Modeling and High-Performance Algorithms for Mission-Constrained Trajectory Optimization

Samet Uzun

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:

Behçet Açıkmeşe, Chair

Mehran Mesbahi

Maryam Fazel

Program Authorized to Offer Degree:

Aeronautics & Astronautics

University of Washington

Abstract

Modeling and High-Performance Algorithms for Mission-Constrained Trajectory Optimization

Samet Uzun

Chair of the Supervisory Committee:
Professor Behçet Açıkmeşe
William E. Boeing Department of Aeronautics & Astronautics

Autonomous systems increasingly rely on optimization-based planning and control to achieve complex tasks while satisfying dynamics, actuator limitations, safety requirements, and mission objectives. As these systems are asked to operate in more demanding environments, two challenges become central. First, mission requirements must be modeled in a mathematically precise and optimization-compatible way. Second, the resulting nonconvex trajectory generation and guidance problems must be solved reliably and efficiently. This dissertation addresses both challenges through the joint development of modeling frameworks and high-performance algorithms for trajectory optimization with mission-level specifications, together with a decentralized Markov-chain synthesis framework for swarm guidance.

The first part of the dissertation develops a common numerical optimal-control backbone based on free-final-time formulations, time-dilation, finite-dimensional control parameterization, multiple-shooting discretization, and exact-penalty finite-dimensional transcriptions. Building on this foundation, the dissertation develops smooth and exact modeling tools for mission-level specifications using generalized mean-based smooth robustness (GMSR). At the logical level, the resulting constructions provide $\mathcal{C}^1$-smooth and exact parameterizations of temporal logic operators. At the temporal level, the framework is extended from discrete-time STL to continuous-time STL through two complementary realizations: a dense-time realization that is sound and complete up to the accuracy of the underlying numerical inte-

gration scheme, and an augmentation-based realization that embeds temporal aggregation directly into augmented dynamics and is particularly attractive for implementation. These constructions avoid the locality and masking behavior of standard quantitative semantics and yield a more favorable landscape for gradient-based optimization.

To solve the resulting optimization problems, the dissertation develops prox-convex, a structure-preserving sequential convex programming algorithm for composite problems in which convex penalties act on smooth features and smooth outer maps act on convex inner features. Each prox-convex step linearizes only the smooth maps while preserving the available convex structure, and the method is equipped with adaptive proximal regularization, optional curvature injection, and convergence guarantees that include sufficient decrease, complexity bounds, and local linear convergence under a local error-bound condition.

The modeling and algorithmic ingredients are then integrated in a range of trajectory optimization applications, including illustrative continuous-time STL examples, nonlinear model predictive control with continuous-time constraint satisfaction, perception-constrained motion planning for information acquisition, rocket landing with compound state-triggered constraints, and quadrotor flight with continuous-time STL requirements.

The dissertation also addresses large-scale distributed guidance through decentralized state-dependent Markov-chain synthesis (DSMC). A state-dependent consensus protocol is first developed and shown to achieve exponential convergence under mild technical conditions without relying on conventional connectivity assumptions on the dynamic graph topology. Building on this protocol, the dissertation develops a decentralized Markov-chain synthesis method that converges exponentially to a desired steady-state distribution while respecting transition constraints and reducing unnecessary state transitions. Its effectiveness is demonstrated on probabilistic swarm-guidance problems, where it achieves substantially faster convergence than existing Markov-chain-based methods.

Taken together, the results of this dissertation provide a unified perspective on how expressive modeling and structure-preserving algorithms can be combined to enable reliable and efficient guidance for both individual autonomous vehicles and large-scale swarms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

Looking back on my graduate school years, I realize that many of my happiest memories are not attached to any single result or paper, but to the people with whom I had the chance to think, work, argue, learn, and grow. I have been fortunate to meet many people whose company I would choose again in any part of life. This dissertation is the result of their support, mentorship, friendship, and generosity.

First and foremost, I would like to thank my advisor, Professor Behçet Açıkmeşe. After doing and accomplishing so much, you have still kept the curiosity and excitement of someone who is at the beginning of a research journey. That has been deeply inspiring to me. I often left our lab meetings full of excitement and new ideas, while carrying the reminder that enthusiasm matters most when paired with focus. From our first conversation, you encouraged me to find my own direction, and then gave me the freedom to do exactly that. Thank you for giving your students a rare amount of freedom, for creating space for us to follow our curiosity, and for providing the resources, trust, and encouragement needed to do so. Thank you also for supporting us in pursuing internships, collaborations, and experiences in different places, which broadened our perspectives and shaped our research in important ways. Before starting my Ph.D., I could not have imagined being part of such an intellectually open and supportive environment. I know how rare this is, and I am deeply grateful for the trust and freedom you gave me throughout this journey.

I am also grateful to my committee members. Professor Mehran Mesbahi, before coming to UW, I learned a great deal about graph theoretic methods from your book. Later, I had the good fortune of taking your course in person and seeing those ideas come alive in the classroom. Beyond your scholarship, I have always appreciated your kindness, warmth, and the effort you put into bringing people together. Professor Maryam Fazel, I feel very fortunate to have taken convex optimization from one of the leading experts in the field. I already knew from your papers what an outstanding researcher you are, but through your course, I also came to know you as an outstanding teacher. Professor Aleksandr Aravkin, thank you for your optimization course and for the energy you brought to class, even at 8:30 in the morning.

While many of us arrived half-asleep, you made the material exciting enough to wake us up. If I ever teach a course one day, I hope to bring even a fraction of that enthusiasm. Professor Eli Livne, thank you for your structural optimization course, which I enjoyed much more than I expected. Although your field is different from my own, I found that much of the course connected naturally to the questions I cared about. I also remain grateful for the flexibility and understanding you showed throughout the class.

I would like to thank Utku Eren, who became one of the most important figures in shaping my Ph.D. During my internship, you gave me exceptional mentorship and helped me see what kinds of problems could be both practically important and rigorously addressed. In three months, we worked on problems that continued to influence the direction of my research long after the internship ended. Much of my Ph.D. afterward was spent trying to build modeling and solution techniques for problems of exactly this kind. Your mentorship has continued to this day, and I am deeply grateful for your guidance, trust, and support.

Thank you to Taewan Kim, who has been a wonderful friend from the very beginning. I have always admired your patience, honesty, precision, humility, and kindness. Many times, I saw you take responsibility for things simply because they needed to be done, even when they were not officially yours to carry. I feel very lucky that we joined the lab around the same time. You are the kind of friend whose presence gives confidence in both research and life.

Thank you to Abhinav G. Kamath. The day after I first arrived in the United States, you reached out, and we met. I remember immediately meeting someone deeply passionate, excited to build things, and already full of knowledge about the lab and its research. That excitement never faded. Every time we sat down to discuss a problem, my own excitement for research multiplied, and our conversations almost always produced several new directions worth pursuing. Toward the end of my Ph.D., working with you on SCvxGEN became one of my most enjoyable collaborations. I only wish we had had even more time to work together.

Thank you to Skye Mceowen. You are one of the most passionate, thoughtful, and easy-to-work-with people I have met. I met you soon after meeting Abhi, and from the beginning, I saw someone with a deep love for spaceflight and an equally strong care for the people around you. That combination remained unchanged throughout the years. I am very happy that we had the chance to collaborate on the EDL trajectory optimization software, and I am excited to see the work you will continue to do in a field where you so clearly belong.

Thank you to Purnanand Elango. Purna, you are one of the most capable, calm, kind, organized, and careful researchers I have had the chance to know. In addition to the quality of your technical work, I have always admired how clearly you explain ideas and how beautifully you write. The research you led later shaped how many of us in the lab thought about trajectory optimization, and it became a foundation for much of what followed. Working with you has been both a privilege and a pleasure, and I am grateful for all I learned from you through our collaborations.

Thank you to Govind M. Chari. I still remember the early days when you would talk with excitement about what you were thinking and observing on the way to morning classes, while I was still trying to wake up. You are one of the sharpest and clearest thinkers I have ever met, and I have always admired the way you combine intelligence with focus, speed, and a remarkable ability to cut directly to the heart of a problem. At the same time, you remain humble, generous, and genuinely fun to talk to, which makes working with you not only productive but also deeply enjoyable. I will miss our conversations and your presentations in the lab, which always brought clarity, energy, and perspective.

Thank you to Dayou Luo. Your ability to look critically and deeply into the mathematics behind the tools we use taught many of us a great deal. Our collaborations have been among the parts of my Ph.D. that I value the most. I am also grateful for the time you spent giving me feedback on my resume, more than once, with patience and care. I am sure that your help played a role in several opportunities I later received.

Thank you to Selahattin Burak Sarsılmaz. When I think of mathematical precision, you are one of the first people who comes to mind. I have rarely met someone so willing to question even widely accepted ideas, and so capable of finding what is missing or not quite right. Since meeting you during my first year, I have admired your insistence on doing things properly rather than simply following trends, and I am grateful for the friendship you offered during those early days. Your way of thinking will stay with me for a long time.

Thank you to Chris Hayner for your energy, initiative, and the impressive software you have been building. I have always admired that your work does not stop at ideas written on paper or code sitting on a desktop. You build systems, test them, fly them, and push ideas toward reality. That spirit is rare and valuable, and I am glad we had the chance to collaborate near the end of my Ph.D. Thank you to Natalia Pavlasek, my first close lab friend after the pandemic. We later shared office space during our MERL internships, and I

have learned a great deal from your discipline and persistence, especially when dealing with difficult problems. I would also like to thank Aman Tiwary for many thoughtful conversations about ideas that were still forming, some of which may become real projects in the future. I will be following your work with excitement. Thank you to Carlos for joining the lab and for the chance to work together, even if briefly. You are already doing excellent work, and I am excited to see where it goes. Thank you to Jason Zhou. Jason, you are one of the boldest students I have met. Leaving industry after several years to return to graduate school and dive deeper into research could not have been an easy decision, and I have always admired the courage behind it. The drone you built with your own hands in pursuit of a speed record was also one of the coolest drones I have ever seen. Thank you to Pierre-Loïc Garoche. An important part of this dissertation grew from the STL ideas that I first encountered through you, and I am grateful for our collaboration. Thank you to John M. Carson III for our collaboration and for creating space for these research directions through your leadership and support.

I am grateful to Michael Szmuk for the guidance and control course I was fortunate to audit, and for the many papers that helped me understand the foundations of the field when I first joined the lab. The organization, clarity, and care you brought to teaching were remarkable. I learned a great deal from your course and would gladly sit through it several more times. Your work has helped shape ACL into the place it is today, and it has helped not only me but many students who came after you. Thank you to Danylo Malyuta. Even before I came to UW, one of your papers was among the coolest and most instructive papers I had read. When I later had the chance to meet you, I found someone who was not only technically brilliant but also kind, thoughtful, and humble. Having your generation just ahead of us has been a great fortune for many of us, and I am grateful for the example you both set.

I would like to thank Nazım Kemal Üre. During my undergraduate years, I first knew of you as someone doing a Ph.D. at MIT and also as an excellent bass guitarist, which was already enough to make you an intriguing figure to me. Then I took your engineering mathematics course, and I think that course changed the direction of my life. I later attended the same course two or three more times, and you became my advisor during my master's studies. Working with you was always fun, energetic, and inspiring. You also introduced me to Behçet Açıkmeşe, who created one of the most important butterfly effects in my academic life. I am deeply grateful for your mentorship and friendship.

Thank you to Tolga Ok. If I trace the butterfly effects in my life, one of the earliest begins with you. Had we never met, I would probably be working on very different problems in a very different place today. From the beginning of my undergraduate years through the end of my master's, our almost uninterrupted friendship and our countless technical, philosophical, and everyday conversations shaped not only the work I do now, but also how I think about life. You are one of the calmest, kindest, easiest people to be around, and also one of the clearest and most critical thinkers I know. I feel very lucky that we met by chance and remained friends for so long.

If I tried to thank everyone individually, the acknowledgments might become half of the dissertation. I would like to thank all members of the Autonomous Controls Laboratory, past and present, for the conversations, feedback, collaboration, and friendship that made my time at UW so meaningful. In particular, I would like to thank Samuel Buckner, Fabio Spada, Avi Mittal, Daniel Calderone, Yue Yu, Ben Chung, Kazuya Echigo, Sarah Li, Justin Ganiban, Oliver Sheridan, and many others whose presence made the lab a better place. I also want to thank all my friends from ITU AI and ITU ARC. The combination of these communities has played a major role in making me who I am today.

Finally, and most importantly, I would like to thank my family. Almost all of the meaning behind this journey rests on you, and without you, none of this would feel complete. To my parents, Nurten & Adnan Uzun, thank you for giving me a life full of love and trust, for giving me both freedom and responsibility, and for supporting every decision I made. I have written many times in these acknowledgments that I was lucky, but my greatest luck began before I could even know it, by being born into your love.

To my sister and brother, Bilge & Rıdvan Uzun, I cannot imagine having had a better sister and brother. Since childhood, I have always known that I could lean on you with complete trust. I wish I could give you the feeling of what it means to have siblings like you. To Zeynep Sare Uzun, the most beloved and beautiful part of our family, until you arrived, I had always been the youngest. Through you, I learned for the first time the joy of loving someone younger in that way.

To my love, Aymina Özcelep, meeting you has been one of the greatest pieces of luck in my life. With you, life became warmer, calmer, and more beautiful than I could have imagined. You brought me a kind of happiness that made life feel like spring. Thank you for being in my life, and for letting me be in yours.

DEDICATION

To my loving family.

Chapter 1

INTRODUCTION

1.1 Background and Contributions

Autonomous systems increasingly rely on optimization-based planning and control to achieve complex tasks while satisfying dynamics, actuator limitations, safety requirements, and mission objectives. In modern guidance, navigation, and control (GNC) architectures, trajectory optimization occupies a central role by converting high-level goals into dynamically feasible reference trajectories that can be executed by lower-level feedback controllers [29, 30, 183]. As the capabilities of autonomous systems grow, so does the complexity of the constraints they must satisfy. This creates a dual challenge. Mission requirements must first be modeled in a mathematically precise yet optimization-compatible way, and the resulting nonconvex trajectory-generation problems must then be solved reliably and efficiently.

A particularly important source of modeling complexity comes from *mission-level specifications*. Temporal logic provides a systematic language for expressing sequencing, timing, persistence, implication, and conditional behaviors in dynamical systems [19, 178]. Signal Temporal Logic (STL), in particular, is well suited to physical systems because its predicates are defined over real-valued signals and its quantitative semantics allow temporal logic requirements to be incorporated into optimization [89]. However, standard robustness constructions based on nested min / max operators are not well matched to high-performance numerical solvers. They are either nonsmooth or rely on approximations that sacrifice exactness, and they often suffer from locality and masking effects that degrade gradient-based optimization. These issues become even more pronounced in continuous time, where node-wise enforcement may fail to guarantee inter-sample safety for `always` specifications and can introduce conservatism for `eventually` specifications [284–286].

At the same time, solving the resulting trajectory-generation problems requires numerical algorithms that exploit structure while preserving the properties that make the problem computationally tractable. In direct optimal-control methods, continuous-time problems are transcribed into finite-dimensional optimization problems through time-dilation, control parameterization, and discretization [29, 30, 36].

Once transcribed, these problems can be solved using nonlinear programming methods such as interior-point methods (IPMs), sequential quadratic programming (SQP), or sequential convex programming (SCP) frameworks [104, 147, 184]. For real-time and embedded applications, solver design is therefore inseparable from modeling design, since the way a problem is formulated strongly influences whether a fast, reliable algorithm can solve it in practice. This dissertation adopts this perspective and develops modeling tools and optimization algorithms in tandem.

A second, complementary theme of the dissertation concerns *distributed guidance* for large populations of agents. When the goal is to guide the spatial distribution of a swarm rather than a single trajectory, probabilistic swarm guidance provides a natural density-based modeling framework, while Markov-chain synthesis provides a mechanism for steering that distribution toward a desired steady state [3]. In that setting, scalability, decentralization, and convergence guarantees become central concerns. Existing probabilistic swarm-guidance methods based on homogeneous or time-inhomogeneous Markov chains can be effective, but they may rely on global information, degrade in sparsely connected state spaces, or provide only asymptotic convergence guarantees [4, 21, 141]. This motivates the development of new state-dependent synthesis rules with exponential convergence guarantees and better practical performance.

The main contributions of this dissertation are as follows.

1. It develops a smooth and exact modeling framework for mission-level specifications based on generalized mean-based smooth robustness (GMSR). At the logical level, the resulting constructions provide $\mathcal{C}^1$-smooth and exact parameterizations of STL operators. At the temporal level, the framework is extended from discrete-time STL to continuous-time

STL through both dense-time and augmentation-based realizations, thereby enabling smooth trajectory optimization with rich mission specifications [284–286].

2. It develops a structure-preserving sequential convex programming algorithm, prox-convex, for a broad class of nonconvex optimization problems of the form

$$F(x) = g(x) + h(C(x)) + s(R(x)),$$

where convex penalties act on smooth features and smooth outer maps act on convex inner features. The method linearizes only the smooth maps, preserves the convex structure, supports adaptive proximal regularization and optional curvature injection, and is equipped with convergence, complexity, and local linear-rate guarantees [291].

3. It integrates these modeling and algorithmic ingredients into trajectory optimization with mission-level specifications. The resulting framework combines free-final-time optimal control, time-dilation, control parameterization, multiple-shooting discretization, exact-penalty formulations, smooth mission-specification modeling, and SCP-based solution methods [104, 284–286]. Its practical utility is demonstrated on illustrative CT-STL examples, nonlinear model predictive control with continuous-time constraint satisfaction, perception-constrained motion planning for information acquisition, rocket landing with compound state-triggered constraints, and quadrotor flight with combined CT-STL requirements [284, 287, 289, 290].

