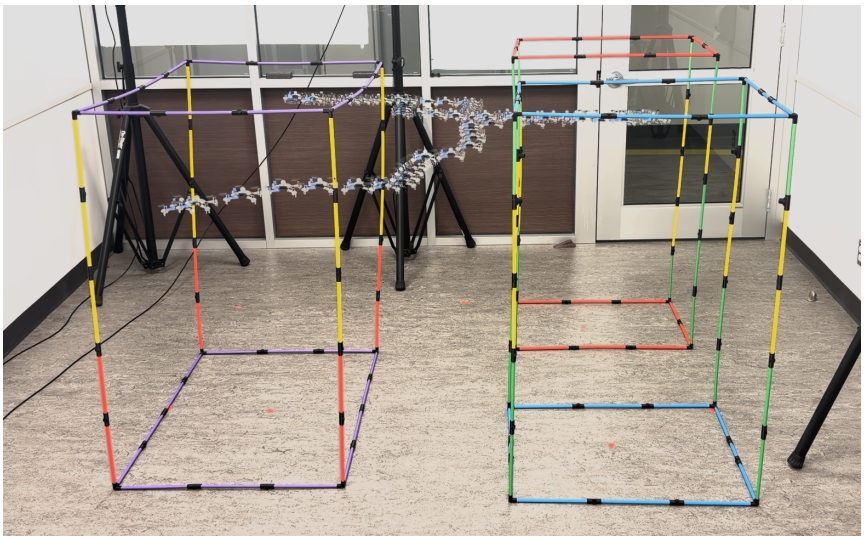


Figure 6.5: The real experiment and executed trajectory for the Crazyflie quadrotor. The colored rectangular boxes represent obstacles.

6.6 Summary

This work presents a patched-CLBF framework for synthesizing switching feedback controllers for continuous-state systems subject to $\text{LTL}_{\setminus\bigcirc}$ specifications and bounded inputs. By linking feasible temporal logic trajectories to CLBF level sets, the method decomposes the synthesis problem into a sequence of safe stabilization tasks and enables efficient online execution. Simulations and a Crazyflie quadrotor experiment demonstrate the effectiveness of the approach. Future work will focus on reducing the conservativeness of the certificate construction and improving scalability to higher-dimensional systems, for which certificate search and patch verification become more challenging. The current framework assumes that a reference word satisfying the specification is given in advance. Integrating certificate verification with Büchi automaton search could enable the direct synthesis of both a satisfying word and its corresponding controller.

Moreover, the proposed framework is sound but not complete: successful synthesis guarantees specification satisfaction, while failure does not preclude the existence of satisfying controllers. This incompleteness stems from the conservative, sufficient-condition-based structure of certificate verification. Recent work derives conditions that are both necessary and sufficient for specific formulations over finite traces, but these conditions do not extend to general LTL synthesis for systems with continuous state spaces [114]. By contrast, the proposed framework provides the formal guarantee that, whenever the procedure succeeds, the synthesized closed-loop behavior satisfies the given specification [145].

Chapter 7

Conclusions and Future Work

This thesis studies planning and control synthesis for nonlinear systems under temporal logic specifications. The main focus is to explore how task-level logical requirements can be systematically incorporated into continuous-time motion planning and feedback control design. To conclude the thesis, this chapter reviews the main contributions and discusses related opportunities for future research.

Continuous-Time Signal Temporal Logic Planning

For continuous-time motion planning under STL specifications, this thesis investigates trajectory representations based on Bézier curves for encoding robustness margins and formulating MIP-based optimization problems. Chapter 3 established soundness guarantees for the segment-level abstraction of STL satisfaction. This framework was further extended in Chapter 5 to the multi-agent setting, where inter-agent safety constraints were incorporated to ensure collision avoidance while preserving STL satisfaction. These results not only support the generation of continuous-time reference trajectories that satisfy STL specifications, but also exploit a time-varying robustness margin to ensure that the corresponding tracking trajectories satisfy the same specifications.

The main limitations of the current framework are conservativeness and scalability. Although the proposed time-varying robustness margin is less conservative than using a fixed margin over the entire time horizon, the Bézier-control-point-based construction can still be affected by the number and distribution of control points. In some cases, increasing the number of control points may reduce the available robustness margin. Future work could investigate alternative polynomial trajectory representations and robustness-margin conditions to further reduce conservativeness while preserving formal satisfaction guarantees.

Scalability is another important limitation, as the computational cost of the MIP formulation grows with both the complexity of the STL formula and the number of agents. While Chapter 5 exploited the backward structure of the specification to reduce the computational burden in the multi-agent setting, further improvements are needed for larger-scale problems. In addition, the current soundness guarantees rely on exact binary encodings of STL predicates and logical operations, which makes extensions to approximate or smooth robustness formulations nontrivial. Future work could therefore explore scalable approaches for temporal-logic-constrained systems, including decomposition-based, distributed, and learning-based methods [75, 143, 110]. Another promising direction is to investigate alternative STL robustness measures [106, 115, 89] that better balance computational efficiency and formal guarantees.

Robust Tracking Control for Temporal Logic Planning

Another key problem considered in this thesis is robust tracking control for nonlinear systems, with the goal of ensuring that the resulting tracking trajectories satisfy STL specifications in the presence of initial tracking errors. Chapter 4 focused on high-dimensional nonlinear quadrotor systems, building on geometric tracking control on SE(3) and incorporating matrix gains to derive formal tracking bounds. The same controller structure was further used in the multi-agent setting in Chapter 5. These bounds quantify the deviation between the planned and actual trajectories, thereby supporting STL satisfaction during closed-loop execution.

However, the geometric tracking controller relies on an accurate quadrotor model. In the presence of model uncertainties or external disturbances, the derived bounds may no longer fully capture the actual tracking errors. Future work could therefore investigate more general robust tracking frameworks, such as contraction-metric-based control [103], robust and adaptive control [119], or control Lyapunov function-based tracking control [99], to improve robustness under model uncertainties and broaden applicability across nonlinear systems.