4. It develops a decentralized state-dependent Markov-chain synthesis (DSMC) framework for swarm guidance. The chapter first introduces a state-dependent consensus protocol with exponential convergence under mild technical conditions and without conventional connectivity assumptions on the dynamic topology. Building on that protocol, it constructs a state-dependent Markov-chain synthesis algorithm that converges exponentially to a desired steady-state distribution while respecting transition constraints and reducing unnecessary state transitions. The resulting swarm-guidance strategy is shown numerically to converge substantially faster than previous Markov-chain-based approaches [288].

1.2 Dissertation Outline

Chapter 2 introduces the continuous-time optimal-control and numerical ingredients used throughout the dissertation. It develops the common transcription backbone based on free-final-time optimal control, time-dilation, finite-dimensional control parameterization, multiple-shooting discretization, and exact-penalty finite-dimensional formulations.

Chapter 3 develops smooth and exact modeling tools for mission-level specifications. It reviews STL and standard robustness semantics, introduces generalized mean-based smooth robustness for logical operators, extends the construction to dense-time temporal operators, develops an augmentation-based realization of continuous-time STL, and studies the resulting smoothness, exactness, monotonicity, and gradient-behavior properties.

Chapter 4 develops prox-convex, a structure-preserving sequential convex programming algorithm for nonconvex composite optimization. The chapter introduces the problem class, derives convex local models and adaptive proximal metrics, establishes the main global and local convergence guarantees, and demonstrates the numerical advantages of preserving convex inner structure.

Chapter 5 brings together the modeling and numerical machinery of the preceding chapters in trajectory optimization applications with mission-level specifications. It begins with illustrative continuous-time STL examples for the `always`, `eventually`, and `until` operators, and then treats richer applications including receding-horizon obstacle avoidance, information-acquisition motion planning, rocket landing with compound state-triggered constraints, and quadrotor flight with continuous-time STL requirements.

Chapter 6 turns from single-trajectory generation to distributed swarm guidance. It develops a consensus protocol with state-dependent weights, builds the decentralized state-dependent Markov-chain synthesis (DSMC) algorithm, establishes exponential convergence guarantees, and demonstrates the resulting framework on probabilistic swarm-guidance examples.

The dissertation concludes with a final chapter summarizing the main findings and discussing directions for future research.

Chapter 2

OPTIMAL CONTROL AND NUMERICAL PRELIMINARIES

2.1 Introduction

Optimization plays a central role in modern guidance, navigation, and control, where it is used to generate reference trajectories and feedforward control inputs that satisfy system dynamics, boundary conditions, and performance objectives while accommodating increasingly complex operational constraints [29, 30]. In trajectory generation problems for autonomous systems, these constraints may arise from actuator limits, state bounds, safety requirements, sensing requirements, or mission-level task objectives. As a result, continuous-time optimal control provides a natural modeling language for expressing such problems, while numerical optimization provides the computational machinery needed to solve them reliably in practice [170, 183].

For general nonlinear systems, practical trajectory optimization is dominated by direct methods, in which the continuous-time optimal control problem is transcribed into a finite-dimensional optimization problem through control parameterization and time discretization [36]. Many widely used software tools follow this paradigm. Collocation- and pseudospectral-based packages such as GPOPS-II and Dymos provide high-level modeling interfaces for aerospace and multidisciplinary applications, while shooting-based environments such as ACADO and acados emphasize high-performance implementations for embedded and real-time control [111, 136, 226, 294]. In many of these workflows, the resulting sparse nonlinear program is solved by generic nonlinear optimization packages such as IPOPT or SNOPT, or by structure-exploiting quadratic programming solvers such as HPIPM inside SQP-based real-time implementations [120, 122, 294, 295].

More recently, sequential convex programming frameworks have emerged as powerful alternatives for nonconvex trajectory generation, especially in applications where one wishes to preserve convex structure while handling complex nonlinear dynamics and constraints [104, 183, 184, 291]. Representative applications include quadrotor path planning and obstacle avoidance [270, 272, 284, 290], aircraft approach and landing [155], spacecraft rendezvous, station keeping, and cislunar relative motion [64, 106, 182, 227, 264], hypersonic reentry [193, 194], and rocket landing [49, 145, 238, 274, 289]. These ideas have also been extended to trajectory optimization with temporal logic specifications and to embedded code generation for real-time implementations [147, 284–286, 289].

The purpose of this chapter is not to survey these algorithms in full detail, but rather to establish the common numerical optimal-control backbone used throughout the dissertation. To that end, we introduce the continuous-time free-final-time optimal-control formulation, its conversion to fixed-horizon form through time-dilation, finite-dimensional control parameterization, multiple-shooting discretization, and the exact-penalty finite-dimensional formulations that will underlie the later modeling and algorithmic developments. These ingredients form the shared numerical substrate on top of which the mission-level specification models of Chapter 3, the structure-preserving optimization framework of Chapter 4, and the application chapters are built.

2.2 Free-final-time optimal control problems

Consider a nonlinear continuous-time dynamical system

$$\dot{x}(t) = F\Big(t, x(t), u(t)\Big), \qquad t \in [t_i, t_f], \tag{2.1}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state, $u(t) \in \mathbb{R}^{n_u}$ is the control input, $t_i \in \mathbb{R}$ is the initial time, and $t_f > t_i$ is a free terminal time. The dynamics map F is assumed to be continuously differentiable in its arguments.

A generic free-final-time optimal control problem may be written in Bolza form as

$$
\begin{aligned}
\underset{x(\cdot),\,u(\cdot),\,t_f}{\text{minimize}} \quad & m_f\Big(t_f, x(t_f)\Big) + \int_{t_i}^{t_f} \ell\Big(t, x(t), u(t)\Big)\, dt && \text{(2.2a)}\\
\text{subject to} \quad & \dot{x}(t) = F\Big(t, x(t), u(t)\Big), \qquad t \in [t_i, t_f], && \text{(2.2b)}\\
& u(t) \in \mathcal{U}, \qquad t \in [t_i, t_f], && \text{(2.2c)}\\
& P\Big(t_i, x(t_i), t_f, x(t_f)\Big) \le 0, && \text{(2.2d)}\\
& Q\Big(t_i, x(t_i), t_f, x(t_f)\Big) = 0. && \text{(2.2e)}
\end{aligned}
$$

Here, ℓ denotes the running cost, m_f denotes the terminal cost, $\mathcal{U} \subset \mathbb{R}^{n_u}$ is a convex admissible input set, and P and Q collect the terminal and boundary inequality and equality constraints, respectively.

Remark 2.1 (Path constraints). Continuous-time path constraints are not included explicitly in (2.2), because the machinery used to model their continuous-time satisfaction is introduced in the next chapter, together with more general mission-level temporal logic specifications. The corresponding modeling technique was first developed for continuous-time path constraints in [104] and later extended to continuous-time STL specifications in [284].

For numerical optimization, it is often convenient to convert (2.2) into Mayer form by augmenting the dynamics with the scalar state

$$
\dot{\eta}(t) = \ell\Big(t, x(t), u(t)\Big), \qquad \eta(t_i) = 0, \tag{2.3}
$$

so that the objective becomes

$$
m_f\Big(t_f, x(t_f)\Big) + \eta(t_f). \tag{2.4}
$$

Accordingly, the state vector x may be interpreted as already including any auxiliary states needed for cost accumulation or later modeling constructions.

2.3 Time-dilation

To handle the free terminal time, the problem is transformed to a fixed normalized horizon using time-dilation. Let $\tau \in [0,1]$ denote the dilated-time variable, and let

$$\tilde{t} : [0,1] \to [t_i, t_f] \tag{2.5}$$

be a strictly increasing continuously differentiable mapping satisfying

$$\tilde{t}(0) = t_i, \qquad \tilde{t}(1) = t_f. \tag{2.6}$$

Its derivative

$$s(\tau) \coloneqq \frac{d\tilde{t}(\tau)}{d\tau} \tag{2.7}$$

is called the dilation factor. Since $\tilde{t}$ must be strictly increasing, we require

$$s(\tau) > 0, \qquad \tau \in [0,1]. \tag{2.8}$$

The terminal time is then recovered from

$$t_f = t_i + \int_0^1 s(\tau)\, d\tau. \tag{2.9}$$

To rewrite the dynamics on the fixed interval $[0,1]$, physical time is appended as an additional state and the time-dilated state and input are defined as

$$\tilde{x}(\tau) \coloneqq \begin{bmatrix} x\big(\tilde{t}(\tau)\big) \\ \tilde{t}(\tau) \end{bmatrix}, \qquad \tilde{u}(\tau) \coloneqq \begin{bmatrix} u\big(\tilde{t}(\tau)\big) \\ s(\tau) \end{bmatrix}. \tag{2.10}$$

Applying the chain rule to (2.1) yields the time-dilated dynamics

$$\frac{d\tilde{x}(\tau)}{d\tau} = \tilde{F}\big(\tilde{x}(\tau), \tilde{u}(\tau)\big) \coloneqq \begin{bmatrix} s(\tau)\, F\Big(\tilde{t}(\tau), x\big(\tilde{t}(\tau)\big), u\big(\tilde{t}(\tau)\big)\Big) \\ s(\tau) \end{bmatrix}. \tag{2.11}$$

Hence, the free-final-time optimal control problem on $[t_i, t_f]$ is equivalently transformed into a fixed-final-time problem on the normalized interval $[0,1]$. The free terminal time is now represented through the additional input $s(\tau)$ rather than through a moving horizon.

A key benefit of time-dilation is that it enables local temporal allocation. Different parts of a trajectory may be stretched or compressed by varying the dilation factor over the normalized horizon, while the computational grid in dilated time remains fixed.

2.4 Control parameterization

The infinite-dimensional optimal control problem is transcribed to a finite-dimensional one by parameterizing the input functions with finitely many variables. Although other parameterizations can also be employed, this dissertation primarily uses first-order hold (FOH) for the physical control and zero-order hold (ZOH) for the dilation factor.

Let the normalized interval be partitioned into K nodes,

$$0 = \tau_1 < \tau_2 < \cdots < \tau_K = 1, \qquad \Delta\tau \coloneqq \tau_{k+1} - \tau_k, \qquad k \in \mathcal{K} \coloneqq \{1, \ldots, K-1\}, \tag{2.12}$$

where, for simplicity, a uniform grid is assumed. The nodal state, control, and dilation-factor values will form the primary decision variables of the discretized problem.

2.4.1 First-order hold for the control input

The physical control input, viewed on the normalized horizon through the map $\tau \mapsto \tilde{t}(\tau)$, is parameterized by FOH:

$$u\Big(\tilde{t}(\tau)\Big) = u_k \frac{\tau_{k+1} - \tau}{\Delta\tau} + u_{k+1} \frac{\tau - \tau_k}{\Delta\tau}, \qquad \tau \in [\tau_k, \tau_{k+1}], \tag{2.13}$$

By slight abuse of notation, we will often refer to this normalized-horizon representation simply as $u(\tau)$.

This choice has several advantages. First, it yields a continuous control signal. Second, if the admissible control set $\mathcal{U}$ is convex, then constraining the nodal inputs,

$$u_k \in \mathcal{U}, \qquad k = 1, \ldots, K, \tag{2.14}$$

is sufficient to guarantee $u(\tau) \in \mathcal{U}$ for all $\tau \in [0, 1]$, since (2.13) is a convex combination of endpoint values.

2.4.2 Zero-order hold for the dilation factor

The dilation factor is parameterized by ZOH:

$$s(\tau) = s_k, \qquad \tau \in [\tau_k, \tau_{k+1}), \qquad k \in \mathcal{K}. \tag{2.15}$$

Thus, the physical duration of interval k is

$$\Delta t_k = \int_{\tau_k}^{\tau_{k+1}} s(\tau)\, d\tau = s_k \Delta\tau. \tag{2.16}$$

To ensure strictly forward physical time, we impose

$$s_k \geq s_{\min} > 0, \qquad k \in \mathcal{K}. \tag{2.17}$$

Why ZOH is used for $s(\tau)$. The use of ZOH for the dilation factor is deliberate. If $s(\tau)$ were parameterized by FOH instead, then the physical interval lengths would satisfy

$$\Delta t_k = \frac{1}{2}(s_k + s_{k+1})\Delta\tau, \tag{2.18}$$

which couples neighboring interval lengths through shared nodal variables. Thus, under FOH, a local change in one interval duration generally propagates to adjacent intervals through the common nodal values. In particular, the interval lengths are no longer independently represented by the nodal dilation variables, and not every positive interval-length sequence can be realized by nonnegative nodal dilation values.

This restriction can already be seen in a simple example. Define the normalized interval lengths

$$\Delta_k := \frac{\Delta t_k}{\Delta\tau}, \qquad k \in \mathcal{K}.$$

Suppose one wishes to realize the positive sequence

$$\Delta = [1,\ 3,\ 1].$$

Under FOH, (2.18) implies

$$\Delta_k = \frac{1}{2}(s_k + s_{k+1}),$$

so the nodal dilation values would have to satisfy

$$s_2 = 2 - s_1, \qquad s_3 = 6 - s_2 = 4 + s_1, \qquad s_4 = 2 - s_3 = -2 - s_1.$$

Hence $s_4 < 0$ for every $s_1 \geq 0$, so no nonnegative FOH dilation sequence can realize the positive interval-length sequence [1, 3, 1].

More generally, the following proposition characterizes exactly which interval-length sequences are realizable under FOH.

Proposition 2.2 (FOH imposes a realizability restriction on interval lengths)**.** *If the dilation factor were parameterized by FOH, then the physical interval lengths would satisfy*

$$\Delta t_k = \frac{1}{2}(s_k + s_{k+1})\Delta\tau.$$

As a result, the normalized interval-length sequence $\Delta_k = \Delta t_k/\Delta\tau$ would have to satisfy a coupled linear system $As = \Delta$ with $s \geq 0$. Such a system is feasible if and only if every alternating sum over every contiguous odd-length block of Δ is nonnegative, namely

$$\sum_{r=i}^{j}(-1)^{r-i}\Delta_r \geq 0, \qquad \forall\, 1 \leq i \leq j \leq K-1, \;\; j-i \text{ even.}$$

Hence, FOH imposes a structural restriction on the realizable physical interval lengths, so not every positive interval-length sequence can be represented by nonnegative nodal dilation values.

Proof. Suppose the dilation factor is parameterized by FOH. Then

$$\Delta t_k = \frac{1}{2}(s_k + s_{k+1})\Delta\tau, \qquad k = 1, \ldots, K-1.$$

With $\Delta_k := \Delta t_k/\Delta\tau$, realizability of a prescribed interval-length sequence is equivalent to

$$As = \Delta, \qquad s \geq 0, \tag{2.19}$$

where

$$A = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & \cdots & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \in \mathbb{R}^{(K-1)\times K}.$$

By Farkas' lemma, (2.19) is feasible if and only if

$$\Delta^\top y \leq 0 \qquad \forall\, y \in \mathbb{R}^{K-1} \text{ s.t. } A^\top y \leq 0. \tag{2.20}$$

The inequalities $A^\top y \le 0$ are $y_1 \le 0$, $y_{i-1} + y_i \le 0 \ \ (2 \le i \le K-1)$, $y_{K-1} \le 0$. The cone defined by these inequalities is polyhedral, and its extreme rays are supported on contiguous odd-length blocks with alternating signs:

$$y_k^{(i,j)} = \begin{cases} (-1)^{k-i+1}, & i \le k \le j, \\ 0, & \text{otherwise}, \end{cases}$$

$1 \le i \le j \le K-1, \ \ j-i$ even. Therefore, it suffices to check (2.20) on these extreme rays, which yields

$$\Delta^\top y^{(i,j)} \le 0 \quad \Longleftrightarrow \quad \sum_{r=i}^{j} (-1)^{r-i} \Delta_r \ge 0,$$

$\forall\, 1 \le i \le j \le K-1, \ \ j-i$ even. This proves Proposition 2.2. ■

Under ZOH, by contrast, the interval lengths are given by

$$\Delta t_k = s_k \Delta\tau,$$

so each interval duration depends only on its own dilation variable. Hence any positive interval-length sequence can be represented directly, without coupling adjacent intervals through shared nodal values. This decoupling is also advantageous numerically, since local time allocations can be adjusted independently, making the temporal mesh more flexible and easier to optimize than under FOH.

For compactness in the next section, define the hybrid nodal input

$$\tilde{u}_k := \begin{bmatrix} u_k \\ s_k \end{bmatrix}, \qquad k \in \mathcal{K}, \tag{2.21}$$

and let

$$\tilde{u}_K := \begin{bmatrix} u_K \\ s_K \end{bmatrix}, \tag{2.22}$$

where the second component of $\tilde{u}_K$ is irrelevant, since the dilation factor is parameterized over intervals and is therefore used only for $k \in \mathcal{K}$.

2.5 Multiple-shooting discretization

After time-dilation and input parameterization, the continuous-time dynamics are transcribed to a finite-dimensional optimization problem using multiple shooting [36, 232]. Let

$$\tilde{x}_k := \tilde{x}(\tau_k), \qquad k = 1, \ldots, K, \tag{2.23}$$

denote the nodal values of the time-dilated state. On each interval $[\tau_k, \tau_{k+1}]$, the state is propagated from $\tilde{x}_k$ under the parameterized inputs introduced in Section 2.4. The corresponding flow map is written as

$$f_{\tau_k}^{\tau_{k+1}}(\tilde{x}_k, \tilde{u}_k, \tilde{u}_{k+1}) := \tilde{x}_k + \int_{\tau_k}^{\tau_{k+1}} \tilde{F}\Big(\tilde{x}(\tau), \tilde{u}(\tau)\Big)\, d\tau, \tag{2.24}$$

where $\tilde{x}(\tau)$ is the continuous-time solution of (2.11) satisfying $\tilde{x}(\tau_k) = \tilde{x}_k$ on that interval. Here, the notation is understood in the hybrid sense introduced above. On interval $[\tau_k, \tau_{k+1}]$, the control component is parameterized by FOH using u_k and u_{k+1}, while the dilation component is parameterized by ZOH using s_k. Whenever $f_{\tau_{K-1}}^{\tau_K}$ is written in terms of $\tilde{u}_K$, the latter is understood as a notational placeholder formed from the terminal control u_K and an unused terminal dilation entry.

Under the standing smoothness assumptions on the dynamics and the chosen control parameterization, the flow map is continuously differentiable with respect to its arguments. The multiple-shooting constraints are then imposed as

$$\tilde{x}_{k+1} = f_{\tau_k}^{\tau_{k+1}}(\tilde{x}_k, \tilde{u}_k, \tilde{u}_{k+1}), \qquad k \in \mathcal{K}, \tag{2.25}$$

or, equivalently, via the defect equations

$$d_k(Z) := \tilde{x}_{k+1} - f_{\tau_k}^{\tau_{k+1}}(\tilde{x}_k, \tilde{u}_k, \tilde{u}_{k+1}) = 0, \qquad k \in \mathcal{K}, \tag{2.26}$$

where Z denotes the stacked decision vector. Stacking these interval defects gives the global defect map

$$D(Z) := \Big(d_1(Z), \ldots, d_{K-1}(Z)\Big). \tag{2.27}$$

A typical decision vector, therefore, takes the form

$$Z := \Big(\tilde{x}_1, \dots, \tilde{x}_K,\, \tilde{u}_1, \dots, \tilde{u}_K\Big),$$

possibly augmented with additional variables depending on the modeling layer introduced later. Recall that the second component of $\tilde{u}_K$ is irrelevant, since the dilation factor is parameterized over intervals and is therefore used only for $k \in \mathcal{K}$.

For gradient-based optimization, one also needs the Jacobians of the flow map and of the defect equations. These can be computed by integrating the associated variational equations alongside the state trajectory on each interval. Define the FOH shape functions

$$\lambda_k^-(\tau) := \frac{\tau_{k+1} - \tau}{\Delta\tau}, \qquad \lambda_k^+(\tau) := \frac{\tau - \tau_k}{\Delta\tau}, \qquad \tau \in [\tau_k, \tau_{k+1}],$$

so that the physical control on interval k is

$$u\Big(\tilde{t}(\tau)\Big) = \lambda_k^-(\tau) u_k + \lambda_k^+(\tau) u_{k+1}.$$

Along the interval trajectory, define

$$\tilde{A}_k(\tau) := \nabla_{\tilde{x}} \tilde{F}\Big(\tilde{x}(\tau), \tilde{u}(\tau)\Big), \qquad \tilde{B}_k(\tau) := \nabla_u \tilde{F}\Big(\tilde{x}(\tau), \tilde{u}(\tau)\Big), \qquad \tilde{S}_k(\tau) := \partial_s \tilde{F}\Big(\tilde{x}(\tau), \tilde{u}(\tau)\Big).$$

The sensitivities of the propagated state with respect to the active nodal variables on interval k are then obtained from the linear time-varying initial-value problems

$$\begin{aligned}
\frac{d}{d\tau}\Phi_k^x(\tau) &= \tilde{A}_k(\tau)\Phi_k^x(\tau), & \Phi_k^x(\tau_k) &= I,\\
\frac{d}{d\tau}\Phi_k^-(\tau) &= \tilde{A}_k(\tau)\Phi_k^-(\tau) + \tilde{B}_k(\tau)\lambda_k^-(\tau), & \Phi_k^-(\tau_k) &= 0,\\
\frac{d}{d\tau}\Phi_k^+(\tau) &= \tilde{A}_k(\tau)\Phi_k^+(\tau) + \tilde{B}_k(\tau)\lambda_k^+(\tau), & \Phi_k^+(\tau_k) &= 0,\\
\frac{d}{d\tau}\Phi_k^s(\tau) &= \tilde{A}_k(\tau)\Phi_k^s(\tau) + \tilde{S}_k(\tau), & \Phi_k^s(\tau_k) &= 0,
\end{aligned} \tag{2.28}$$

for $\tau \in [\tau_k, \tau_{k+1}]$. Their terminal values yield the Jacobians of the flow map:

$$\begin{aligned}
A_k &:= \Phi_k^x(\tau_{k+1}) = \partial_{\tilde{x}_k} f_{\tau_k}^{\tau_{k+1}}(\tilde{x}_k, \tilde{u}_k, \tilde{u}_{k+1}),\\
B_k^- &:= \Phi_k^-(\tau_{k+1}) = \partial_{u_k} f_{\tau_k}^{\tau_{k+1}}(\tilde{x}_k, \tilde{u}_k, \tilde{u}_{k+1}),\\
B_k^+ &:= \Phi_k^+(\tau_{k+1}) = \partial_{u_{k+1}} f_{\tau_k}^{\tau_{k+1}}(\tilde{x}_k, \tilde{u}_k, \tilde{u}_{k+1}),\\
S_k &:= \Phi_k^s(\tau_{k+1}) = \partial_{s_k} f_{\tau_k}^{\tau_{k+1}}(\tilde{x}_k, \tilde{u}_k, \tilde{u}_{k+1}).
\end{aligned} \tag{2.29}$$

Consequently, the Jacobians of the defect constraints are obtained immediately as

$$\partial_{\tilde{x}_{k+1}} d_k(Z) = I, \qquad \partial_{\tilde{x}_k} d_k(Z) = -A_k,$$
$$\partial_{u_k} d_k(Z) = -B_k^-, \qquad \partial_{u_{k+1}} d_k(Z) = -B_k^+, \qquad \partial_{s_k} d_k(Z) = -S_k.$$

If one prefers to work directly with the hybrid inputs $\tilde{u}_k$ and $\tilde{u}_{k+1}$, the corresponding Jacobians are obtained by assembling these blocks together with the obvious zero blocks for inactive components. The same variational equations may also be integrated only up to an intermediate time $\tau \in [\tau_k, \tau_{k+1}]$, which yields the Jacobians of intermediate integration subnodes. Thus, every subnode state on interval k is a differentiable function only of the local variables $\tilde{x}_k$, $\tilde{u}_k$, and $\tilde{u}_{k+1}$. This observation will be important in Chapter 3, where dense-time STL robustness values are evaluated at integration subnodes and differentiated with respect to the original nodal decision variables.

The main appeal of multiple shooting is that the nonlinear dynamics are propagated directly on each interval, while inter-interval consistency is enforced only through the defect constraints. This makes the resulting transcription well-suited to nonlinear trajectory optimization and to linearization-based numerical methods.

2.6 Exact-penalty convex-constrained formulation

After time-dilation, parameterization, and multiple-shooting discretization, the continuous-time optimal control problem becomes a finite-dimensional nonlinear optimization problem. A useful representation is obtained by separating the constraints into

1. convex constraints retained explicitly in the feasible set, and
2. nonlinear equality and inequality constraints moved into the objective through exact penalties.

This yields a convex-constrained nonconvex optimization problem.

Let Ω denote the explicitly retained convex feasible set. This set may include, for example, convex bounds on states and inputs, convex cone constraints, and positivity constraints on the dilation variables. Let the remaining nonlinear constraints be written abstractly as

$$D_{\text{eq}}(Z) = 0, \qquad D_{\text{ineq}}(Z) \leq 0, \tag{2.30}$$

where D_{eq} typically includes the multiple-shooting defect map and any other nonconvex equalities, while D_{ineq} collects the remaining nonlinear inequalities. Then the discretized problem can be written as

$$\begin{aligned}
\underset{Z}{\text{minimize}} \quad & J_d(Z) && \text{(2.31a)}\\
\text{subject to} \quad & Z \in \Omega, && \text{(2.31b)}\\
& D_{\text{eq}}(Z) = 0, && \text{(2.31c)}\\
& D_{\text{ineq}}(Z) \leq 0. && \text{(2.31d)}
\end{aligned}$$

An ℓ_1-type exact-penalty formulation of (2.31) is

$$\underset{Z \in \Omega}{\text{minimize}} \quad J_d(Z) + w_{\text{eq}}^\top \left| D_{\text{eq}}(Z) \right| + w_{\text{ineq}}^\top \left[D_{\text{ineq}}(Z) \right]_+, \tag{2.32}$$

where w_{eq} and w_{ineq} are elementwise positive penalty-weight vectors, $|\cdot|$ is applied elementwise, and $[\cdot]_+$ denotes the elementwise positive-part operator.

For sufficiently large finite penalty weights, (2.32) is exact in the standard sense. Under standard regularity assumptions, feasible local minimizers or stationary points of the penalized problem correspond to local minimizers or KKT points of the original constrained problem; see the classical exact-penalty literature [80, 129, 215]. Thus, selected nonlinear constraints may be shifted from the feasible set into the objective without losing the desired optimality structure. This representation is particularly useful because it produces a problem with a convex feasible set and a nonconvex objective, which is precisely the structural form exploited later in the dissertation.

Chapter 3

SMOOTH AND EXACT MODELING OF MISSION-LEVEL SPECIFICATIONS

3.1 Introduction

Temporal logic provides a systematic language for specifying complex task requirements in dynamical systems [19]. Its expressive power makes it especially attractive for mission-level specification, since it can encode sequencing, timing, persistence, implication, and conditional behaviors in a mathematically precise manner. In this sense, temporal logic provides a natural bridge between high-level mission requirements and optimization-based trajectory generation. As a result, it has become an important tool in planning and control, with applications spanning autonomous driving [250], robotic manipulation [162], multi-agent coordination [85], mobile robotics [55, 58, 249], and UAV mission planning [23, 224]. Temporal logic constraints have also been incorporated into learning frameworks to enforce structural and behavioral properties in classification, prediction, preference learning, and policy synthesis [11, 12, 39, 148, 176]. These developments have created a strong demand for formulations that integrate temporal logic specifications directly into modern optimization and learning pipelines [164].

Among temporal logics, linear temporal logic (LTL) has played a central role in formal synthesis for discrete-event systems [228]. Classical approaches rely on automata-theoretic constructions [13], together with receding-horizon variants and supporting software tools [299, 300]. Mixed-integer formulations have also been developed to encode temporal specifications directly into optimization problems [149]. For continuous and hybrid systems, however, specifications must capture not only logical structure but also quantitative timing and predicates over real-valued signals. Metric temporal logic (MTL) [159] and metric interval temporal logic (MITL) [14] enrich temporal logic with bounded timing intervals, while signal temporal logic (STL) augments these dense-time modalities with predicates over continuous-valued signals [178]. This makes STL especially well suited to trajectory generation and control for physical systems.

A key step toward optimization-based synthesis with STL was the introduction of quantitative semantics. Robust semantics for MITL were proposed in [109], and the now-standard space robustness (SR), together with time robustness, was introduced for STL in [89]. These semantics assign a real-valued robustness score to a formula by recursively combining predicate values through min and max operators, thereby turning logical satisfaction into a numerical quantity that can be optimized [27, 172]. In this way, temporal specifications can be encoded as objective terms or constraints within optimization-based synthesis and control.

Most optimization-oriented work has focused on discrete-time STL (DT-STL), where the signal is evaluated only on a finite temporal grid [59, 123, 128, 173, 187, 196, 198, 223, 233, 244, 245, 247, 292]. Existing DT-STL methods can be understood through a trade-off between exactness, smoothness, and numerical tractability. Mixed-integer encodings preserve Boolean correctness, but their worst-case complexity typically scales poorly with the number of binary variables [233, 244, 245, 247]. Direct use of SR avoids combinatorial structure, but the underlying min and max operators make the resulting objectives and constraints nonsmooth, which can severely hinder gradient-based optimization. This has motivated a large body of work on smooth approximations of DT-STL robustness. Log-sum-exp relaxations were introduced in [223] and extended to cumulative robustness in [128]; softmax-based approximations were proposed in [123]; and polynomial smoothings were used in [187]. While these formulations improve differentiability, they are inherently approximate and typically recover exact Boolean consistency only in an asymptotic limit of the smoothing parameter.

A different line of work has sought to improve numerical behavior by reducing the dependence of robustness on isolated critical samples. This is motivated by two well-known pathologies of standard robustness, namely *locality*, where the robustness value depends on a single time instant, and *masking*, where one subformula dominates a conjunction or disjunction and suppresses information from the others [197]. Averaged STL [10], discrete average space robustness (DASR) and simplified DASR [173], and arithmetic–geometric mean (AGM) robustness [196] all attempt to distribute information more evenly across time and subformulas. These semantics improve certain numerical aspects, but they do not fully resolve the tension between smoothness and exactness. Some retain nonsmooth operators, some require auxiliary soundness constraints, and some apply only to restricted classes of formulas. Weighted generalizations of SR and AGM have also been developed to encode preferences among sub-specifications [59, 198]. Another exact alternative is proposed in [292], but the resulting functions are non-monotone and not $\mathcal{C}^1$, which limits their suitability for numerical optimization.

Against this background, the discrete-time modeling line developed in this dissertation builds on the generalized mean-based smooth robustness (D-GMSR) introduced in [286]. D-GMSR replaces the key min and max operators of standard discrete-time robustness with generalized-mean compositions that are $\mathcal{C}^1$-smooth while preserving the sign of the original robustness semantics. As a result, the resulting parameterization is both sound and complete with respect to the underlying STL semantics, while simultaneously providing first-order sensitivity properties that directly mitigate locality and masking. This yields a smooth, exact, and numerically attractive DT-STL parameterization. More broadly, such smooth exact formulations enable the use of high-performance numerical solvers with convergence guarantees for optimization problems involving STL specifications [1, 92, 147, 291]. Table 3.1 summarizes this landscape and highlights the combination of properties that motivates the generalized mean-based smooth robustness construction adopted in this dissertation.

For continuous-time systems, however, enforcing STL only on a finite temporal grid is often inadequate. For `always`-type specifications, node-wise satisfaction does not preclude inter-sample violations and therefore cannot by itself guarantee safety along the full trajectory. For `eventually`-type specifications, restricting satisfaction to mesh points can be unnecessarily conservative and may degrade optimality, since the desired event may occur between nodes even when no sampled state satisfies the predicate. These issues motivate the study of

	[89]	[223]	[123]	[196]	[292]	D-GMSR
$\mathcal{C}^1$-smoothness	×	✓	✓	×	×	✓
Soundness	✓	○	✓	✓	✓	✓
Completeness	✓	○	○	✓	✓	✓
Monotonicity	✓	✓	✓	✓	×	✓
Locality & Masking	×	△	△	✓	✓	✓

Table 3.1: Comparison with representative robustness measures from the STL literature. Here, ○ indicates a property that is recovered only for very large smoothing parameters, while △ indicates a property that is obtained only for sufficiently small smoothing parameters.

continuous-time STL (CT-STL), where the semantics are enforced over dense-time rather than only over a discretization grid.

A steadily growing literature addresses dense-time STL satisfaction for continuous-time systems. On the control-synthesis side, several works translate STL requirements into time-varying invariance and reachability conditions and enforce them using control barrier functions (CBFs) or related barrier-based controllers. Representative examples include foundational CBF constructions [172], continuous-time model predictive control (MPC) formulations [68], actuation-constrained CBF frameworks [56], and extensions to certain nested formulas [306]. These approaches are valuable for online controller synthesis, but they do not by themselves provide a general smooth parameterization of STL specifications that can be inserted directly into a standard trajectory optimization framework.

On the trajectory optimization side, dense-time guarantees have so far been obtained mainly through a few specialized mechanisms. One approach augments mixed-integer planners with CBF-derived intersample conditions under zero-order hold, thereby restoring dense-time correctness between sampling instants [304]. Another guarantees continuous-time satisfaction for quadrotor missions by planning over sparse waypoint sequences, enforcing strictified sampled-time specifications, and combining these with intersample deviation bounds and a hierarchical tracking architecture [224]. More recent methods optimize directly over continuous trajectory families, including sound mixed-integer formulations over continuous Bézier trajectories for temporally robust multi-agent planning [293], Bézier control-point and acceleration conditions for CT-STL satisfaction with time-varying robustness [311], and their extension to multiple quadrotors through piecewise Bézier reference generation with explicit tracking-error bounds [310]. A recent sampling-based planner also encodes a fragment of STL into a forward-invariant time-varying set and searches for dynamically feasible trajectories within that set [188]. On the semantics side, average-based robustness has been extended to continuous-time signals [197], but these formulations remain nonsmooth and do not by themselves yield a tractable reformulation for standard gradient-based nonlinear programming.

In parallel, continuous-time successive convexification frameworks were developed to guarantee continuous-time satisfaction of path constraints, i.e., `always`-type specifications, in trajectory optimization [104]. These methods augment the system dynamics with penalized path-constraint states, so that continuous-time enforcement of the original path constraints is achieved by enforcing satisfaction of the corresponding augmented dynamical constraints. Their numerical effectiveness has been demonstrated in several applications, including GPU-accelerated trajectory optimization for 6-DoF rocket landing [66], nonlinear model predictive control for obstacle avoidance [287], and trajectory optimization for 6-DoF aircraft approach and landing [155]. This framework was later combined with GMSR in [289] to handle the continuous-time satisfaction of compound state-triggered constraints, which are `implication`-type mission specifications, in a 6-DoF rocket landing problem. Its practical utility was further illustrated in perception-constrained drone motion planning [290] and in a related rocket-landing problem with implication-type constraints solved using an auto-tuned SCP method [195]. In parallel, these ideas were incorporated into SCvxGEN, an automatic code-generation framework for fast embedded trajectory optimization [147]. These developments show that smooth and exact formulations are possible for important subclasses of mission specifications, but they remain focused primarily on `always`-type logic.

Taken together, the existing literature still falls short of a smooth and exact treatment of general CT-STL in trajectory optimization. Available methods either remain node-based, achieve dense-time guarantees only through specialized machinery such as barrier certificates, strictification, mixed-integer encodings, or restricted trajectory families, or apply only to limited subclasses of continuous-time specifications. In particular, a general framework that naturally accommodates richer temporal operators such as `eventually` and `until` while remaining smooth and numerically suitable for gradient-based optimization has been lacking.