Certified Feedback Controller under Temporal Logic Specifications

Another direction explored in this thesis is certified feedback control for temporal logic planning. Chapter 6 studied patching CLBFs as certified feedback controllers for characterizing winning sets from which the LTL specifications can be satisfied. This framework operates directly in the continuous state space and supports online implementation.

However, scalability remains a challenge for this approach in high-dimensional systems, especially during the verification step. In addition, the current construction provides soundness guarantees, while completeness is generally difficult to guarantee. Future research could investigate how counterexample-guided synthesis [1, 104] and neural network

verification [60, 57] can be adapted to improve the construction and verification of patching CLBFs under LTL specifications.

Real-World Experiments

The final aspect considered in this thesis is the real-world implementation of the proposed planning and control frameworks. Chapter 6 presented a flight experiment on the Crazyflie platform, where the quadrotor tracked the trajectories generated by the patching CLBF-based planning framework and satisfied the corresponding LTL specifications.

A natural direction for future work is to extend the proposed LTL-based framework to uncertainty-aware robotic planning and control settings, including realistic disturbances, sensing noise, and communication delays. These factors may widen the gap between formally synthesized plans and the behavior of physical robots. Possible extensions include temporal logic control under stochastic uncertainty [29], as well as semantic-map-based temporal logic planning under perceptual and environmental uncertainty [45, 59].

References

[1] Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, and Andrea Peruffo. Formal synthesis of lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, 2021.

[2] Houssam Abbas, Yash Vardhan Pant, and Rahul Mangharam. Temporal logic robustness for general signal classes. In *Proc. of HSCC*, pages 45–56, 2019.

[3] Vangalur S Alagar and Kasilingam Periyasamy. *Specification of Software Systems*. Springer, 2011.

[4] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[5] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[6] Tomáš Babiak, Martin Blicha, Aaron Bradley, Alexandre Duret-Lutz, Hubert Garavel, Jan Křetínský, Dejan Ničković, et al. The Hanoi omega-automata format. In *International Conference on Computer Aided Verification (CAV)*, pages 479–486. Springer, 2015.

[7] Changmin Baek and Kyunghoon Cho. Deep learning-enhanced sampling-based path planning for ltl mission specifications. *Sensors*, 24(10):2998, 2024.

[8] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.

[9] Ezio Bartocci, Jyotirmoy Deshmukh, Alexandre Donzé, Georgios Fainekos, Oded Maler, Dejan Nickovic, and Sriram Sankaranarayanan. Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications. In *Lectures on Runtime Verification*, volume 10457 of *Lecture Notes in Computer Science*, pages 135–175. Springer, 2018.

[10] Charalampos P Bechlioulis and George A Rovithakis. Robust adaptive control of feedback linearizable mimo nonlinear systems with prescribed performance. *IEEE Transactions on Automatic Control*, 53(9):2090–2099, 2008.

[11] Calin Belta and Sadra Sadraddini. Formal methods for control synthesis: An optimization perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:115–140, 2019.

[12] Taylor Bergeron, Zachary Serlin, and Kevin Leahy. Comp-ltl: Temporal logic planning via zero-shot policy composition. *CoRR*, abs/2408.04215, 2024.

[13] Graeme Best, Rohit Garg, John Keller, Geoffrey A Hollinger, and Sebastian Scherer. Multi-robot, multi-sensor exploration of multifarious environments with full mission aerial autonomy. *The International Journal of Robotics Research*, 43(4):485–512, 2024.

[14] Andrea Bisoffi and Dimos V. Dimarogonas. A hybrid barrier certificate approach to satisfy linear temporal logic specifications. In *2018 Annual American Control Conference (ACC)*, pages 634–639. IEEE, 2018.

[15] Francesco Borrelli, Paolo Falcone, Tamas Keviczky, Jahan Asgari, and Davor Hrovat. Mpc-based approach to active steering for autonomous vehicle systems. *International Journal of Vehicle Autonomous Systems*, 3(2-4):265–291, 2005.

[16] Xiangwei Bu. Prescribed performance control approaches, applications and challenges: A comprehensive survey. *Asian Journal of Control*, 25(1):241–261, 2023.

[17] Francesco Bullo and Andrew D. Lewis. *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*, volume 49 of *Texts in Applied Mathematics*. Springer, New York, 2004.

[18] Ali Tevfik Büyükkoçak, Derya Aksaray, and Yasin Yazicioglu. Distributed planning of multi-agent systems with coupled temporal logic specifications. In *AIAA Scitech 2021 Forum*, page 1123, 2021.

[19] Fabio Camilli, Lars Grüne, and Fabian Wirth. Control Lyapunov functions and zubov's method. *SIAM Journal on Control and Optimization*, 47(1):301–326, 2008.

[20] Maria Charitidou and Dimos V Dimarogonas. Distributed MPC with continuous-time STL constraint satisfaction guarantees. *IEEE Control Systems Letters*, 8:211–216, 2024.

[21] Nalin A. Chaturvedi, Amit K. Sanyal, and N. Harris McClamroch. Rigid-body attitude control. *IEEE Control Systems Magazine*, 31(3):30–51, 2011.

[22] Fei Chen and Dimos V Dimarogonas. Funnel-based cooperative control of leader-follower multi-agent systems under signal temporal logic specifications. In *Proc. of ECC*, pages 906–911. IEEE, 2022.

[23] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, 1999.

[24] Hongkai Dai, Chuanrui Jiang, Hongchao Zhang, and Andrew Clark. Verification and synthesis of compatible control lyapunov and control barrier functions. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 8178–8185. IEEE, 2024.

[25] Jian S. Dai. Euler–rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory*, 92:144–152, 2015.

[26] Adnan Darwiche. Decomposable negation normal form. *Journal of the ACM*, 48(4):608–647, 2001.