The modeling developments consolidated in this chapter address this gap by building on the generalized mean-based smooth robustness framework introduced in [286] and its subsequent continuous-time extensions [284, 285]. At the logical level, D-GMSR provides smooth and exact parameterizations of the Boolean operators of STL [286]. At the temporal level, this chapter develops two complementary realizations of continuous-time STL. The first is an augmentation-based realization, in which temporal aggregation is embedded directly into augmented continuous-time dynamics [284]. In particular, continuous-time satisfaction of `always`-type specifications is handled in the spirit of [104], by augmenting the system with penalized constraint states and enforcing the resulting augmented dynamical constraints. This realization is attractive from an implementation standpoint because it is largely transcription-independent and can be incorporated into existing optimal-control pipelines with minimal structural changes. The second is a dense-time realization, in which CT-STL formulas are evaluated directly on the dense trajectory representation already used in numerical integration [285]. This realization is sound and complete up to the numerical accuracy of the underlying integration scheme. By evaluating predicates on the integration subnodes, the dense-time realization can permit substantially coarser temporal grids without sacrificing inter-sample safety or logical fidelity. In turn, this can reduce the dimension of the resulting nonconvex program, which is often the dominant factor in overall trajectory generation cost [285].

These constructions together provide a smooth and optimization-friendly framework for modeling mission-level specifications in continuous time. In contrast to existing CT-STL approaches that rely on conservative inter-sample bounds, specialized trajectory parameterizations, or mixed-integer formulations, the proposed framework guarantees continuous-time satisfaction of `always`-type specifications, which is critical for safety constraints such as obstacle avoidance, while removing the node-induced conservatism of `eventually`-type specifications by allowing satisfaction at any time within the prescribed interval. Moreover, by inheriting the favorable sensitivity properties of GMSR, the resulting formulations avoid the locality and masking issues that commonly degrade standard quantitative semantics, thereby yielding a more favorable landscape for gradient-based solvers.

The remainder of the chapter is organized as follows. Section 3.2 reviews the syntax of STL together with the standard Boolean and quantitative robustness semantics that serve as the semantic target of the proposed constructions. Section 3.3 develops generalized mean-based smooth robustness for STL, beginning with the logical operators and then extending the construction to dense-time temporal operators. Section 3.4 presents the augmentation-based realization of CT-STL. Section 3.5 introduces weighted extensions that allow preferences among sub-specifications to be encoded. Finally, Section 3.6 summarizes the main theoretical and numerical properties of the resulting robustness measures, including soundness, completeness, monotonicity, locality, masking, and gradient behavior.

3.2 Signal Temporal Logic and standard robustness semantics

This section reviews the syntax of Signal Temporal Logic (STL), together with its standard Boolean and quantitative robustness semantics. The continuous-time presentation serves as the primary semantic reference for this chapter, while the sampled discrete-time restriction is recorded afterward since it forms the starting point for the discrete-time generalized mean-based smooth robustness (D-GMSR) construction developed later.

3.2.1 Signal Temporal Logic

Let

$$x : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_x}$$

be a continuous-time signal. Let μ be an atomic predicate associated with a real-valued predicate function

$$g : \mathbb{R}^{n_x} \to \mathbb{R}.$$

Its Boolean evaluation at time t is defined by

$$(x, t) \models \mu \iff g(x(t)) \geq 0. \tag{3.1}$$

That is, the predicate μ is satisfied at time t if and only if $g(x(t)) \geq 0$.

For numerical optimization, the regularity of the predicate functions is important.

Assumption 3.1. *All predicate functions are $\mathcal{C}^1$-smooth.*

The syntax of an STL formula φ is defined recursively as

$$\varphi \ ::= \ \top \ \mid \ \mu \ \mid \ \neg\varphi \ \mid \ \varphi_1 \wedge \varphi_2 \ \mid \ \varphi_1 \boldsymbol{U}_{[a,b]} \varphi_2, \tag{3.2}$$

where φ_1 and φ_2 are STL subformulas, $\neg$ and $\wedge$ denote negation and conjunction, respectively, and $\boldsymbol{U}_{[a,b]}$ denotes the bounded `until` operator over the interval $[a, b] \subset \mathbb{R}_{\geq 0}$ with $0 \leq a \leq b$ [178].

The standard derived operators are recovered from the primitive syntax via

$$\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2), \tag{3.3}$$

$$\varphi_1 \implies \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2, \tag{3.4}$$

$$\boldsymbol{F}_{[a,b]}\varphi \equiv \top \boldsymbol{U}_{[a,b]}\varphi, \tag{3.5}$$

$$\boldsymbol{G}_{[a,b]}\varphi \equiv \neg \boldsymbol{F}_{[a,b]} \neg\varphi. \tag{3.6}$$

Here, $\boldsymbol{F}_{[a,b]}$ and $\boldsymbol{G}_{[a,b]}$ denote `eventually` and `always`, respectively.

Throughout this chapter, we consider finite-horizon signals and formulas with finite horizon, and we assume that the temporal intervals are chosen so that all evaluations appearing in the semantics are well defined.

3.2.2 Boolean semantics

The Boolean semantics determine whether a signal satisfies a formula at a given time instant. We write

$$(x, t) \models \varphi$$

to denote that the signal x satisfies the formula φ at time t, and

$$(x, t) \not\models \varphi$$

otherwise.

Definition 3.2 (Continuous-time STL Boolean semantics [89, 178]). *For a continuous-time signal $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_x}$ and time $t \in \mathbb{R}_{\geq 0}$, the Boolean satisfaction relation $(x, t) \models \varphi$ is defined recursively as summarized in Table 3.2.*

3.2.3 Standard quantitative robustness semantics

While Boolean semantics provide only a binary `true`/`false` evaluation, optimization-based synthesis requires a real-valued measure of the degree of satisfaction or violation of a specification. The standard choice is the quantitative robustness semantics introduced in [89, 109].

Let $\rho^{\varphi}(x, t)$ denote the robustness of formula φ at time t. These semantics are sign-consistent with Boolean satisfaction in the sense that

$$(x, t) \models \varphi \iff \rho^{\varphi}(x, t) \geq 0. \tag{3.7}$$

Moreover, $\rho^{\varphi}(x, t) > 0$ implies strict satisfaction, $\rho^{\varphi}(x, t) < 0$ implies violation, and $\rho^{\varphi}(x, t) = 0$ corresponds to the boundary between satisfaction and violation.

Definition 3.3 (Continuous-time standard robustness [89, 109]). *The standard quantitative robustness semantics of STL are defined recursively as summarized in Table 3.3.*

Formula φ	**Boolean semantics** $(x,t) \models \varphi$
$\top$	`true`
μ	$g(x(t)) \geq 0$
$\neg\varphi$	$\neg\big((x,t) \models \varphi\big)$
$\varphi_1 \wedge \varphi_2$	$(x,t) \models \varphi_1 \ \wedge \ (x,t) \models \varphi_2$
$\varphi_1 \vee \varphi_2$	$(x,t) \models \varphi_1 \ \vee \ (x,t) \models \varphi_2$
$\varphi_1 \implies \varphi_2$	$\neg\big((x,t) \models \varphi_1\big) \ \vee \ (x,t) \models \varphi_2$
$\boldsymbol{G}_{[a,b]}\varphi$	$\forall\, t' \in [t+a, t+b],\ (x,t') \models \varphi$
$\boldsymbol{F}_{[a,b]}\varphi$	$\exists\, t' \in [t+a, t+b]$ s.t. $(x,t') \models \varphi$
$\varphi_1 \boldsymbol{U}_{[a,b]} \varphi_2$	$\exists\, t_1 \in [t+a, t+b]$ s.t. $(x,t_1) \models \varphi_2$
	and $\forall\, t_2 \in [t, t_1],\ (x,t_2) \models \varphi_1$

Table 3.2: Continuous-time Boolean semantics for STL.

Because the signals and predicate functions considered in this dissertation are continuous and the temporal intervals are compact, the extrema in Table 3.3 are attained. Accordingly, the standard continuous-time robustness can be written using min and max directly rather than inf and sup.

The standard robustness semantics provide an exact quantitative characterization of STL satisfaction. For example, the robustness of an `always` formula is determined by the least robust point in the prescribed interval, whereas the robustness of an `eventually` formula is determined by the most favorable point in that interval. Likewise, the robustness of an `until` formula captures the best admissible tradeoff between maintaining the first subformula and achieving the second one within the specified time window.

Formula φ	**Quantitative robustness** $\rho^{\varphi}(x,t)$
μ	$g(x(t))$
$\neg\varphi$	$-\rho^{\varphi}(x,t)$
$\varphi_1 \wedge \varphi_2$	$\min\Big(\rho^{\varphi_1}(x,t),\ \rho^{\varphi_2}(x,t)\Big)$
$\varphi_1 \vee \varphi_2$	$\max\Big(\rho^{\varphi_1}(x,t),\ \rho^{\varphi_2}(x,t)\Big)$
$\varphi_1 \implies \varphi_2$	$\max\Big(-\rho^{\varphi_1}(x,t),\ \rho^{\varphi_2}(x,t)\Big)$
$\boldsymbol{G}_{[a,b]}\varphi$	$\min_{t' \in [t+a,t+b]} \rho^{\varphi}(x,t')$
$\boldsymbol{F}_{[a,b]}\varphi$	$\max_{t' \in [t+a,t+b]} \rho^{\varphi}(x,t')$
$\varphi_1 \boldsymbol{U}_{[a,b]} \varphi_2$	$\max_{t_1 \in [t+a,t+b]} \min\Big(\rho^{\varphi_2}(x,t_1),\ \min_{t_2 \in [t,t_1]} \rho^{\varphi_1}(x,t_2)\Big)$

Table 3.3: Standard quantitative robustness semantics for continuous-time STL.

3.2.4 Discrete-time restriction

For optimization on sampled trajectories, STL is often evaluated only on a finite temporal grid. This leads to the standard discrete-time STL semantics and to the corresponding discrete-time space robustness (D-SR), which serves as the starting point for D-GMSR [286].

Let

$$x : \mathbb{Z}_{\geq 0} \to \mathbb{R}^{n_x}$$

be a discrete-time signal. The logical recursion is unchanged, while the temporal operators are evaluated over finite index sets. In particular, the discrete-time robustness of the temporal operators is given by

$$\rho^{\boldsymbol{F}_{[a,b]}\varphi}(x,k) = \max\Big(\rho^{\varphi}(x,k+a),\ \rho^{\varphi}(x,k+a+1),\ \ldots,\ \rho^{\varphi}(x,k+b)\Big), \tag{3.8}$$

$$\rho^{\boldsymbol{G}_{[a,b]}\varphi}(x,k) = \min\Big(\rho^{\varphi}(x,k+a),\ \rho^{\varphi}(x,k+a+1),\ \ldots,\ \rho^{\varphi}(x,k+b)\Big), \tag{3.9}$$

$$\rho^{\varphi_1 \boldsymbol{U}_{[a,b]}\varphi_2}(x,k) = \max\Big(\min\Big(\rho^{\boldsymbol{G}_{[a,a]}\varphi_1}(x,k),\ \rho^{\varphi_2}(x,k+a)\Big), \\ \min\Big(\rho^{\boldsymbol{G}_{[a,a+1]}\varphi_1}(x,k),\ \rho^{\varphi_2}(x,k+a+1)\Big),\dots, \\ \min\Big(\rho^{\boldsymbol{G}_{[a,b]}\varphi_1}(x,k),\ \rho^{\varphi_2}(x,k+b)\Big)\Big), \tag{3.10}$$

with the logical operators still represented by the same min / max recursion as in Table 3.3.

A common way to smooth the discrete-time space robustness (D-SR) semantics is to replace the key min and max operators by smooth approximations. In particular, the discrete-time smooth space robustness (D-SSR) construction uses the smoothed operators

$$\widetilde{\min}_\kappa(y) \coloneqq -\frac{1}{\kappa}\log\left(\sum_{i=1}^{n} e^{-\kappa y_i}\right), \tag{3.11a}$$

$$\widetilde{\max}_\kappa(y) \coloneqq \frac{\sum_{i=1}^{n} y_i e^{\kappa y_i}}{\sum_{i=1}^{n} e^{\kappa y_i}}, \tag{3.11b}$$

where $y \in \mathbb{R}^n$ and $\kappa > 0$ is a sharpness parameter that controls the approximation accuracy [123, 223]. The resulting smoothed robustness is sound and $\mathcal{C}^1$-smooth, and converges pointwise to the standard discrete-time robustness as the sharpness parameter increases. However, as discussed later in this chapter, these sharp smooth approximations remain vulnerable to locality and masking because their gradients become concentrated on a small subset of active samples or subformulas.

The discrete-time restriction is computationally natural, but it introduces a structural limitation for continuous-time trajectory optimization. Even if an `always`-type specification is satisfied at all nodes, violations may still occur between nodes, and `eventually`-type specifications may become unnecessarily conservative when satisfaction is artificially restricted to the sampling grid. These issues motivate the dense-time constructions developed in the next sections.

The standard semantics reviewed above, therefore, play two roles in this chapter. First, they define the exact semantic target that the proposed smooth constructions must preserve. Second, their nested min / max structure reveals the two main modeling challenges addressed next, namely the nonsmoothness of the logical and temporal operators and the mismatch between node-based evaluation and dense-time mission requirements. Section 3.3 addresses the first of these through generalized mean-based smooth robustness, and then extends the resulting framework to dense-time temporal operators.

3.3 GMSR: A smooth and exact parameterization of CT-STL specifications

This section develops the baseline generalized mean-based smooth robustness (GMSR) constructions used throughout the remainder of the chapter. We begin with the logical operators, where GMSR replaces the nonsmooth min / max operators of the standard robustness semantics with smooth and exact generalized-mean compositions. We then extend the same idea to dense-time temporal operators by evaluating the resulting smooth logical constructions over a dense representation of the continuous-time signal.

Remark 3.4 (Simplified version of GMSR). The presentation below uses the simplified uniform-weight form adopted in the CT-STL line of work [284, 286]; the more general weighted and curvature-controlled family is deferred to Section 3.5.

3.3.1 Logical operators

The starting point is the generalized mean-based smooth and exact parameterization of conjunction and disjunction introduced for discrete-time STL in [286].

For a vector $y \in \mathbb{R}^n$, define

$$^{\wedge}h^c(y) \;\coloneqq\; \Big(M_0^c\big(|y|_+^2\big)\Big)^{\frac{1}{2}} - \Big(M_1^c\big(|y|_-^2\big)\Big)^{\frac{1}{2}}, \tag{3.12}$$

and

$$^{\vee}h^c(y) \;\coloneqq\; -\,^{\wedge}h^c(-y), \tag{3.13}$$

where $c \in \mathbb{R}_{++}$ is a positive regularization shift, and

$$|y|_+ \;\coloneqq\; \max(y, 0), \qquad |y|_- \;\coloneqq\; \min(y, 0) \tag{3.14}$$

are applied elementwise. The generalized means in (3.12) are

$$M_0^c(z) \coloneqq \left(c^n + \prod_{i=1}^{n} z_i\right)^{1/n}, \tag{3.15}$$

$$M_1^c(z) \coloneqq c + \frac{1}{n}\sum_{i=1}^{n} z_i, \tag{3.16}$$

for $z \in \mathbb{R}^n_{\geq 0}$.

The function ${}^{\wedge}h^c$ is designed to play the role of a smooth exact surrogate for conjunction, while ${}^{\vee}h^c$ plays the corresponding role for disjunction. Both can be evaluated in $O(n)$ elementary operations, so their computational cost scales linearly with the number of inputs.

The key property is that these functions preserve the sign pattern of conjunction and disjunction exactly:

$${}^{\wedge}h^c(y) \geq 0 \iff y_i \geq 0, \quad \forall i \in \{1, \ldots, n\}, \tag{3.17}$$

$${}^{\vee}h^c(y) \geq 0 \iff \exists\, i \in \{1, \ldots, n\} \text{ such that } y_i \geq 0. \tag{3.18}$$

Thus, ${}^{\wedge}h^c$ and ${}^{\vee}h^c$ provide smooth and exact parameterizations of conjunction and disjunction. Formal soundness and completeness statements are deferred to Section 3.6.

Remark 3.5 (Sign-dependent behavior and the role of c). The values of ${}^{\wedge}h^c$ and ${}^{\vee}h^c$ depend on the signs of the entries of y as follows:

$${}^{\wedge}h^c(y) = \begin{cases} \left(M_0^c(|y|_+^2)\right)^{1/2} - c^{1/2}, & y_i \geq 0,\ \forall i, \\ c^{1/2} - \left(M_1^c(|y|_-^2)\right)^{1/2}, & \text{otherwise}, \end{cases} \tag{3.19}$$

$${}^{\vee}h^c(y) = \begin{cases} c^{1/2} - \left(M_0^c(|y|_-^2)\right)^{1/2}, & y_i \leq 0,\ \forall i, \\ \left(M_1^c(|y|_+^2)\right)^{1/2} - c^{1/2}, & \text{otherwise}. \end{cases} \tag{3.20}$$

Accordingly, in the all-positive orthant, ${}^{\wedge}h^c$ is governed by the shifted geometric-mean branch, whereas in the mixed-sign or violating regime, it is governed by the shifted arithmetic-mean branch. The situation is reversed for ${}^{\vee}h^c$.

The parameter c regularizes the transition near the switching regions. If $c = 0$, then M_0^c and M_1^c reduce to the geometric and arithmetic means, respectively. In that limiting

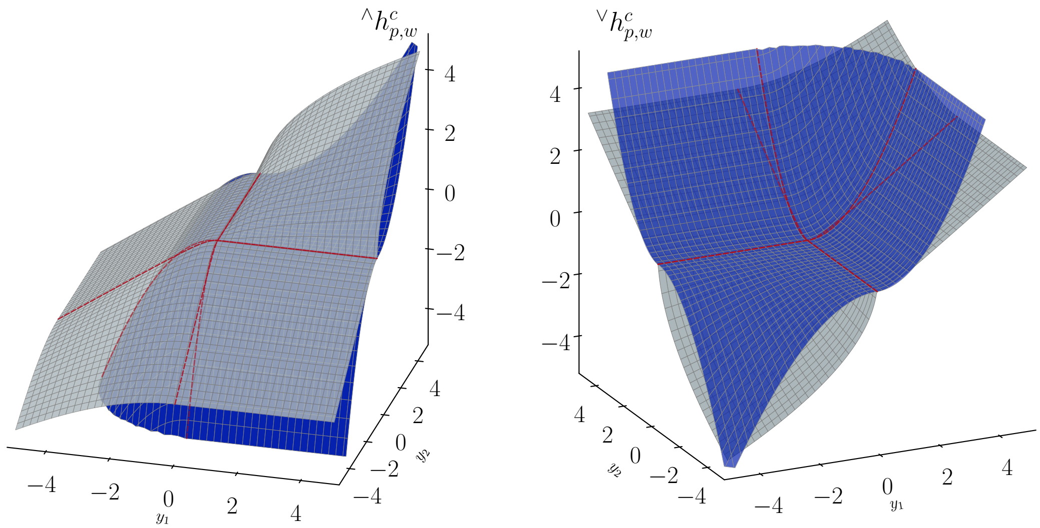


Figure 3.1: Illustration of the logical GMSR key functions $^{\wedge}h^c$ and $^{\vee}h^c$ for a two-dimensional input $y = (y_1, y_2)$ and two representative values of the shift parameter c. In both cases, $^{\wedge}h^c(y) \geq 0$ if and only if both inputs are nonnegative, whereas $^{\vee}h^c(y) \geq 0$ if and only if at least one input is nonnegative. Increasing c smooths and flattens the transition near the switching regions.

case, the geometric branch can produce unbounded derivatives near zero. For every $c > 0$, the functions $^{\wedge}h^c$ and $^{\vee}h^c$ become $\mathcal{C}^1$-smooth while preserving exact sign consistency. As c increases, the transition near the switching region becomes flatter and the gradients become smaller; as c decreases, the functions more closely reflect the unshifted generalized-mean behavior, but with larger gradients near zero. In practice, it is typically preferable to choose a small positive value of c, so as to retain smoothness and bounded gradients without making the robustness landscape overly flat.

The behavior described in Remark 3.5 is illustrated in Figure 3.1, which shows how the functions change as the shift parameter c varies.

Let $\Gamma^{\varphi}_{\boldsymbol{c}}(x, t)$ denote the GMSR robustness of formula φ at time t along the continuous-time

Formula φ	GMSR robustness $\Gamma^{\varphi}_{\boldsymbol{c}}(x,t)$
μ	$g(x(t))$
$\neg\varphi$	$-\Gamma^{\varphi}_{\boldsymbol{c}}(x,t)$
$\varphi_1 \wedge \varphi_2$	${}^{\wedge}h^{c_1}\Big(\Gamma^{\varphi_1}_{\boldsymbol{c}}(x,t), \Gamma^{\varphi_2}_{\boldsymbol{c}}(x,t)\Big)$
$\varphi_1 \vee \varphi_2$	${}^{\vee}h^{c_1}\Big(\Gamma^{\varphi_1}_{\boldsymbol{c}}(x,t), \Gamma^{\varphi_2}_{\boldsymbol{c}}(x,t)\Big)$
$\varphi_1 \Rightarrow \varphi_2$	${}^{\vee}h^{c_1}\Big(-\Gamma^{\varphi_1}_{\boldsymbol{c}}(x,t), \Gamma^{\varphi_2}_{\boldsymbol{c}}(x,t)\Big)$

Table 3.4: Smooth and exact GMSR parameterization of the logical operators of STL.

signal x, where $\boldsymbol{c}$ collects the positive shifts associated with the recursive GMSR compositions appearing in the parse tree of φ. The logical operators are then parameterized as summarized in Table 3.4.

Accordingly, for predicates of the form $\varphi_i := (g_i(x) \geq 0)$,

$$(x,t) \models \bigwedge_{i=1}^{n} \varphi_i \iff {}^{\wedge}h^{c}\Big(g_1(x(t)), \ldots, g_n(x(t))\Big) \geq 0, \tag{3.21}$$

$$(x,t) \models \bigvee_{i=1}^{n} \varphi_i \iff {}^{\vee}h^{c}\Big(g_1(x(t)), \ldots, g_n(x(t))\Big) \geq 0. \tag{3.22}$$

More generally, the recursion in Table 3.4 yields

$$(x,t) \models \varphi \iff \Gamma^{\varphi}_{\boldsymbol{c}}(x,t) \geq 0 \tag{3.23}$$

for any formula φ constructed from predicates through the logical operators shown in the table.

Remark 3.6. The sign conditions in (3.17)–(3.18) are equivalently expressed by the simplified tests

$$^{\wedge}h^c(y) \geq 0 \iff \sum_{i=1}^{n} |y_i|_-^2 = 0, \tag{3.24}$$

$$^{\vee}h^c(y) \geq 0 \iff \prod_{i=1}^{n} |y_i|_-^2 = 0. \tag{3.25}$$

These equivalent conditions clarify the exactness of the logical parameterization. However, the full GMSR expressions (3.12)–(3.13) are numerically preferable, since the generalized-mean normalization and outer square roots yield better-scaled robustness values and gradients than the raw sum/product tests.

3.3.2 Dense-time temporal extension

The logical constructions above resolve the nonsmoothness of the Boolean operators. To obtain a smooth and exact CT-STL parameterization, it remains to extend GMSR from discrete-time temporal operators to dense-time temporal operators. The key idea, developed in the CT-STL line of work [284, 285], is to evaluate the temporal operators directly on the dense trajectory representation already generated by numerical integration, rather than on a separate coarse sampling grid. This avoids introducing an additional STL discretization and allows the specifications to be enforced on the same dense numerical trajectory representation used by the optimizer.

Suppose the trajectory is parameterized by nodal variables

$$X := \{x_k\}_{k=1}^{K}, \qquad U := \{u_k\}_{k=1}^{K},$$

over a grid

$$0 = t_1 < t_2 < \cdots < t_K = T.$$

On each interval $[t_k, t_{k+1}]$, numerical integration under the chosen control parameterization produces intermediate subnodes

$$t_k = t_k^0 < t_k^1 < \cdots < t_k^N = t_{k+1}.$$

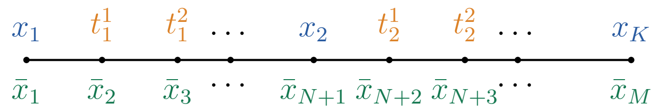


Figure 3.2: Relation between nodal states, integration subnodes, and the flattened dense-time trajectory representation used for CT-STL evaluation. The optimization variables remain the nodal states and controls; the dense-time states are generated by numerical integration from those nodal variables.

Flattening these subnodes over the horizon yields the dense-time grid

$$\{\bar{t}_m\}_{m=1}^{M}, \qquad M = (K-1)N + 1,$$

with associated dense-time states

$$\bar{x}_m := x(\bar{t}_m), \qquad m = 1, \dots, M.$$

The relation between the nodal states, interval subnodes, and flattened dense-time states is illustrated in Figure 3.2.

The optimization variables remain only the nodal variables in (X, U). The dense-time states $\bar{x}_m$ are not independent decision variables; rather, they are generated by numerical integration and are therefore well-defined functions of (X, U). Moreover, provided the numerical integration scheme is differentiable, each $\bar{x}_m$ is differentiable with respect to (X, U); in practice, these derivatives are obtained from the same interval-wise variational equations used in the multiple-shooting discretization of Chapter 2. Consequently, the CT-STL robustness values evaluated on the dense-time grid are themselves smooth functions of the original discretized trajectory variables.

For an atomic predicate $\mu := (g(x) \geq 0)$, define

$$\tilde{\Gamma}^{\mu}(X, U; \bar{t}_m) \; := \; g(\bar{x}_m), \tag{3.26}$$

where the dependence on (X, U) enters entirely through the dense numerical trajectory.

Formula φ	**Dense-time GMSR robustness at** $\bar{t}_\ell$
$\boldsymbol{F}_{[a,b]}\varphi$	${}^{\vee}h^{c_1}\Big(\big[\,\tilde{\Gamma}^{\varphi}_{\boldsymbol{c}}(X,U;\bar{t}_m)\,\big]_{m\in\mathcal{I}_\ell[a,b]}\Big)$
$\boldsymbol{G}_{[a,b]}\varphi$	${}^{\wedge}h^{c_1}\Big(\big[\,\tilde{\Gamma}^{\varphi}_{\boldsymbol{c}}(X,U;\bar{t}_m)\,\big]_{m\in\mathcal{I}_\ell[a,b]}\Big)$

Table 3.5: Exact dense-time GMSR extension of the `always` and `eventually` operators on a dense trajectory representation.

For composite formulas, the values $\tilde{\Gamma}^{\varphi}_{\boldsymbol{c}}(X,U;\bar{t}_m)$ are obtained recursively from the logical constructions of Section 3.3.1.

Now consider a formula evaluated at dense-time node $\bar{t}_\ell$ and a temporal interval $[a,b]$. Define the corresponding dense-time index set

$$\mathcal{I}_\ell[a,b] \;\coloneqq\; \Big\{m\in\{1,\dots,M\} : \bar{t}_m\in[\bar{t}_\ell+a,\;\bar{t}_\ell+b]\Big\}, \tag{3.27}$$

assuming the interval lies within the time horizon. If a boundary point is not already present on the dense grid, it may be inserted by locally refining the corresponding integration subinterval.

The exact dense-time GMSR realizations of `always` and `eventually` are summarized in Table 3.5.

Thus, `always` is realized as a smooth dense-time conjunction over all samples in the interval, while `eventually` is realized as a smooth dense-time disjunction over those samples. Similarly, the `until` operator is parameterized by introducing, for each admissible witness time $\bar{t}_m$,

$$z_m \;\coloneqq\; {}^{\wedge}h^{c_2}\Big(\tilde{\Gamma}^{\varphi_2}_{\boldsymbol{c}}(X,U;\bar{t}_m),\;{}^{\wedge}h^{c_3}\Big(\big[\,\tilde{\Gamma}^{\varphi_1}_{\boldsymbol{c}}(X,U;\bar{t}_q)\,\big]_{q=\ell}^{m}\Big)\Big), \tag{3.28}$$

and then applying a dense-time disjunction over all admissible witness indices:

$$\tilde{\Gamma}^{\varphi_1\boldsymbol{U}_{[a,b]}\varphi_2}_{\boldsymbol{c}}(X,U;\bar{t}_\ell) \;\coloneqq\; {}^{\vee}h^{c_1}\Big(\big[\,z_m\,\big]_{m\in\mathcal{I}_\ell[a,b]}\Big). \tag{3.29}$$

This construction mirrors the standard robustness semantics exactly on the dense trajectory representation:

$$(x, \bar{t}_\ell) \models \boldsymbol{F}_{[a,b]}\varphi \iff \tilde{\Gamma}_{\boldsymbol{c}}^{\boldsymbol{F}_{[a,b]}\varphi}(X, U; \bar{t}_\ell) \geq 0, \tag{3.30}$$

$$(x, \bar{t}_\ell) \models \boldsymbol{G}_{[a,b]}\varphi \iff \tilde{\Gamma}_{\boldsymbol{c}}^{\boldsymbol{G}_{[a,b]}\varphi}(X, U; \bar{t}_\ell) \geq 0, \tag{3.31}$$

$$(x, \bar{t}_\ell) \models \varphi_1 \boldsymbol{U}_{[a,b]}\varphi_2 \iff \tilde{\Gamma}_{\boldsymbol{c}}^{\varphi_1 \boldsymbol{U}_{[a,b]}\varphi_2}(X, U; \bar{t}_\ell) \geq 0. \tag{3.32}$$

Accordingly, the dense-time realization is smooth with respect to the optimization variables and exact on the dense numerical trajectory representation. When the dense grid is generated by numerical integration of the continuous-time dynamics, the only remaining discrepancy with the underlying continuous-time signal is the numerical integration error itself.

This dense-time viewpoint has two practical consequences. First, it guarantees inter-sample satisfaction of `always`-type specifications on the dense numerical trajectory representation, which is essential for path constraints such as obstacle avoidance. Second, it removes the node-induced conservatism of `eventually`-type specifications by allowing satisfaction at any dense-time sample in the interval, rather than forcing satisfaction only at the coarse transcription nodes. Thus, although the dense-time realization requires additional predicate evaluations on the integration subnodes, it can also permit substantially coarser transcription grids without sacrificing logical fidelity or inter-sample safety [285].

From an implementation standpoint, the derivatives required by gradient-based solvers are obtained directly by the chain rule. For an atomic predicate,

$$\nabla_{(X,U)}\tilde{\Gamma}^{\mu}(X, U; \bar{t}_m) = \nabla g(\bar{x}_m)\, \nabla_{(X,U)}\bar{x}_m, \tag{3.33}$$

and for composite formulas the same principle applies recursively through the smooth logical operators ${}^{\wedge}h^c$ and ${}^{\vee}h^c$. If $\bar{x}_m$ lies on the interval $[t_k, t_{k+1}]$, then $\bar{x}_m$ depends only on the local nodal variables x_k, u_k, and u_{k+1}. Its Jacobians with respect to those variables are obtained by evaluating at the corresponding intermediate subnode the same variational sensitivity equations used to compute the multiple-shooting flow-map Jacobians in Section 2.5; see

(2.28)–(2.29). Therefore, the dense-time CT-STL robustness is not only smooth in principle, but also directly differentiable in practice with respect to the optimization variables.

3.4 Augmentation-based realization of CT-STL specifications

Section 3.3 extended GMSR to dense-time temporal operators by evaluating the STL semantics directly on a dense numerical representation of the continuous-time trajectory. That realization is exact on the chosen dense-time trajectory representation, up to the numerical accuracy of the underlying integration scheme, but it is tied to the discretization used to represent the signal. An alternative realization is obtained by embedding temporal aggregation directly into augmented continuous-time dynamics. This augmentation-based viewpoint is largely transcription-independent and can therefore be incorporated into an existing optimal-control formulation by augmenting the state alone. The construction was first developed for continuous-time path constraints in [104] and then extended to CT-STL temporal operators in [284].

For notational simplicity, the temporal formulas in this section are presented as being evaluated at time 0. The construction at a general evaluation time t_0 is obtained by replacing the signal $x(\cdot)$ with the shifted signal

$$x_{t_0}(s) \coloneqq x(t_0 + s), \qquad s \geq 0,$$

and applying the same formulas to x_{t_0}.

3.4.1 Continuous-time temporal aggregates

In the discrete-time GMSR constructions, temporal operators are built by applying smooth conjunction/disjunction operators to a finite collection of robustness values over a temporal window. This leads to generalized means involving finite products and finite sums. The augmentation-based continuous-time realization replaces these finite products and sums by their continuous-time analogues, namely a normalized geometric integral [22, 265] and a normalized ordinary integral.

Let $I = [a, b] \subset \mathbb{R}_{\geq 0}$ be a compact interval with length

$$|I| \coloneqq b - a.$$

For a scalar signal $y : I \to \mathbb{R}$, define

$$\mathcal{M}^c_{0,\varepsilon}(y; I) \coloneqq c + \exp\left(\frac{1}{|I|}\int_a^b \log\Big(|y(t)|_+^2 + \varepsilon\Big)\, dt\right), \tag{3.34}$$

$$\mathcal{M}^c_1(y; I) \coloneqq c + \frac{1}{|I|}\int_a^b |y(t)|_-^2\, dt, \tag{3.35}$$

where $c \in \mathbb{R}_{++}$ is the positive GMSR shift inherited from the logical construction, and $\varepsilon \in \mathbb{R}_{++}$ is an additional regularization used only in the geometric-integral term.

The quantity in (3.34) is the continuous-time analogue of the geometric-mean branch in the discrete GMSR construction, while (3.35) is the continuous-time analogue of the arithmetic-mean branch. Accordingly, the continuous-time conjunction/disjunction surrogates are

$$^{\wedge}\mathcal{H}^c_\varepsilon(y; I) \coloneqq \Big(\mathcal{M}^c_{0,\varepsilon}(y; I)\Big)^{1/2} - \Big(\mathcal{M}^c_1(y; I)\Big)^{1/2}, \tag{3.36}$$

$$^{\vee}\mathcal{H}^c_\varepsilon(y; I) \coloneqq -\,^{\wedge}\mathcal{H}^c_\varepsilon(-y; I). \tag{3.37}$$

The shift $\varepsilon > 0$ is needed because the geometric-integral branch involves a logarithm, and $|y(t)|_+^2$ may vanish on part or all of the interval. Thus, the augmentation-based realization is not strictly exact for fixed $\varepsilon > 0$. Its only source of approximation is this regularization, and the resulting effect can be made arbitrarily small in practice, as discussed in Section 3.4.3.

A useful feature of the augmentation-based construction is that the geometric-integral term admits two equivalent continuous-time realizations.

Multiplicative realization. Introduce the auxiliary state

$$\dot{\eta}(t) = \frac{1}{|I|}\,\eta(t)\,\log\Big(|y(t)|_+^2 + \varepsilon\Big), \qquad \eta(a) = 1. \tag{3.38}$$

Then

$$\eta(b) = \exp\left(\frac{1}{|I|}\int_a^b \log\Big(|y(t)|_+^2 + \varepsilon\Big)\, dt\right), \tag{3.39}$$

so that

$$\mathcal{M}^{c}_{0,\varepsilon}(y; I) = c + \eta(b). \tag{3.40}$$

Because the dynamics are normalized by $|I|$, the state evolves on the scale of the geometric mean rather than the geometric product.

Logarithmic realization. Alternatively, define

$$\dot{\zeta}(t) = \frac{1}{|I|} \log\Big(|y(t)|_+^2 + \varepsilon\Big), \qquad \zeta(a) = 0. \tag{3.41}$$

Then

$$\zeta(b) = \frac{1}{|I|} \int_a^b \log\Big(|y(t)|_+^2 + \varepsilon\Big)\, dt, \qquad \eta(b) = e^{\zeta(b)}. \tag{3.42}$$

The two realizations are equivalent in exact arithmetic. In the sequel, we use the multiplicative realization because it embeds directly into the augmented dynamics in a compact form.