[27] Ratnangshu Das and Pushpak Jagtap. Prescribed-time reach-avoid-stay specifications for unknown systems: A spatiotemporal tubes approach. *IEEE Control Systems Letters*, 8:946–951, 2024.

[28] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[29] Xuchu Ding, Stephen L. Smith, Calin Belta, and Daniela Rus. Optimal control of markov decision processes with linear temporal logic constraints. *IEEE Transactions on Automatic Control*, 59(5):1244–1257, 2014.

[30] Adel Dokhanchi, Bardh Hoxha, and Georgios Fainekos. On-line monitoring for temporal logic robustness. In *International Conference on Runtime Verification*, pages 231–246. Springer, 2014.

[31] Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *Proc. of FORMATS*, 2010.

[32] Alexandre Duret-Lutz, Alexandre Lewkowicz, Amaury Fauchille, Thibaud Michaud, Etienne Renault, and Laurent Xu. Spot 2.0—a framework for ltl and ω-automata manipulation. In *Automated Technology for Verification and Analysis*, volume 9938 of *Lecture Notes in Computer Science*, pages 122–129. Springer, 2016.

[33] Joe Eappen, Zikang Xiong, Dipam Patel, Aniket Bera, and Suresh Jagannathan. Scaling safe multi-agent control for signal temporal logic specifications. *arXiv preprint arXiv:2501.05639*, 2025.

[34] Rüdiger Ehlers. Short witnesses and accepting lassos in ω-Automata. In Adrian-Horia Dediu, Henning Fernau, and Carlos Martín-Vide, editors, *Language and Automata Theory and Applications*, volume 6031 of *Lecture Notes in Computer Science*, pages 261–272, Berlin, Heidelberg, 2010. Springer.

[35] Mohamed Elhesasy, Tarek N Dief, Mohammed Atallah, Mohamed Okasha, Mohamed M Kamra, Shigeo Yoshida, and Mostafa A Rushdi. Non-linear model predictive control using casadi package for trajectory tracking of quadrotor. *Energies*, 16(5):2143, 2023.

[36] Georgios E. Fainekos, Antoine Girard, Hadas Kress-Gazit, and George J. Pappas. Temporal logic motion planning for dynamic robots. *Automatica*, 45(2):343–352, 2009.

[37] Georgios E Fainekos and George J Pappas. Robustness of temporal logic specifications. In *International Workshop on Formal Approaches to Software Testing*, pages 178–192. Springer, 2006.

[38] Georgios E Fainekos and George J Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42):4262–4291, 2009.

[39] Chuchu Fan, Kristina Miller, and Sayan Mitra. Fast and guaranteed safe controller synthesis for nonlinear vehicle models. In *Computer Aided Verification*, pages 629–652, Cham, 2020. Springer International Publishing.

[40] Samira S Farahani, Rupak Majumdar, Vinayak S Prabhu, and Sadegh Esmaeil Zadeh Soudjani. Shrinking horizon model predictive control with chance-constrained signal temporal logic specifications. In *Proc. of ACC*, pages 1740–1746. IEEE, 2017.

[41] Samira S. Farahani, Vasumathi Raman, and Richard M. Murray. Robust model predictive control for signal temporal logic synthesis. *IFAC-PapersOnLine*, 48(27):323–328, 2015.

[42] Gerald Farin, Josef Hoschek, and M-S Kim. *Handbook of computer aided geometric design*. Elsevier, 2002.

[43] Thilina Fernando, Jiten Chandiramani, Taeyoung Lee, and Hector Gutierrez. Robust adaptive geometric tracking controls on SO(3) with an application to the attitude dynamics of a quadrotor UAV. In *2011 50th IEEE conference on decision and control and European control conference*, pages 7380–7385. IEEE, 2011.

[44] Victor Freire and Xiangru Xu. Flatness-based quadcopter trajectory planning and tracking with continuous-time safety guarantees. *IEEE Transactions on Control Systems Technology*, 31(6):2319–2334, 2023.

[45] Jie Fu, Nikolay Atanasov, Ufuk Topcu, and George J. Pappas. Optimal temporal logic planning in probabilistic semantic maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3690–3697, 2016.

[46] Jean Gallier and Dianna Xu. Computing exponentials of skew-symmetric matrices and logarithms of orthogonal matrices. *International Journal of Robotics and Automation*, 18(1):10–20, 2003.

[47] Kanishke Gamagedara, Mahdis Bisheban, Evan Kaufman, and Taeyoung Lee. Geometric controls of a quadrotor UAV with decoupled yaw control. In *Proc. of ACC*, pages 3285–3290. IEEE, 2019.

[48] Sicun Gao, Soonho Kong, and Edmund M Clarke. dReal: an SMT solver for nonlinear theories over the reals. In *Proc. of CADE*, pages 208–214, 2013.

[49] Paul Gastin and Denis Oddoux. Fast LTL to Büchi automata translation. In *International Conference on Computer Aided Verification (CAV)*, pages 53–65. Springer, 2001.

[50] Yann Gilpin, Vince Kurtz, and Hai Lin. A smooth robustness measure of signal temporal logic for symbolic control. *IEEE Control Systems Letters*, 5(1):241–246, January 2021.

[51] Weibin Gu, Stefano Primatesta, and Alessandro Rizzo. Robust adaptive control for aggressive quadrotor maneuvers via SO(3) and backstepping techniques. *Robotics and Autonomous Systems*, page 104942, 2025.

[52] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2026.

[53] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, volume 222 of *Graduate Texts in Mathematics*. Springer, New York, 2 edition, 2013.

[54] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2012.

[55] Yong Huang, Zhiyan Chen, Jing Chu, Haoran Wang, and Siliang Sun. Long-term path planning with optimal deployment of a charging station for monitoring photovoltaic solar farms. *Scientific Reports*, 14:17279, 2024.

[56] Davide Invernizzi and Marco Lovera. Geometric tracking control of a quadcopter tiltrotor UAV. *IFAC-PapersOnLine*, 50(1):11565–11570, 2017.