The additive realization of the arithmetic-integral term is the same as in [104]:

$$\dot{\xi}(t) = \frac{1}{|I|} |y(t)|_-^2, \qquad \xi(a) = 0, \tag{3.43}$$

so that

$$\mathcal{M}^{c}_{1}(y; I) = c + \xi(b). \tag{3.44}$$

3.4.2 Auxiliary-state realization of temporal specifications

Let $\Gamma^{\varphi}_{\boldsymbol{c}}(x, t)$ denote the logical GMSR robustness introduced in Section 3.3.1. The augmentation-based realizations of the CT-STL temporal operators are summarized in Table 3.6.

Table 3.6 shows that the temporal operators are still built from the same conjunction/disjunction pattern as in the logical and dense-time constructions. The difference is that finite aggregations over samples are replaced by continuous-time integral surrogates. These formulas become practically useful once the corresponding temporal aggregates are realized

Formula φ	Augmentation-based robustness at time 0
$\boldsymbol{G}_I\varphi$	$\Gamma^{\boldsymbol{G}_I\varphi}_{\boldsymbol{c},\varepsilon}(x,0) = {}^{\wedge}\mathcal{H}^{c_1}_{\varepsilon}\Big(t \mapsto \Gamma^{\varphi}_{\boldsymbol{c}}(x,t);\, I\Big)$
$\boldsymbol{F}_I\varphi$	$\Gamma^{\boldsymbol{F}_I\varphi}_{\boldsymbol{c},\varepsilon}(x,0) = {}^{\vee}\mathcal{H}^{c_1}_{\varepsilon}\Big(t \mapsto \Gamma^{\varphi}_{\boldsymbol{c}}(x,t);\, I\Big)$
$\varphi_1\boldsymbol{U}_I\varphi_2$	$\Gamma^{\varphi_1\boldsymbol{U}_I\varphi_2}_{\boldsymbol{c},\varepsilon}(x,0) = {}^{\vee}\mathcal{H}^{c_1}_{\varepsilon}\Big(z(\cdot);\, I\Big),$
	$z(t) = {}^{\wedge}h^{c_2}\Big(\Gamma^{\varphi_2}_{\boldsymbol{c}}(x,t),\, q_{\varphi_1}(t)\Big),$
	$q_{\varphi_1}(t) = {}^{\wedge}\mathcal{H}^{c_3}_{\varepsilon}\Big(s \mapsto \Gamma^{\varphi_1}_{\boldsymbol{c}}(x,s);\, [0,t]\Big)$

Table 3.6: Augmentation-based realization of the CT-STL temporal operators.

through auxiliary continuous-time states. We now give these realizations explicitly for `always`, `eventually`, and `until`.

Let $\mathbf{1}_I(t)$ denote the indicator of the interval $I = [a,b]$, i.e.,

$$\mathbf{1}_I(t) = \begin{cases} 1, & t \in I, \\ 0, & t \notin I. \end{cases}$$

`always`. For $\psi := \boldsymbol{G}_I\varphi$, define the augmented state

$$x^{\psi}(t) := \Big(x(t), \eta_G(t), \xi_G(t)\Big),$$

with auxiliary dynamics

$$\dot{\eta}_G(t) = \frac{1}{|I|}\mathbf{1}_I(t)\,\eta_G(t)\,\log\Big(|\Gamma^{\varphi}_{\boldsymbol{c}}(x,t)|^2_+ + \varepsilon\Big), \qquad \eta_G(0) = 1, \tag{3.45}$$

$$\dot{\xi}_G(t) = \frac{1}{|I|}\mathbf{1}_I(t)\,|\Gamma^{\varphi}_{\boldsymbol{c}}(x,t)|^2_-, \qquad \xi_G(0) = 0. \tag{3.46}$$

Then, for any terminal time $t_{\mathrm{f}} \geq b$,

$$\Gamma^{\boldsymbol{G}_I\varphi}_{\boldsymbol{c},\varepsilon}(x,0) = \Big(c + \eta_G(t_{\mathrm{f}})\Big)^{1/2} - \Big(c + \xi_G(t_{\mathrm{f}})\Big)^{1/2}. \tag{3.47}$$

`eventually`. For $\psi := \boldsymbol{F}_I\varphi$, define the augmented state

$$x^{\psi}(t) := \Big(x(t), \eta_F(t), \xi_F(t)\Big),$$

with auxiliary dynamics

$$\dot{\eta}_F(t) = \frac{1}{|I|}\mathbf{1}_I(t)\,\eta_F(t)\,\log\Big(|\Gamma^{\varphi}_{\boldsymbol{c}}(x,t)|^2_- + \varepsilon\Big), \qquad \eta_F(0) = 1, \tag{3.48}$$

$$\dot{\xi}_F(t) = \frac{1}{|I|}\mathbf{1}_I(t)\,|\Gamma^{\varphi}_{\boldsymbol{c}}(x,t)|^2_+, \qquad \xi_F(0) = 0. \tag{3.49}$$

Then, for any terminal time $t_{\mathrm{f}} \geq b$,

$$\Gamma^{\boldsymbol{F}_I\varphi}_{\boldsymbol{c},\varepsilon}(x,0) = \Big(c + \xi_F(t_{\mathrm{f}})\Big)^{1/2} - \Big(c + \eta_F(t_{\mathrm{f}})\Big)^{1/2}. \tag{3.50}$$

Remark 3.7 (Relation to continuous-time path constraints). When φ is an atomic predicate corresponding to a continuous-time path constraint, the state ξ_G in (3.46) recovers the path-constraint accumulation mechanism developed in [104]. Thus, the augmentation-based realization of $\boldsymbol{G}_I\varphi$ extends the continuous-time path-constraint construction by supplementing the negative-part integral with the geometric-margin term inherited from GMSR.

`until`. The `until` operator requires both a running prefix condition and an outer witness condition. For

$$\psi := \varphi_1 \boldsymbol{U}_I \varphi_2, \qquad I = [a,b],$$

the first step is to construct a continuous-time surrogate for the statement that φ_1 has held on the prefix interval $[0,t]$.

Introduce the prefix states η_1 and ξ_1 via

$$\dot{\eta}_1(t) = \eta_1(t)\,\log\Big(|\Gamma^{\varphi_1}_{\boldsymbol{c}}(x,t)|^2_+ + \varepsilon\Big), \qquad \eta_1(0) = 1, \tag{3.51}$$

$$\dot{\xi}_1(t) = |\Gamma^{\varphi_1}_{\boldsymbol{c}}(x,t)|^2_-, \qquad \xi_1(0) = 0. \tag{3.52}$$

To convert these cumulative quantities into prefix-normalized aggregates on $[0, t]$, define

$$\widehat{\eta}_1(t) = \begin{cases} \eta_1(t)^{1/t}, & t > 0, \\ |\Gamma_{\boldsymbol{c}}^{\varphi_1}(x, 0)|_+^2 + \varepsilon, & t = 0, \end{cases} \tag{3.53}$$

$$\widehat{\xi}_1(t) = \begin{cases} \xi_1(t)/t, & t > 0, \\ |\Gamma_{\boldsymbol{c}}^{\varphi_1}(x, 0)|_-^2, & t = 0, \end{cases} \tag{3.54}$$

and set

$$q_{\varphi_1}(t) = \Big(c_3 + \widehat{\eta}_1(t)\Big)^{1/2} - \Big(c_3 + \widehat{\xi}_1(t)\Big)^{1/2}. \tag{3.55}$$

This is a smooth continuous-time surrogate for the statement that φ_1 has held on the prefix interval $[0, t]$.

Next, combine this prefix signal with φ_2 at each time using the logical GMSR conjunction

$$z(t) = {}^\wedge h^{c_2}\Big(\Gamma_{\boldsymbol{c}}^{\varphi_2}(x, t),\, q_{\varphi_1}(t)\Big). \tag{3.56}$$

The outer witness condition is then realized by an `eventually`-type augmentation over the interval I. Introduce auxiliary states η_U and ξ_U satisfying

$$\dot{\eta}_U(t) = \frac{1}{|I|}\mathbf{1}_I(t)\, \eta_U(t)\, \log\Big(|z(t)|_-^2 + \varepsilon\Big), \qquad \eta_U(0) = 1, \tag{3.57}$$

$$\dot{\xi}_U(t) = \frac{1}{|I|}\mathbf{1}_I(t)\, |z(t)|_+^2, \qquad \xi_U(0) = 0. \tag{3.58}$$

Then, for any terminal time $t_{\mathrm{f}} \geq b$,

$$\Gamma_{\boldsymbol{c},\varepsilon}^{\varphi_1 \boldsymbol{U}_I \varphi_2}(x, 0) = \Big(c_1 + \xi_U(t_{\mathrm{f}})\Big)^{1/2} - \Big(c_1 + \eta_U(t_{\mathrm{f}})\Big)^{1/2}. \tag{3.59}$$

Thus, the augmentation-based realization of `until` has a two-level structure. It first constructs a running `always`-type prefix condition for φ_1 and then performs an `eventually`-type witness search over the interval I for the conjunction of that prefix condition with φ_2. This mirrors the structure of the standard robustness semantics, but through smooth continuous-time integral surrogates.

Remark 3.8 (Regularization near $t = 0$). The normalization in (3.53)–(3.54) can be regularized in practice by replacing t with $t + \delta$ for a small $\delta > 0$. This avoids numerical issues near $t = 0$ while preserving the intended prefix-aggregation behavior. The induced perturbation can be made arbitrarily small as $\delta \to 0^+$.

3.4.3 Discussion

The augmentation-based realization is attractive because it requires only state augmentation and therefore integrates naturally into existing optimal-control pipelines. Its main advantage is implementation flexibility. Once the auxiliary dynamics are introduced, the temporal operators are handled through an augmented continuous-time system and standard boundary conditions, without requiring explicit storage or manipulation of dense-time samples.

At the same time, the realization is only approximately exact because of the regularization parameter ε required by the geometric-integral term. This approximation enters only through the shifted logarithmic branch, and its practical effect can often be made arbitrarily small. For critical `always`-type safety constraints such as obstacle avoidance, a small physical buffer in the predicate restores strict engineering safety, while asymmetric weighting can reduce the influence of the geometric-integral term relative to the negative-part integral. In the limiting case, for `always`-type constraints one may retain only the integral penalty on the negative part, thereby recovering the CTCS-style path-constraint construction of [104], at the cost of losing the positive-margin shaping effect contributed by the geometric term. For `eventually`-type specifications, if the target set has nonempty interior, the positive-part integral can dominate the shifted geometric term, making the conservatism induced by ε arbitrarily small. The same principle carries over to `until` through its inner `always`-type prefix condition and outer `eventually`-type witness condition.

The exact dense-time realization and the augmentation-based realization are therefore complementary. The dense-time realization is semantically exact on the dense numerical trajectory representation, whereas the augmentation-based realization provides a more transcription-friendly continuous-time alternative whose only source of approximation is the geometric-integral regularization. The next section extends both constructions to weighted formulations that allow relative priorities among subformulas and time instants to be encoded directly within the robustness model.

3.5 Weighted extensions

The constructions in Sections 3.3 and 3.4 used the baseline GMSR choice, namely unit weights and the arithmetic-mean branch with exponent $p = 1$. That baseline is often sufficient and is the form used in most of the examples in this dissertation. Nevertheless, many trajectory generation problems naturally call for nonuniform priorities. Some subformulas may be more important than others, some time instants may deserve stronger emphasis than others, and some temporal witnesses may be preferred over others. Weighted robustness semantics have been explored previously for standard robustness and AGM-type constructions [59, 198], and the general D-GMSR family already supports such extensions in a smooth and exact manner [286]. This section records the corresponding weighted extensions for the logical operators and for both CT-STL temporal realizations.

3.5.1 Weighted logical operators

Let $y \in \mathbb{R}^n$, let $c \in \mathbb{R}_{++}$, let $p \in \mathbb{Z}_{++}$, and let

$$w = (w_1, \ldots, w_n) \in \mathbb{Z}^n_{++}$$

be a strictly positive weight vector. Define the weighted generalized means

$$M^c_{0,w}(z) \coloneqq \left(c^{\mathbf{1}^\top w} + \prod_{i=1}^{n} z_i^{w_i} \right)^{1/(\mathbf{1}^\top w)}, \tag{3.60}$$

$$M^c_{p,w}(z) \coloneqq \left(c^p + \frac{1}{\mathbf{1}^\top w} \sum_{i=1}^{n} w_i z_i^p \right)^{1/p}, \tag{3.61}$$

for $z \in \mathbb{R}^n_{\geq 0}$. The weighted GMSR logical operators are then

$$^{\wedge}h^c_{p,w}(y) := \left(M^c_{0,w}\left(|y|^2_+\right)\right)^{1/2} - \left(M^c_{p,w}\left(|y|^2_-\right)\right)^{1/2}, \tag{3.62}$$

$$^{\vee}h^c_{p,w}(y) := -\,^{\wedge}h^c_{p,w}(-y). \tag{3.63}$$

The unit-weight, arithmetic-mean baseline of Section 3.3.1 is recovered by taking

$$w = \mathbf{1}, \qquad p = 1.$$

A further structural consequence of combining smoothness with exact sign preservation is that the key GMSR operators cannot be idempotent.

Remark 3.9 (Non-idempotence of the logical GMSR operators). Note that $^{\wedge}h^c_{p,w}$ and $^{\vee}h^c_{p,w}$ are non-idempotent since they are $\mathcal{C}^1$-smooth and preserve the soundness of the robustness measure [292, Proposition 1]. This implies that the values of $^{\wedge}h^c_{p,w}(y)$ and $^{\vee}h^c_{p,w}(y)$ at $y = a\mathbf{1}$ for $a \in \mathbb{R}$ are not equal to a. More precisely, for $a > 0$,

$$^{\wedge}h^c_{p,w}(a\mathbf{1}) = \left(c^{\mathbf{1}^\top w} + a^{2\mathbf{1}^\top w}\right)^{1/(2\mathbf{1}^\top w)} - c^{1/2} \neq a,$$

while for $a < 0$,

$$^{\wedge}h^c_{p,w}(a\mathbf{1}) = c^{1/2} - \left(c^p + |a|^{2p}\right)^{1/(2p)} \neq a.$$

The same conclusion holds for $^{\vee}h^c_{p,w}$ by the identity

$$^{\vee}h^c_{p,w}(y) = -^{\wedge}h^c_{p,w}(-y).$$

This non-idempotence is not a defect of the construction, but rather a structural price paid for exact sign preservation together with $\mathcal{C}^1$-smoothness.

The exponent p also controls the limiting relation between the weighted GMSR operators and the extremum operators that they replace.

Proposition 3.10 (Limit behavior)**.** *Let y and w be fixed. Then the following limits hold:*

- *If $\min(y) < 0$ and $c^{1/2} < |\min(y)|$, then*

$$\lim_{p\to\infty} {^{\wedge}h^c_{p,w}(y)} = \min(y) + c^{1/2}.$$

- *If* $\max(y) > 0$ *and* $c^{1/2} < \max(y)$*, then*

$$\lim_{p\to\infty} {}^{\vee}h^{c}_{p,w}(y) = \max(y) - c^{1/2}.$$

Proof. If $z \in \mathbb{R}^n_+$, then

$$M^0_{\infty,w}(z) = \max(z)$$

for any $w \in \mathbb{Z}^n_{++}$ [260]. If $c^{1/2} < \max(z)$, then

$$M^c_{\infty,w}(z) = \max(z).$$

Therefore, if $\min(y) < 0$ and $c^{1/2} < |\min(y)|$, then

$$\begin{aligned}
\lim_{p\to\infty} {}^{\wedge}h^{c}_{p,w}(y) &= c^{1/2} - \lim_{p\to\infty}\Big(M^c_{p,w}(|y|^2_-)\Big)^{1/2} \\
&= c^{1/2} - \Big(\max(|y|^2_-)\Big)^{1/2} \\
&= c^{1/2} - |\max(|y_i|_-)| \\
&= \min(y) + c^{1/2}.
\end{aligned}$$

Since

$${}^{\vee}h^{c}_{p,w}(y) := -{}^{\wedge}h^{c}_{p,w}(-y),$$

it follows that

$$\lim_{p\to\infty} {}^{\vee}h^{c}_{p,w}(y) = -\min(-y) - c^{1/2} = \max(y) - c^{1/2}$$

if $\max(y) > 0$ and $c^{1/2} < \max(y)$. ■

Remark 3.11 (Practical choice of p and w). The general weighted GMSR family allows both curvature and weighting to be tuned. In practice, the parameter p controls how extremum-like the power-mean branch becomes, while the weight vector w controls the relative influence of different subformulas or time samples [286]. Larger values of p make the power-mean term closer to an extremum-type aggregation, whereas $p = 1$ yields the arithmetic-mean choice used throughout the baseline constructions in this dissertation. In the numerical developments underlying this dissertation, the choice $p = 1$ is generally preferred.

Formula φ	**Weighted GMSR robustness** $\Gamma^{\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t)$
μ	$g(x(t))$
$\neg\varphi$	$-\Gamma^{\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t)$
$\varphi_1 \wedge \varphi_2$	${}^{\wedge}h^{c_1}_{p_1,w_1}\Big(\Gamma^{\varphi_1}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t), \Gamma^{\varphi_2}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t)\Big)$
$\varphi_1 \vee \varphi_2$	${}^{\vee}h^{c_1}_{p_1,w_1}\Big(\Gamma^{\varphi_1}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t), \Gamma^{\varphi_2}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t)\Big)$
$\varphi_1 \Rightarrow \varphi_2$	${}^{\vee}h^{c_1}_{p_1,w_1}\Big(-\Gamma^{\varphi_1}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t), \Gamma^{\varphi_2}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t)\Big)$

Table 3.7: Weighted GMSR parameterization of the logical operators of STL.