[57] Radoslav Ivanov, James Weimer, Rajeev Alur, George J. Pappas, and Insup Lee. Verisig: Verifying safety properties of hybrid systems with neural network controllers. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 169–178. ACM, 2019.

[58] Aris Kanellopoulos, Filippos Fotiadis, Chuangchuang Sun, Zhe Xu, Kyriakos G. Vamvoudakis, Ufuk Topcu, and Warren E. Dixon. Temporal-logic-based intermittent, optimal, and safe continuous-time learning for trajectory tracking. In *60th IEEE Conference on Decision and Control (CDC)*, pages 4172–4177. IEEE, 2021.

[59] Yiannis Kantaros, Samarth Kalluraya, Qi Jin, and George J. Pappas. Perception-based temporal logic planning in uncertain semantic maps. *IEEE Transactions on Robotics*, 38(4):2536–2556, 2022.

[60] Guy Katz, Derek A. Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, David L. Dill, Mykel J. Kochenderfer, and Clark Barrett. The marabou framework for verification and analysis of deep neural networks. In *Computer Aided Verification*, pages 443–452. Springer, 2019.

[61] Milad Kazemi, Rupak Majumdar, Mahmoud Salamati, Sadegh Soudjani, and Ben Wooding. Data-driven abstraction-based control synthesis. *Nonlinear Analysis: Hybrid Systems*, 52:101467, 2024.

[62] Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

[63] Marius Kloetzer and Calin Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1):287–297, 2008.

[64] Praneeth Kolapalli, Stephen L. Smith, and Yash Vardhan Pant. Drive-by-logic: Trajectory generation for nonholonomic ground robots with signal temporal logic objectives. In *2025 IEEE Conference on Control Technology and Applications (CCTA)*, pages 200–207. IEEE, 2025.

[65] Hadas Kress-Gazit, Georgios E. Fainekos, and George J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, 2009.

[66] Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman. Synthesis for robots: Guarantees and feedback for robot behavior. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:211–236, may 2018.

[67] M. Krstic and P. Tsiotras. Inverse optimal stabilization of a rigid spacecraft. *IEEE Transactions on Automatic Control*, 44(5):1042–1049, 1999.

[68] Vince Kurtz and Hai Lin. Trajectory optimization for high-dimensional nonlinear systems under stl specifications. *IEEE Control Systems Letters*, 5(4):1429–1434, 2020.

[69] Vince Kurtz and Hai Lin. A more scalable mixed-integer encoding for metric temporal logic. *IEEE Control Systems Letters*, 6:1718–1723, 2021.

[70] Vince Kurtz and Hai Lin. Temporal logic motion planning with convex optimization via graphs of convex sets. *IEEE Transactions on Robotics*, 39(5):3791–3804, 2023.

[71] Vincent Kurtz and Hai Lin. Mixed-integer programming for signal temporal logic with fewer binary variables. *IEEE Control Systems Letters*, 6:2635–2640, 2022.

[72] Morteza Lahijanian, Sean B. Andersson, and Calin Belta. Temporal logic motion planning and control with probabilistic satisfaction guarantees. *IEEE Transactions on Robotics*, 28(2):396–409, 2012.

[73] Steven M LaValle. *Planning Algorithms*. Cambridge University Press, May 2006.

[74] Nhan-Khanh Le, Erfaun Noorani, Sandra Hirche, and John Baras. Time-robust path planning with piece-wise linear trajectory for signal temporal logic specifications. *arXiv preprint arXiv:2403.10735*, 2024.

[75] Kevin Leahy, Austin Jones, and Cristian-Ioan Vasile. Fast decomposition of temporal logic specifications for heterogeneous teams. *IEEE Robotics and Automation Letters*, 7(2):2297–2304, 2022.

[76] Taeyoung Lee. Robust adaptive attitude tracking on SO(3) with an application to a quadrotor UAV. *IEEE Transactions on Control Systems Technology*, 21(5):1924–1930, 2012.

[77] Taeyoung Lee. Global exponential attitude tracking controls on SO(3). *IEEE Transactions on Automatic Control*, 60(10):2837–2842, 2015.

[78] Taeyoung Lee, Melvin Leok, and N. Harris McClamroch. Geometric tracking control of a quadrotor uav on se(3). In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 5420–5425. IEEE, 2010.

[79] Karen Leung, Nikos Aréchiga, and Marco Pavone. Backpropagation through signal temporal logic specifications: Infusing logical structure into gradient-based methods. *The International Journal of Robotics Research*, 42(6):356–370, 2023.

[80] Horacio Leyva, Julio Solis-Daun, and Rodolfo Suárez. Global clf stabilization of systems with control inputs constrained to a hyperbox. *SIAM Journal on Control and Optimization*, 51(1):745–766, 2013.

[81] Yinan Li. *Robustly complete temporal logic control synthesis for nonlinear systems.* PhD thesis, University of Waterloo, 2019.

[82] Yinan Li and Jun Liu. ROCS: A robustly complete control synthesis tool for nonlinear dynamical systems. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control*, HSCC '18, pages 130–135, New York, NY, USA, 2018. ACM.

[83] Yinan Li, Jun Liu, and Necmiye Ozay. Computing finite abstractions with robustness margins via local reachable set over-approximation. *IFAC-PapersOnLine*, 48(27):1–6, 2015.

[84] Yinan Li, Zhibing Sun, and Jun Liu. A specification-guided framework for temporal logic control of nonlinear systems. *IEEE Transactions on Automatic Control*, 68(4):2002–2017, 2023.

[85] Kaier Liang, Mingyu Cai, and Cristian-Ioan Vasile. Control barrier function for linearizable systems with high relative degrees from signal temporal logics: A reference governor approach. In *Proc. of ACC*, pages 1676–1681, 2024.

[86] Lars Lindemann and Dimos V. Dimarogonas. Control barrier functions for signal temporal logic tasks. *IEEE Control Systems Letters*, 3(1):96–101, 2019.

[87] Lars Lindemann and Dimos V Dimarogonas. Decentralized control barrier functions for coupled multi-agent systems under signal temporal logic tasks. In *Proc. of ECC*, pages 89–94. IEEE, 2019.

[88] Lars Lindemann and Dimos V Dimarogonas. Feedback control strategies for multi-agent systems under a fragment of signal temporal logic tasks. *Automatica*, 106:284–293, 2019.

[89] Lars Lindemann and Dimos V. Dimarogonas. Robust control for signal temporal logic specifications using discrete average space robustness. *Automatica*, 101:377–387, 2019.

[90] Lars Lindemann and Dimos V Dimarogonas. Funnel control for fully actuated systems under a fragment of signal temporal logic specifications. *Nonlinear Analysis: Hybrid Systems*, 39:100973, 2021.

[91] Lars Lindemann, Christos K Verginis, and Dimos V Dimarogonas. Prescribed performance control for signal temporal logic specifications. In *Proc. of CDC*, pages 2997–3002. IEEE, 2017.

[92] Jun Liu. Formal verification of control lyapunov-barrier functions for safe stabilization with bounded controls. *arXiv preprint arXiv:2511.10510*, 2025.

[93] Jun Liu and Maxwell Fitzsimmons. Computing control lyapunov-barrier functions: Softmax relaxation and smooth patching with formal guarantees. *arXiv preprint arXiv:2510.02223*, 2025.

[94] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. LyZNet: A lightweight Python tool for learning and verifying neural Lyapunov functions and regions of attraction. In *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control*, 2024.

[95] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. Physics-informed neural network Lyapunov functions: PDE characterization, learning, and verification. *Automatica*, 175:112193, 2025.

[96] Jun Liu and Necmiye Ozay. Finite abstractions with robustness margins for temporal logic-based control synthesis. *Nonlinear Analysis: Hybrid Systems*, 22:1–15, 2016.

[97] Jun Liu, Necmiye Ozay, Ufuk Topcu, and Richard M. Murray. Synthesis of reactive switching protocols from temporal logic specifications. *IEEE Transactions on Automatic Control*, 58(7):1771–1785, 2013.

[98] Siyuan Liu, Adnane Saoud, and Dimos V Dimarogonas. Controller synthesis of collaborative signal temporal logic tasks for multi-agent systems via assume-guarantee contracts. *IEEE Transactions on Automatic Control*, 2025.

[99] Zhen-Guo Liu, Lingrong Xue, and Zong-Yao Sun. A new robust adaptive tracking strategy to uncertain time-delay nonlinear systems with a general form. *Automatica*, 146:110560, 2022.

[100] Matt Luckcuck, Marie Farrell, Louise A Dennis, Clare Dixon, and Michael Fisher. Formal specification and verification of autonomous robotic systems: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–41, 2019.

[101] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

[102] Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, volume 3253 of *Lecture Notes in Computer Science*, pages 152–166. Springer, 2004.

[103] Ian R. Manchester and Jean-Jacques E. Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Transactions on Automatic Control*, 62(6):3046–3053, 2017.

[104] Daniele Masti, Filippo Fabiani, Giorgio Gnecco, and Alberto Bemporad. Counterexample guided inductive synthesis of control lyapunov functions for uncertain systems. *IEEE Control Systems Letters*, 7:1309–1314, 2023.

[105] Noushin Mehdipour, Cristian-Ioan Vasile, and Calin Belta. Arithmetic-geometric mean robustness for control from signal temporal logic specifications. In *Proc. of ACC*, pages 1690–1695. IEEE, 2019.

[106] Noushin Mehdipour, Cristian-Ioan Vasile, and Calin Belta. Average-based robustness for continuous-time signal temporal logic. In *Proc. of CDC*, pages 5312–5317. IEEE, 2019.

[107] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, pages 2520–2525. IEEE, 2011.

[108] Yiming Meng and Jun Liu. Towards learning and verifying maximal lyapunov-barrier functions with a zubov pde formulation. *arXiv preprint arXiv:2511.09523*, 2025.

[109] Yiming Meng and Jun Liu. Characterization of safe stabilization and control lyapunov-barrier functions via zubov equation formulation. *arXiv preprint arXiv:2604.00941*, 2026.

[110] Yue Meng and Chuchu Fan. Signal temporal logic neural predictive control. *IEEE Robotics and Automation Letters*, 8(12):7719–7726, 2023.

[111] Richard M Murray, Zexiang Li, and S. Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[112] Petter Nilsson and Aaron D. Ames. Barrier functions: Bridging the gap between planning from specifications and safety-critical control. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 765–772. IEEE, 2018.

[113] Luyao Niu and Andrew Clark. Control barrier functions for abstraction-free control synthesis under temporal logic constraints. In *59th IEEE Conference on Decision and Control (CDC)*, pages 816–823. IEEE, 2020.

[114] Luyao Niu, Andrew Clark, and Radha Poovendran. Necessary and sufficient conditions for satisfying linear temporal logic constraints using control barrier certificates. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 8589–8595. IEEE, 2023.

[115] Yash Vardhan Pant, Houssam Abbas, and Rahul Mangharam. Smooth operator: Control using the smooth robustness of temporal logic. In *Proc. of CCTA*, pages 1235–1240. IEEE, 2017.

[116] Yash Vardhan Pant, Houssam Abbas, Rhudii A Quaye, and Rahul Mangharam. Fly-by-logic: Control of multi-drone fleets with temporal logic objectives. In *Proc. of ICCPS*, pages 186–197. IEEE, 2018.

[117] Yash Vardhan Pant, Max Z. Li, Alena Rodionova, Rhudii A. Quaye, Houssam Abbas, Megan S. Ryerson, and Rahul Mangharam. FADS: A framework for autonomous drone safety using temporal logic-based trajectory planning. *Transportation Research Part C: Emerging Technologies*, 130:103275, 2021.