The weight vector $w \in \mathbb{Z}^n_{++}$ can be used to encode priorities. If the satisfaction of one subformula is more important than others, or if certain time instants should matter more strongly, the corresponding entries of w may be chosen larger. The partial derivatives of the weighted GMSR operators with respect to their active arguments are proportional to these weights, so larger weights directly increase the influence of the corresponding subformulas or time indices on both the robustness value and its gradient.

As in the unweighted case, the sign pattern is unchanged by positive weights:

$$ {}^{\wedge}h^{c}_{p,w}(y) \geq 0 \iff y_i \geq 0, \quad \forall i, \tag{3.64} $$

$$ {}^{\vee}h^{c}_{p,w}(y) \geq 0 \iff \exists\, i \text{ such that } y_i \geq 0. \tag{3.65} $$

Thus, weighting modifies the geometry of the robustness landscape without changing the underlying logical semantics.

Let $\Gamma^{\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t)$ denote the weighted GMSR robustness of formula φ, where the vectors $\boldsymbol{c}$, $\boldsymbol{p}$, and $\boldsymbol{w}$ collect the parameters of the weighted GMSR operators appearing in the parse tree of φ. The weighted logical recursion is summarized in Table 3.7.

Formula φ	Weighted dense-time GMSR robustness at $\bar{t}_\ell$
$\boldsymbol{F}_{[a,b]}\varphi$	${}^{\vee}h^{c_1}_{p_1,w^F_{\ell,[a,b]}}\Big([\,\tilde{\Gamma}^{\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(X,U;\bar{t}_m)\,]_{m\in\mathcal{I}_\ell[a,b]}\Big)$
$\boldsymbol{G}_{[a,b]}\varphi$	${}^{\wedge}h^{c_1}_{p_1,w^G_{\ell,[a,b]}}\Big([\,\tilde{\Gamma}^{\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(X,U;\bar{t}_m)\,]_{m\in\mathcal{I}_\ell[a,b]}\Big)$

Table 3.8: Weighted dense-time GMSR extension of `always` and `eventually`.

3.5.2 Weighted dense-time temporal extension

The dense-time realization of Section 3.3.2 is extended by assigning weights directly to the ordered robustness values over the dense-time interval of interest. Fix an evaluation time $\bar{t}_\ell$ and interval $[a,b]$, and recall the dense-time index set

$$\mathcal{I}_\ell[a,b] = \Big\{m : \bar{t}_m \in [\bar{t}_\ell + a, \bar{t}_\ell + b]\Big\}.$$

For each temporal operator occurrence, one may choose a strictly positive weight vector

$$w_{\ell,[a,b]} \in \mathbb{Z}_{++}^{|\mathcal{I}_\ell[a,b]|}.$$

These weights can be used, for example, to emphasize earlier or later times within the interval, to prioritize specific dense-time samples, or to encode nonuniform importance over the temporal window.

The weighted dense-time parameterizations of $\boldsymbol{F}$ and $\boldsymbol{G}$ are given in Table 3.8.

For the `until` operator, weighting enters both the outer witness search and the inner prefix aggregation. For each admissible witness index $m \in \mathcal{I}_\ell[a,b]$, define

$$z_m := {}^{\wedge}h^{c_2}_{p_2,w_2}\Big(\tilde{\Gamma}^{\varphi_2}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(X,U;\bar{t}_m),\ {}^{\wedge}h^{c_3}_{p_3,w^{\mathrm{pre}}_{\ell:m}}\Big([\,\tilde{\Gamma}^{\varphi_1}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(X,U;\bar{t}_q)\,]_{q=\ell}^{m}\Big)\Big), \tag{3.66}$$

where

$$w^{\mathrm{pre}}_{\ell:m} \in \mathbb{Z}_{++}^{m-\ell+1}$$

weights the prefix interval $[\bar{t}_\ell, \bar{t}_m]$. The weighted dense-time robustness of the `until` formula is then

$$\tilde{\Gamma}^{\varphi_1 \boldsymbol{U}_{[a,b]} \varphi_2}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(X, U; \bar{t}_\ell) = {}^{\vee}h^{c_1}_{p_1, w^U_{\ell,[a,b]}}\Big([\, z_m \,]_{m \in \mathcal{I}_\ell[a,b]} \Big), \tag{3.67}$$

where

$$w^U_{\ell,[a,b]} \in \mathbb{Z}^{|\mathcal{I}_\ell[a,b]|}_{++}$$

weights the candidate witness times.

As before, the unit-weight baseline is recovered by taking all temporal weight vectors equal to the all-ones vector and all temporal exponents equal to 1. Because all weights remain strictly positive, the sign-consistency and exactness properties of the dense-time realization are preserved. Weighting changes the relative influence of samples and subformulas, but not the underlying truth conditions.

3.5.3 Weighted augmentation-based extension

The augmentation-based realization can be weighted in continuous time by replacing discrete weight vectors over dense samples with a strictly positive weighting density over the time interval. Let $I = [a, b]$ and let

$$\omega : I \to \mathbb{R}_{++}$$

be a strictly positive weighting density. Define the normalization constant

$$\Omega_I := \int_I \omega(t)\, dt. \tag{3.68}$$

The weighted continuous-time aggregates are then

$$\mathcal{M}^c_{0,\omega,\varepsilon}(y; I) = c + \exp\bigg(\frac{1}{\Omega_I} \int_I \omega(t)\, \log\Big(|y(t)|^2_+ + \varepsilon \Big)\, dt \bigg), \tag{3.69}$$

$$\mathcal{M}^c_{p,\omega}(y; I) = \bigg(c^p + \frac{1}{\Omega_I} \int_I \omega(t)\, |y(t)|^{2p}_-\, dt \bigg)^{1/p}. \tag{3.70}$$

The corresponding weighted continuous-time conjunction/disjunction surrogates are

$${}^{\wedge}\mathcal{H}^c_{p,\omega,\varepsilon}(y; I) = \Big(\mathcal{M}^c_{0,\omega,\varepsilon}(y; I) \Big)^{1/2} - \Big(\mathcal{M}^c_{p,\omega}(y; I) \Big)^{1/2}, \tag{3.71}$$

$${}^{\vee}\mathcal{H}^c_{p,\omega,\varepsilon}(y; I) = -\, {}^{\wedge}\mathcal{H}^c_{p,\omega,\varepsilon}(-y; I). \tag{3.72}$$

The baseline augmentation-based realization is again recovered by taking $p = 1$ and constant weighting densities. Positive weighting densities preserve the same qualitative semantics as in the unweighted case, but allow selected portions of the interval to receive more influence in both the robustness value and its gradient.

These weighted aggregates induce weighted augmentation-based realizations of `always`, `eventually`, and `until` by replacing the corresponding unweighted temporal surrogates in Section 3.4 with their weighted counterparts. Since this extension is entirely parallel to the unweighted construction, the explicit operator-level formulas are omitted here for brevity.

3.6 Theoretical and numerical properties of the proposed robustness measures

This section summarizes the main properties of the robustness constructions developed in Sections 3.3–3.5. The emphasis is on smoothness, soundness, completeness, locality and masking, and the resulting gradient behavior that makes these constructions suitable for trajectory optimization. The logical GMSR constructions inherit their main properties from the discrete-time generalized mean-based smooth robustness framework [286], while the continuous-time dense-time and augmentation-based realizations extend those properties in different ways [284].

3.6.1 Smoothness and exactness

In Propositions 3.12 and 3.13, we assume

$$c_i \in \mathbb{R}_{++}, \qquad p_i \in \mathbb{Z}_{++}, \qquad w_i \in \mathbb{Z}_{++}^{n_i}, \qquad \forall i \in \mathbb{Z}_{++},$$

and do not restate this for brevity.

Proposition 3.12 ($\mathcal{C}^1$-smoothness of logical GMSR)**.** *Assume that all predicate functions are $\mathcal{C}^1$-smooth. Then, for every finite-horizon STL formula φ constructed recursively from predicates using negation, conjunction, disjunction, and implication, the corresponding weighted logical GMSR robustness*

$$\Gamma^{\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(\cdot, t)$$

is $\mathcal{C}^1$ with respect to its signal arguments.

Proof. It is enough to show that the key operators ${}^{\wedge}h^c_{p,w}$ and ${}^{\vee}h^c_{p,w}$ are $\mathcal{C}^1$ on $\mathbb{R}^n$, since the STL robustness is obtained by recursively composing these functions with $\mathcal{C}^1$ predicate functions.

First, note that for a scalar variable $r \in \mathbb{R}$,

$$|r|_+^2 = \max(r,0)^2, \qquad |r|_-^2 = \min(r,0)^2,$$

are both $\mathcal{C}^1$ functions of r. Indeed,

$$\frac{d}{dr}|r|_+^2 = 2|r|_+, \qquad \frac{d}{dr}|r|_-^2 = 2|r|_-,$$

and both derivatives are continuous. Hence the vector-valued maps

$$y \mapsto |y|_+^2, \qquad y \mapsto |y|_-^2$$

are $\mathcal{C}^1$ on $\mathbb{R}^n$.

Next, consider the weighted generalized means

$$M^c_{0,w}(z) = \left(c^{\mathbf{1}^\top w} + \prod_{i=1}^n z_i^{w_i}\right)^{1/(\mathbf{1}^\top w)},$$
$$M^c_{p,w}(z) = \left(c^p + \frac{1}{\mathbf{1}^\top w}\sum_{i=1}^n w_i z_i^p\right)^{1/p},$$

defined for $z \in \mathbb{R}^n_{\geq 0}$. Because each $w_i \in \mathbb{Z}_{++}$ and $p \in \mathbb{Z}_{++}$, the maps

$$z \mapsto \prod_{i=1}^n z_i^{w_i}, \qquad z \mapsto \sum_{i=1}^n w_i z_i^p$$

are polynomial-type smooth functions on $\mathbb{R}^n_{\geq 0}$. Since $c > 0$, the quantities

$$c^{\mathbf{1}^\top w} + \prod_{i=1}^n z_i^{w_i} \quad \text{and} \quad c^p + \frac{1}{\mathbf{1}^\top w}\sum_{i=1}^n w_i z_i^p$$

are strictly positive for every $z \in \mathbb{R}^n_{\geq 0}$. Therefore, the outer power maps

$$r \mapsto r^{1/(\mathbf{1}^\top w)}, \qquad r \mapsto r^{1/p}$$

are $\mathcal{C}^\infty$ on the relevant positive domains, and hence both

$$M^c_{0,w}, \qquad M^c_{p,w}$$

are $\mathcal{C}^\infty$ on $\mathbb{R}^n_{\geq 0}$.

Composing these smooth maps with $|y|^2_+$ and $|y|^2_-$ shows that

$$y \mapsto M^c_{0,w}(|y|^2_+), \qquad y \mapsto M^c_{p,w}(|y|^2_-)$$

are $\mathcal{C}^1$ on $\mathbb{R}^n$. Since $c > 0$, both quantities are bounded below by c, so the square-root map is smooth on their ranges. It follows that

$$^\wedge h^c_{p,w}(y) = \left(M^c_{0,w}(|y|^2_+)\right)^{1/2} - \left(M^c_{p,w}(|y|^2_-)\right)^{1/2}$$

is $\mathcal{C}^1$ on $\mathbb{R}^n$. Finally,

$$^\vee h^c_{p,w}(y) = -^\wedge h^c_{p,w}(-y)$$

is also $\mathcal{C}^1$.

Since the predicate functions are $\mathcal{C}^1$ by assumption, and every logical robustness formula is obtained by finitely many compositions of $\mathcal{C}^1$ predicate functions with the $\mathcal{C}^1$ key operators $^\wedge h^c_{p,w}$ and $^\vee h^c_{p,w}$, the resulting robustness

$$\Gamma^\varphi_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(\cdot, t)$$

is $\mathcal{C}^1$. The baseline case is the special case $p_i = 1$ and $w_i = \mathbf{1}$. ■

Proposition 3.13 (Soundness and completeness of logical GMSR)**.** *For every finite-horizon STL formula φ constructed recursively from predicates using negation, conjunction, disjunction, and implication,*

$$(x,t) \models \varphi \iff \Gamma^\varphi_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t) \geq 0. \tag{3.73}$$

Thus, the logical GMSR constructions are sound and complete, and hence exact, with respect to the standard STL semantics.

Proof. We first prove the sign-exactness of the conjunction operator ${}^{\wedge}h^c_{p,w}$.

Let $y \in \mathbb{R}^n$. From the definitions,

$$M^c_{0,w}(|y|^2_+) \geq c, \qquad M^c_{p,w}(|y|^2_-) \geq c.$$

Moreover,

$$M^c_{0,w}(|y|^2_+) > c \iff \prod_{i=1}^n |y_i|_+^{2w_i} > 0 \iff |y_i|_+ > 0,\ \forall i \iff y_i > 0,\ \forall i.$$

Likewise,

$$M^c_{p,w}(|y|^2_-) > c \iff \sum_{i=1}^n w_i |y_i|_-^{2p} > 0 \iff \exists\, i \text{ such that } |y_i|_- > 0 \iff \exists\, i \text{ such that } y_i < 0.$$

Since the positive and negative parts cannot be simultaneously active for the same coordinate, the following implications hold:

$$M^c_{0,w}(|y|^2_+) > c \implies M^c_{p,w}(|y|^2_-) = c,$$

and

$$M^c_{p,w}(|y|^2_-) > c \implies M^c_{0,w}(|y|^2_+) = c.$$

We now characterize the sign of ${}^{\wedge}h^c_{p,w}(y)$.

If some entry of y is negative, then $M^c_{p,w}(|y|^2_-) > c$ and $M^c_{0,w}(|y|^2_+) = c$, so

$${}^{\wedge}h^c_{p,w}(y) = c^{1/2} - \Big(M^c_{p,w}(|y|^2_-)\Big)^{1/2} < 0.$$

Hence

$$\min(y) < 0 \implies {}^{\wedge}h^c_{p,w}(y) < 0.$$

If all entries of y are nonnegative, then $|y|^2_- = 0$, so $M^c_{p,w}(|y|^2_-) = c$, and

$${}^{\wedge}h^c_{p,w}(y) = \Big(M^c_{0,w}(|y|^2_+)\Big)^{1/2} - c^{1/2} \geq 0.$$

Moreover, the value is strictly positive if and only if all entries are strictly positive. Thus

$$\min(y) \geq 0 \iff {}^{\wedge}h^c_{p,w}(y) \geq 0.$$

Therefore, ${}^{\wedge}h^{c}_{p,w}$ has exactly the same sign pattern as $\min(y)$.

Since

$$ {}^{\vee}h^{c}_{p,w}(y) = -{}^{\wedge}h^{c}_{p,w}(-y), $$

it follows immediately that ${}^{\vee}h^{c}_{p,w}$ has exactly the same sign pattern as $\max(y)$:

$$ {}^{\vee}h^{c}_{p,w}(y) \geq 0 \iff \max(y) \geq 0. $$

We now prove (3.73) by structural induction on the formula φ. The atomic predicate case follows directly from the definition

$$ (x,t) \models \mu \iff g(x(t)) \geq 0 \iff \Gamma^{\mu}(x,t) \geq 0. $$

The claim for negation, conjunction, disjunction, and implication then follows immediately by structural induction on the parse tree of φ, since the recursive GMSR construction mirrors the recursive Boolean semantics exactly, with ${}^{\wedge}h^{c}_{p,w}$ and ${}^{\vee}h^{c}_{p,w}$ replacing the sign behavior of min and max. Therefore,

$$ (x,t) \models \varphi \iff \Gamma^{\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,t) \geq 0 $$

for every finite-horizon STL formula φ. ■

The exact dense-time realization introduced in Section 3.3.2 inherits the key properties of the logical GMSR operators directly. Indeed, the temporal constructions for `always`, `eventually`, and `until` are obtained by applying the same smooth and sign-exact operators ${}^{\wedge}h^{c}_{p,w}$ and ${}^{\vee}h^{c}_{p,w}$ to vectors of robustness values indexed over dense-time. Consequently, on the dense numerical trajectory representation, the resulting CT-STL robustness is $\mathcal{C}^1$ with respect to the optimization variables, and sound and complete up to the numerical accuracy of the underlying integration scheme.

By contrast, the augmentation-based realization introduces a regularization parameter $\varepsilon > 0$ in the geometric-integral term.

Proposition 3.14 (Augmentation-based smoothness and approximate exactness)**.** *Assume that all shift parameters satisfy $c_i \in \mathbb{R}_{++}$, all exponents satisfy $p_i \in \mathbb{Z}_{++}$, all discrete weight* *vectors satisfy* $w_i \in \mathbb{Z}_{++}^{n_i}$*, all predicate functions are* $\mathcal{C}^1$*-smooth, the underlying continuous-time dynamics are* $\mathcal{C}^1$ *in state and input, and every weighting density* $\omega : I \to \mathbb{R}_{++}$ *is fixed and continuous on its compact interval* I*. Then the augmentation-based temporal operators, both in their weighted and unweighted forms, are* $\mathcal{C}^1$ *with respect to the signal and decision variables. However, for fixed* $\varepsilon > 0$*, the augmentation-based realization is only approximately exact. Its deviation from exactness is solely due to the* ε*-regularization required in the geometric-integral term.*

Proof. Consider the scalar functions

$$\alpha_\varepsilon(r) \coloneqq \log(|r|_+^2 + \varepsilon), \qquad \beta_p(r) \coloneqq |r|_-^{2p}.$$

Since $r \mapsto |r|_+^2$ and $r \mapsto |r|_-^2$ are $\mathcal{C}^1$, and $|r|_+^2 + \varepsilon \geq \varepsilon > 0$, both α_ε and β_p are $\mathcal{C}^1$ on $\mathbb{R}$.