[118] Yash Vardhan Pant, He Yin, Murat Arcak, and Sanjit A. Seshia. Co-design of control and planning for multi-rotor UAVs with signal temporal logic specifications. In *2021 American Control Conference (ACC)*, pages 4209–4216. IEEE, 2021.

[119] Omkar Sudhir Patil, Runhan Sun, Shubhendu Bhasin, and Warren E. Dixon. Adaptive control of time-varying parameter systems with asymptotic tracking. *IEEE Transactions on Automatic Control*, 67(9):4809–4815, 2022.

[120] Doron Peled and Thomas Wilke. Stutter-invariant temporal properties are expressible without the next-time operator. *Information Processing Letters*, 63(5):243–246, 1997.

[121] Jean C Pereira, Valter JS Leite, and Guilherme V Raffo. Nonlinear model predictive control on SE(3) for quadrotor aggressive maneuvers. *Journal of Intelligent & Robotic Systems*, 101(3):62, 2021.

[122] Vasumathi Raman, Alexandre Donzé, Dorsa Sadigh, Richard M. Murray, and Sanjit A. Seshia. Reactive synthesis from signal temporal logic specifications. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, HSCC '15, pages 239–248, New York, NY, USA, 2015. Association for Computing Machinery.

[123] Vasumathi Raman, Alexandre Donzé, Mehdi Maasoumy, Richard M. Murray, Alberto Sangiovanni-Vincentelli, and Sanjit A. Seshia. Model predictive control with signal temporal logic specifications. In *Proc. of CDC*, pages 81–87, 2014.

[124] Nishanth Rao, Suresh Sundaram, and Pushpak Jagtap. Temporal waypoint navigation of multi-UAV payload system using barrier functions. In *Proc. of ECC*, pages 1–6. IEEE, 2023.

[125] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. Stabilization with guaranteed safety using control lyapunov–barrier function. *Automatica*, 66:39–47, 2016.

[126] Sadra Sadraddini and Calin Belta. Robust temporal logic model predictive control. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 772–779. IEEE, 2015.

[127] Mohamed Serry, Yating Yuan, Haocheng Chang, and Jun Liu. Reach-avoid control synthesis for a quadrotor uav with formal safety guarantees. *IEEE Transactions on Automatic Control*, pages 1–15, 2026.

[128] Daqian Shao and Marta Kwiatkowska. Sample efficient model-free reinforcement learning from ltl specifications with optimality guarantees. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4180–4189, 2023.

[129] Maryam Sharifi and Dimos V Dimarogonas. Fixed-time convergent control barrier functions for coupled multi-agent systems under STL tasks. In *Proc. of ECC*, pages 793–798. IEEE, 2021.

[130] Eduardo D. Sontag. A "universal" construction of Artstein's theorem on nonlinear stabilization. *Systems & Control Letters*, 13(2):117–123, 1989.

[131] Koushil Sreenath, Taeyoung Lee, and Vijay Kumar. Geometric control and differential flatness of a quadrotor uav with a cable-suspended load. In *52nd IEEE conference on decision and control*, pages 2269–2274. IEEE, 2013.

[132] Mohit Srinivasan and Samuel Coogan. Control of mobile robots using barrier functions under temporal logic specifications. *IEEE Transactions on Robotics*, 37(2):363–374, 2021.

[133] Mohit Srinivasan, Samuel Coogan, and Magnus Egerstedt. Control of multi-agent systems with finite time control barrier certificates and temporal logic. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 1991–1996. IEEE, 2018.

[134] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[135] Dawei Sun, Jingkai Chen, Sayan Mitra, and Chuchu Fan. Multi-agent motion planning from signal temporal logic specifications. *IEEE Robotics and Automation Letters*, 7(2):3451–3458, 2022.

[136] Maria Svorenova, Jan Kretinsky, Martin Chmelik, Krishnendu Chatterjee, Ivana Cerna, and Calin Belta. Temporal logic control for stochastic linear systems using abstraction refinement of probabilistic games. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 259–268, 2015.

[137] Chang Tian, Yan Yang, Zhongjiao Shi, and Zhijie Liu. Control barrier navigation functions for multi-agent system under signal temporal logic tasks. In *2023 IEEE International Conference on Unmanned Systems (ICUS)*, pages 357–362. IEEE, 2023.

[138] Manuel Toscano-Moreno, Anthony Mandow, María Alcázar Martínez, and Alfonso José García-Cerezo. Spin-based linear temporal logic path planning for ground vehicle missions with motion constraints on digital elevation models. *Sensors*, 24(16):5166, 2024.

[139] Joris Verhagen, Lars Lindemann, and Jana Tumova. Temporally robust multi-agent stl motion planning in continuous time. *arXiv preprint arXiv:2310.10585*, 2023.

[140] Joris Verhagen, Lars Lindemann, and Jana Tumova. Temporally robust multi-agent stl motion planning in continuous time. In *Proc. of ACC*, pages 251–258. IEEE, 2024.

[141] Juan Pablo Vielma. Mixed integer linear programming formulation techniques. *Siam Review*, 57(1):3–57, 2015.

[142] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[143] Eleftherios E. Vlahakis, Lars Lindemann, and Dimos V. Dimarogonas. Distributed sequential receding horizon control of multi-agent systems under recurring signal temporal logic. In *2024 European Control Conference (ECC)*, pages 1284–1289, 2024.

[144] Jing Wang, Xu Chu Ding, Morteza Lahijanian, Ioannis Ch. Paschalidis, and Calin Belta. Temporal logic motion control using actor-critic methods. *The International Journal of Robotics Research*, 34(10):1329–1344, 2015.

[145] Tichakorn Wongpiromsarn, Ufuk Topcu, and Andrew Lamperski. Automata theory meets barrier certificates: Temporal logic verification of nonlinear systems. *IEEE Transactions on Automatic Control*, 61(11):3344–3355, 2015.

[146] Wei Xiao, Calin A. Belta, and Christos G. Cassandras. High order control lyapunov-barrier functions for temporal logic specifications. In *2021 American Control Conference (ACC)*, pages 4886–4891. IEEE, 2021.

[147] Jiahong Xu, Lijie Wang, Yang Liu, and Hong Xue. Finite-time prescribed performance optimal attitude control for quadrotor UAV. *Applied Mathematical Modelling*, 120:752–768, 2023.

[148] Guang Yang, Calin Belta, and Roberto Tron. Continuous-time signal temporal logic planning with control barrier functions. In *Proc. of ACC*, pages 4612–4618. IEEE, 2020.

[149] Tiange Yang, Yuanyuan Zou, Shaoyuan Li, et al. Signal temporal logic synthesis under model predictive control: A low complexity approach. *Control Engineering Practice*, 143:105782, 2024.

[150] Pian Yu, Yulong Gao, Frank J. Jiang, Karl Henrik Johansson, and Dimos V. Dimarogonas. Online control synthesis for uncertain systems under signal temporal logic specifications. *The International Journal of Robotics Research*, 43(6):765–790, 2024.

[151] Pian Yu, Yong Li, David Parker, and Marta Kwiatkowska. Planning with linear temporal logic specifications: Handling quantifiable and unquantifiable uncertainty. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3940–3946, 2025.

[152] Pian Yu, Xiao Tan, and Dimos V Dimarogonas. Continuous-time control synthesis under nested signal temporal logic specifications. *IEEE Transactions on Robotics*, 2024.

[153] Xinyi Yu, Chuwei Wang, Dingran Yuan, Shaoyuan Li, and Xiang Yin. Model predictive control for signal temporal logic specifications with time interval decomposition. In *Proc. of CDC*, 2023.

[154] Yating Yuan and Yu Liu. Continuous-time control synthesis for multiple quadrotors under signal temporal logic specifications. *arXiv preprint arXiv:2505.07240*, 2025.

[155] Yating Yuan, Thanin Quartz, and Jun Liu. Signal temporal logic planning with time-varying robustness. *IEEE Control Systems Letters*, 8:3015–3020, 2024.

[156] Ruikun Zhou, Thanin Quartz, Hans De Sterck, and Jun Liu. Neural Lyapunov control of unknown nonlinear systems with stability guarantees. *Advances in Neural Information Processing Systems*, 35:29113–29125, 2022.

[157] Xiaoyi Zhou, Yuanyuan Zou, Shaoyuan Li, Xianwei Li, and Hao Fang. Distributed model predictive control for multi-robot systems with conflicting signal temporal logic tasks. *IET Control Theory & Applications*, 16(5):554–572, 2022.

[158] Yuanyuan Zou, Xu Su, Shaoyuan Li, Yugang Niu, and Dewei Li. Event-triggered distributed predictive control for asynchronous coordination of multi-agent systems. *Automatica*, 99:92–98, 2019.

Appendix

Appendix A

Appendix for Chapter 4

A.1 Derivatives of the Error Functions

Given the error functions (4.3.2)–(4.3.6), their time derivatives are derived as follows.

- **Derivative of $e_p(t)$.**

$$\dot{e}_p(t) = \dot{y}_{tr}(t) - \dot{y}_d(t) = e_v(t). \tag{A.1.1}$$

- **Derivative of $e_v(t)$.** From (4.2.2) and (4.3.20), one has

$$\begin{aligned} m\dot{e}_v(t) &= m\ddot{y}_{tr}(t) - m\ddot{y}_d(t) \\ &= -(K_p e_p(t) + K_v e_v(t)) + (f(t)R(t)e_3 - F_d(t)) . \end{aligned} \tag{A.1.2}$$

  Assume that $\|F_d(t)\| > 0$ for all $t \geq 0$. Since $f(t) = F_d^\top(t)R(t)e_3$, $b_3(t) = R(t)e_3$, and $b_{3,d}(t) = F_d(t)/\|F_d(t)\|$, the force error $\Delta_f(t) := f(t)R(t)e_3 - F_d(t)$ satisfies

$$\begin{aligned} \Delta_f(t) &= \big(F_d^\top(t)b_3(t)\big)b_3(t) - \|F_d(t)\|b_{3,d}(t) \\ &= \|F_d(t)\| \left(\big(b_{3,d}^\top(t)b_3(t)\big)b_3(t) - b_{3,d}(t)\right) . \end{aligned} \tag{A.1.3}$$

  Thus, $\dot{e}_v(t) = -\frac{1}{m}\big(K_p e_p(t) + K_v e_v(t) - \Delta_f(t)\big)$.

- **Derivative of $\Psi_K(t)$.** Let $G(t) = R_d^\top(t)R(t) \in \mathrm{SO}(3)$. Given $\hat{\omega}_d(t) \in \mathfrak{so}(3)$, the following properties hold:

$$\hat{\omega}_d^\top(t) = -\hat{\omega}_d(t), \tag{A.1.4}$$

$$G(t)G^\top(t) = \mathrm{I}_3, \tag{A.1.5}$$

$$G^\top(t)\hat{\omega}_d(t)G(t) = \big(G^\top(t)\omega_d(t)\big)^\wedge . \tag{A.1.6}$$

Using the rigid body rotational kinematics $\dot{R}(t) = R(t)\hat{\omega}(t)$ and $\dot{R}_d(t) = R_d(t)\hat{\omega}_d(t)$, one can expand the time derivative of $G(t)$. Applying the matrix transpose property $(R_d(t)\hat{\omega}_d(t))^\top = \hat{\omega}_d^\top(t)R_d^\top(t)$, recalling the skew-symmetry $\hat{x}^\top = -\hat{x}$ (2.3.7) and the rotational property $R\hat{x}R^\top = \widehat{Rx}$ (2.3.11) from Lemma 2.3.1, and using $G(t)G^\top(t) = \mathrm{I}_3$, the time derivative $\dot{G}(t)$ is evaluated as:

$$\begin{aligned}
\dot{G}(t) &= \dot{R}_d^\top(t)R(t) + R_d^\top(t)\dot{R}(t) \\
&= \left(R_d(t)\hat{\omega}_d(t)\right)^\top R(t) + R_d^\top(t)\left(R(t)\hat{\omega}(t)\right) \\
&= \hat{\omega}_d^\top(t)R_d^\top(t)R(t) + R_d^\top(t)R(t)\hat{\omega}(t) \\
&= -\hat{\omega}_d(t)G(t) + G(t)\hat{\omega}(t) \\
&= -G(t)G^\top(t)\hat{\omega}_d(t)G(t) + G(t)\hat{\omega}(t) \\
&= -G(t)\left(G^\top(t)\omega_d(t)\right)^\wedge + G(t)\hat{\omega}(t) \\
&= G(t)\left(\hat{\omega}(t) - \left(G^\top(t)\omega_d(t)\right)^\wedge\right) \\
&= G(t)\left(\omega(t) - G^\top(t)\omega_d(t)\right)^\wedge \\
&= G(t)\hat{e}_\omega(t).
\end{aligned} \tag{A.1.7}$$

Thus, recalling the trace identity $\mathrm{tr}(A\hat{x}) = -x^\top(A - A^\top)^\vee$ (2.3.9) from Lemma 2.3.1, it follows that

$$\begin{aligned}
\dot{\Psi}_K(t) &= -\frac{1}{2}\mathrm{tr}\left(K_R G(t)\hat{e}_\omega(t)\right) \\
&= \frac{1}{2}e_\omega^\top(t)\left(K_R G(t) - \left(K_R G(t)\right)^\top\right)^\vee \\
&= e_\omega^\top(t)e_{K,R}(t).
\end{aligned} \tag{A.1.8}$$

- **Derivative of $e_{K,R}(t)$.** Let $A(t) = K_R G(t)$. Since $K_R \in \mathbb{S}^3_{++}$, one has $A^\top(t) = G^\top(t)K_R$. Recalling the matrix identity $A\hat{x} + \hat{x}A^\top = \left(\left(\mathrm{tr}(A)\mathrm{I}_3 - A^\top\right)x\right)^\wedge$ (2.3.10) from Lemma 2.3.1, it follows that

$$\begin{aligned}
\dot{e}_{K,R}(t) &= \frac{1}{2}\left(A(t)\hat{e}_\omega(t) + \hat{e}_\omega(t)A^\top(t)\right)^\vee \\
&= \frac{1}{2}\left(\mathrm{tr}(A(t))\mathrm{I}_3 - A^\top(t)\right)e_\omega(t) \\
&= \frac{1}{2}\left(\mathrm{tr}(K_R G(t))\mathrm{I}_3 - G^\top(t)K_R\right)e_\omega(t) \\
&= \mathfrak{C}(t)e_\omega(t),
\end{aligned} \tag{A.1.9}$$

where $\mathfrak{C}(t) := \frac{1}{2}\left(\mathrm{tr}(K_R G(t))\mathrm{I}_3 - G^\top(t)K_R\right)$.

- **Derivative of $e_\omega(t)$.** Let $G(t) = R_d^\top(t)R(t)$ and $e_\omega(t) = \omega(t) - G^\top(t)\omega_d(t)$. Substituting $\dot{G}(t) = G(t)\hat{e}_\omega(t)$ derived above, recalling the skew-symmetry $\hat{x}^\top = -\hat{x}$ (2.3.7) from Lemma 2.3.1, and using the rotational dynamics along with the control torque $\tau(t)$ (defined in (4.3.22)), one has

$$\begin{aligned}
\dot{e}_\omega(t) &= \dot{\omega}(t) - \dot{G}^\top(t)\omega_d(t) - G^\top(t)\dot{\omega}_d(t) \\
&= \dot{\omega}(t) - \hat{e}_\omega^\top(t)G^\top(t)\omega_d(t) - G^\top(t)\dot{\omega}_d(t) \\
&= \dot{\omega}(t) + \hat{e}_\omega(t)G^\top(t)\omega_d(t) - G^\top(t)\dot{\omega}_d(t) \\
&= J^{-1}\big(-\omega(t)\times J\omega(t) + \tau(t)\big) + \hat{e}_\omega(t)G^\top(t)\omega_d(t) - G^\top(t)\dot{\omega}_d(t) \\
&= J^{-1}\big(-e_{K,R}(t) - K_\omega e_\omega(t)\big) - \hat{\omega}(t)G^\top(t)\omega_d(t) + \hat{e}_\omega(t)G^\top(t)\omega_d(t).
\end{aligned} \tag{A.1.10}$$

Let $q(t) := G^\top(t)\omega_d(t)$. By the linearity of the hat map, $e_\omega(t) = \omega(t) - q(t)$ implies $\hat{e}_\omega(t) = \hat{\omega}(t) - \hat{q}(t)$. Hence, the last two terms in $\dot{e}_\omega(t)$ can be simplified as

$$\begin{aligned}
-\hat{\omega}(t)G^\top(t)\omega_d(t) + \hat{e}_\omega(t)G^\top(t)\omega_d(t) &= -\hat{\omega}(t)q(t) + \hat{e}_\omega(t)q(t) \\
&= -\hat{\omega}(t)q(t) + (\hat{\omega}(t) - \hat{q}(t))\,q(t) \\
&= -\hat{q}(t)q(t) \\
&= 0,
\end{aligned} \tag{A.1.11}$$

where the last equality follows from the cross-product property $\hat{x}x = x\times x = 0$ (2.3.8) in Lemma 2.3.1. Therefore, the time derivative of the angular velocity error reduces to

$$\dot{e}_\omega(t) = J^{-1}\big(-e_{K,R}(t) - K_\omega e_\omega(t)\big) = -J^{-1}\big(e_{K,R}(t) + K_\omega e_\omega(t)\big). \tag{A.1.12}$$