For a fixed positive weighting density ω on a compact interval I, multiplication by $\omega(t)$ and normalization by

$$\Omega_I = \int_I \omega(t)\, dt > 0$$

preserve smoothness. Hence the weighted continuous-time aggregates

$$\mathcal{M}_{0,\omega,\varepsilon}^c(y; I) = c + \exp\Big(\frac{1}{\Omega_I} \int_I \omega(t) \alpha_\varepsilon(y(t))\, dt\Big)$$

and

$$\mathcal{M}_{p,\omega}^c(y; I) = \Big(c^p + \frac{1}{\Omega_I} \int_I \omega(t) \beta_p(y(t))\, dt\Big)^{1/p}$$

are $\mathcal{C}^1$ functions of the signal y. The corresponding surrogates

$$^\wedge\mathcal{H}_{p,\omega,\varepsilon}^c, \qquad ^\vee\mathcal{H}_{p,\omega,\varepsilon}^c$$

are therefore $\mathcal{C}^1$ as well. The unweighted case is recovered by taking constant densities and $p = 1$.

The logical robustness functions $\Gamma^\varphi_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x, t)$ are $\mathcal{C}^1$ by Proposition 3.12. Therefore, all right-hand sides appearing in the augmented dynamics for $\boldsymbol{G}$, $\boldsymbol{F}$, and $\boldsymbol{U}$, in both the weighted and unweighted constructions, are compositions of $\mathcal{C}^1$ functions. Hence the augmented vector

Construction	$\mathcal{C}^1$-smoothness	Soundness	Completeness	Monotonicity	Locality & Masking
Logical GMSR layer	✓	✓	✓	✓	✓
Dense-time GMSR realization	✓	✓	✓	✓	✓
Augmentation-based GMSR realization	✓	○	○	✓	✓

Table 3.9: Summary of the main properties of the proposed GMSR constructions. Here, ○ indicates approximate soundness and completeness due only to the geometric-integral regularization parameter $\varepsilon > 0$.

field is $\mathcal{C}^1$ in its state and input arguments. By the standard smooth dependence of ODE solutions on initial conditions and parameters, the corresponding auxiliary states and their terminal values depend $\mathcal{C}^1$-smoothly on the decision variables. Since the augmentation-based robustness values are obtained from those terminal auxiliary states by addition and square root, they are themselves $\mathcal{C}^1$.

It remains to explain approximate exactness. The only difference between the ideal exact continuous-time construction and the implemented one is the replacement of $|y(t)|_+^2$ by $|y(t)|_+^2 + \varepsilon$ inside the logarithm of the geometric-integral branch. If $|y(t)|_+^2$ vanishes on part of the interval, the unregularized logarithm is undefined, whereas the regularized expression remains finite. Thus, for every fixed $\varepsilon > 0$, the geometric-integral branch is a regularized surrogate of the ideal exact construction. The integral branch is unchanged, and the weighting densities only reweight the corresponding integrands; they do not alter the underlying sign logic and do not introduce any additional approximation. Accordingly, the only source of deviation from exact sign-exactness is the regularization parameter ε, and this deviation vanishes as the regularization is reduced. ■

The main distinctions among the constructions are summarized in Table 3.9.

The next subsection develops the gradient-based interpretation underlying the monotonicity and locality/masking behavior summarized in Table 3.9.

3.6.2 Gradient behavior, locality, and masking

The main numerical advantage of GMSR is not only that it is smooth and exact, but also that it distributes gradient information across multiple relevant subformulas and time instants. This subsection makes that mechanism explicit. We first contrast the gradient structure of standard smooth min/max-based robustness with that of GMSR, then verify the resulting operator behavior numerically, illustrate its impact on a trajectory optimization problem, and finally explain how the same mechanism extends to the continuous-time CT-STL constructions.

Gradient behavior of standard smooth robustness. Let $\tilde{\rho}$ denote a smooth min/max-based relaxation of discrete-time space robustness, such as the D-SSR construction in [123]. In such relaxations, the values of multiple predicate functions across subformulas and time instants are still composed through smooth approximations of the min and max operators. As a result, the robustness value is typically dominated by one extremizing predicate at one extremizing time instant, which is precisely the source of the locality and masking pathologies [197, 286]. In other words, although the robustness function is smooth, its effective gradient information remains highly localized.

To make this precise, let $f_i : \mathbb{R}^m \to \mathbb{R}$ for all $i \in \{1, \ldots, n\}$, let

$$y = (f_1(x), f_2(x), \ldots, f_n(x)),$$

and define the index sets

$$S_{\min} := \{\, i : f_i(x) \le f_j(x),\ \forall j \in \{1, \ldots, n\} \,\}, \qquad S_{\max} := \{\, i : f_i(x) \ge f_j(x),\ \forall j \in \{1, \ldots, n\} \,\}.$$

For sufficiently large sharpness parameter κ, the gradients of the smooth min/max approximations satisfy

$$\nabla_x \widetilde{\min}_\kappa(y) = \sum_{i=1}^{n} a_i \nabla_x f_i(x), \tag{3.74}$$

$$\nabla_x \widetilde{\max}_\kappa(y) = \sum_{i=1}^{n} b_i \nabla_x f_i(x), \tag{3.75}$$

where

$$a_i \approx \begin{cases} \dfrac{1}{\mathrm{card}(S_{\min})}, & i \in S_{\min}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.76}$$

$$b_i \approx \begin{cases} \dfrac{1}{\mathrm{card}(S_{\max})}, & i \in S_{\max}, \\ 0, & \text{otherwise}. \end{cases} \tag{3.77}$$

Hence, when D-SSR is used inside a gradient-based iterative optimizer, the update direction is determined almost entirely by the gradient of a single predicate at a single time instant. All other predicate values and time-indexed evaluations are nearly ignored. This is the mechanism underlying locality and masking [123, 286].

Example 3.15 (Illustration of D-SSR operator behavior)**.** *Consider the operator-level maximization of*

$$\tilde{\rho}^{\varphi_1 \wedge \cdots \wedge \varphi_5}(x,k) \qquad \textit{and} \qquad \tilde{\rho}^{\varphi_1 \vee \cdots \vee \varphi_5}(x,k),$$

where $\tilde{\rho}^{\varphi_i}(x,k) = f_i(x_k) \in \mathbb{R}$ *for* $i \in \{1,\ldots,5\}$ *and the smooth min/max approximation uses* $\kappa = 25$*. Equivalently,*

$$\tilde{\rho}^{\varphi_1 \wedge \cdots \wedge \varphi_5}(x,k) = \widetilde{\min}_\kappa(y), \qquad \tilde{\rho}^{\varphi_1 \vee \cdots \vee \varphi_5}(x,k) = \widetilde{\max}_\kappa(y),$$

with

$$y = (f_1(x_k), f_2(x_k), \ldots, f_5(x_k)).$$

The same operator-level interpretation also applies to

$$\tilde{\rho}^{\boldsymbol{G}_{[1:5]}\varphi}(x,0) = \widetilde{\min}_\kappa(y), \qquad \tilde{\rho}^{\boldsymbol{F}_{[1:5]}\varphi}(x,0) = \widetilde{\max}_\kappa(y),$$

when $\tilde{\rho}^{\varphi}(x,k) = f(x_k)$ *and*

$$y = (f(x_1), f(x_2), \ldots, f(x_5)).$$

Figure 3.3 illustrates the resulting numerical behavior. During the maximization of the conjunction-type expression, only the predicate attaining the minimum value changes *significantly at each iteration, and the active minimum shifts only when that predicate has been sufficiently improved. During the maximization of the disjunction-type expression, only the predicate attaining the maximum value changes significantly, so the same witness typically remains active throughout the iterations. These plots numerically verify the gradient concentration predicted by* (3.74)–(3.77)*.*

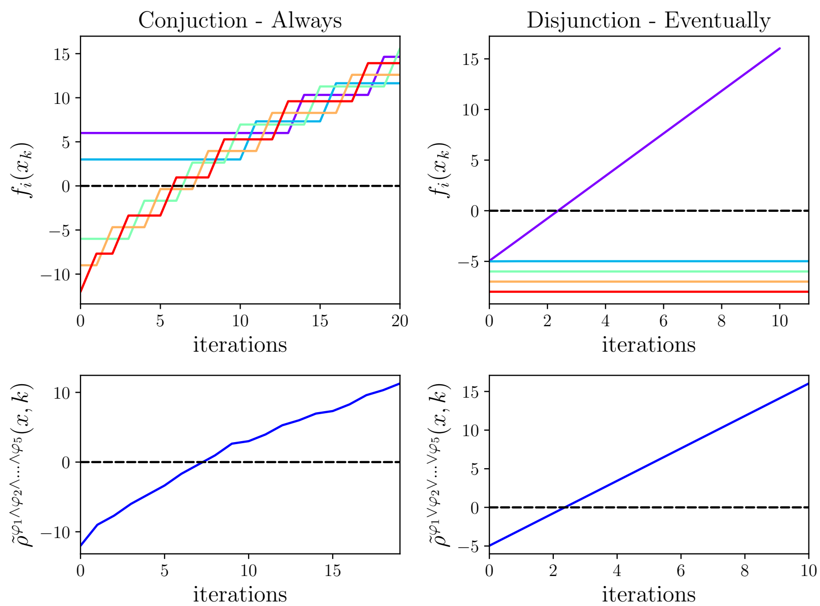


Figure 3.3: Changes in the predicate values and robustness values while maximizing conjunction/always and disjunction/eventually operators using D-SSR. The color gradient from purple to red corresponds to predicate values from $f_1(x_k)$ to $f_5(x_k)$, or equivalently from $f(x_1)$ to $f(x_5)$ in the temporal interpretation.

Gradient behavior of D-GMSR. The key functions of D-GMSR are built from generalized means in a way that avoids the gradient collapse of (3.74)–(3.77). Let again $f_i : \mathbb{R}^m \to \mathbb{R}$ for all $i \in \{1, \ldots, n\}$, let

$$y = (f_1(x), f_2(x), \ldots, f_n(x)),$$

and consider the weighted GMSR key operators ${}^{\wedge}h^{c}_{p,w}$ and ${}^{\vee}h^{c}_{p,w}$. Then

$$\nabla_x {}^{\wedge}h^{c}_{p,w}(y) = c_0^{+}\nabla_x M^{c}_{0,w}(|y|_{+}^{2}) - c_p^{-}\nabla_x M^{c}_{p,w}(|y|_{-}^{2}), \tag{3.78}$$

$$\nabla_x {}^{\vee}h^{c}_{p,w}(y) = c_p^{+}\nabla_x M^{c}_{p,w}(|y|_{+}^{2}) - c_0^{-}\nabla_x M^{c}_{0,w}(|y|_{-}^{2}), \tag{3.79}$$

where

$$\nabla_x M^{c}_{0,w}(|y|_{\square}^{2}) = c_{0,m}^{\square}\sum_{i=1}^{n} w_i \frac{1}{|y_i|_{\square}}\nabla_x f_i(x), \tag{3.80}$$

$$\nabla_x M^{c}_{p,w}(|y|_{\square}^{2}) = c_{p,m}^{\square}\sum_{i=1}^{n} w_i |y_i|_{\square}^{2p-1}\nabla_x f_i(x), \tag{3.81}$$

with $\square \in \{+,-\}$ and

$$c_0^{\square} = \frac{1}{2}\Big(M^{c}_{0,w}(|y|_{\square}^{2})\Big)^{-1/2}, \tag{3.82a}$$

$$c_p^{\square} = \frac{1}{2}\Big(M^{c}_{p,w}(|y|_{\square}^{2})\Big)^{-1/2}, \tag{3.82b}$$

$$c_{0,m}^{\square} = \frac{2\prod_{j=1}^{n}|y_j|_{\square}^{2w_j}}{\mathbf{1}^{\top}w\Big(M^{c}_{0,w}(|y|_{\square}^{2})\Big)^{\mathbf{1}^{\top}w-1}}, \tag{3.82c}$$

$$c_{p,m}^{\square} = \frac{2}{\mathbf{1}^{\top}w\Big(M^{c}_{p,w}(|y|_{\square}^{2})\Big)^{p-1}}. \tag{3.82d}$$

Substituting (3.80)–(3.81) into (3.78)–(3.79) yields the weighted gradient-sum representation

$$\nabla_x {}^{\wedge}h^{c}_{p,w}(y) = \sum_{i=1}^{n} w_i a_i \nabla_x f_i(x), \tag{3.83a}$$

$$\nabla_x {}^{\vee}h^{c}_{p,w}(y) = \sum_{i=1}^{n} w_i b_i \nabla_x f_i(x), \tag{3.83b}$$

where

$$a_i = \begin{cases} c_0^{+}c_{0,m}^{+}\dfrac{1}{|y_i|_{+}}, & {}^{\wedge}h^{c}_{p,w}(y) > 0, \\[2ex] -\,c_p^{-}c_{p,m}^{-}|y_i|_{-}^{2p-1}, & \text{otherwise}, \end{cases} \tag{3.84a}$$

$$b_i = \begin{cases} -\,c_0^{-}c_{0,m}^{-}\dfrac{1}{|y_i|_{-}}, & {}^{\vee}h^{c}_{p,w}(y) < 0, \\[2ex] c_p^{+}c_{p,m}^{+}|y_i|_{+}^{2p-1}, & \text{otherwise}. \end{cases} \tag{3.84b}$$

Thus, unlike D-SSR, the gradient of D-GMSR is a weighted sum of the gradients of multiple active predicates. The relative weights of these active contributions are determined by the sign pattern of the arguments and by the inverse- or direct-proportional scaling in (3.84a)–(3.84b).

Example 3.16 (Illustration of D-GMSR operator behavior)**.** *Consider the same operator-level setup as in Example 3.15, but now maximize*

$$\Gamma^{\varphi_1\wedge\cdots\wedge\varphi_5}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,k) \qquad \textit{and} \qquad \Gamma^{\varphi_1\vee\cdots\vee\varphi_5}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,k),$$

with parameters

$$c = 10^{-8}, \qquad p = 1, \qquad w = \mathbf{1}.$$

Equivalently,

$$\Gamma^{\varphi_1\wedge\cdots\wedge\varphi_5}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,k) = {}^{\wedge}h^c_{p,w}(y), \qquad \Gamma^{\varphi_1\vee\cdots\vee\varphi_5}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,k) = {}^{\vee}h^c_{p,w}(y),$$

where

$$y = (f_1(x_k), f_2(x_k), \ldots, f_5(x_k)).$$

The same interpretation applies to

$$\Gamma^{\boldsymbol{G}_{[1:5]}\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,0) = {}^{\wedge}h^c_{p,w}(y), \qquad \Gamma^{\boldsymbol{F}_{[1:5]}\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,0) = {}^{\vee}h^c_{p,w}(y),$$

when

$$y = (f(x_1), f(x_2), \ldots, f(x_5)).$$

Figure 3.4 numerically verifies the analytic derivative statements above. For conjunction-type aggregation, when positive and negative values coexist, only the negative-valued arguments are pushed upward until they meet at 0*. When all arguments are positive, they all continue to increase, with smaller margins increasing more strongly. For disjunction-type aggregation, if all values are negative, all of them move upward, with the near-feasible ones moving most strongly; if positive and negative values coexist, only the positive-valued arguments continue to increase, and the strongest witness grows fastest [286].*

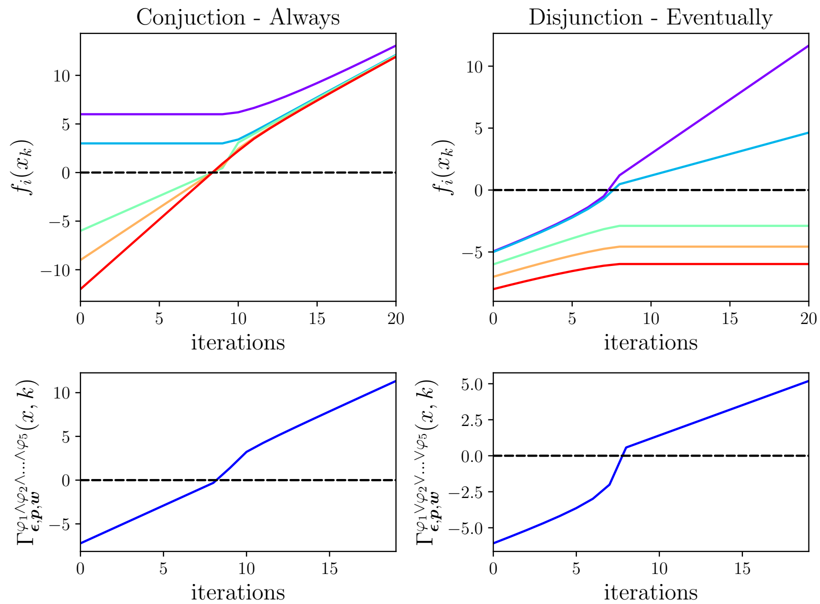


Figure 3.4: Changes in the predicate values and robustness values while maximizing conjunction/always and disjunction/eventually operators using D-GMSR. The color gradient from purple to red corresponds to predicate values from $f_1(x_k)$ to $f_5(x_k)$, or equivalently from $f(x_1)$ to $f(x_5)$ in the temporal interpretation.

Examples 3.15 and 3.16 verify the gradient mechanisms of the D-SSR and D-GMSR operators. The behavior observed in Figure 3.4 is a direct consequence of the monotonicity and locality/masking properties of the GMSR key operators, which we state next.

Monotonicity, locality, and masking. It follows directly from the derivative expressions above that the logical GMSR key operators are monotone.

Proposition 3.17 (Monotonicity)**.** *The functions* ${}^{\wedge}h^c_{p,w} : \mathbb{R}^n \to \mathbb{R}$ *and* ${}^{\vee}h^c_{p,w} : \mathbb{R}^n \to \mathbb{R}$ *are monotone. More specifically, each function is nondecreasing in each of its arguments.*

Proof. We prove that the partial derivatives of ${}^{\wedge}h^c_{p,w}$ and ${}^{\vee}h^c_{p,w}$ with respect to each of their variables are nonnegative. Equation (3.83) implies that

$$\partial_{y_i} {}^{\wedge}h^c_{p,w}(y) = w_i a_i, \qquad \partial_{y_i} {}^{\vee}h^c_{p,w}(y) = w_i b_i.$$

The nonnegativity of the constants appearing in the derivative construction preceding (3.83) implies that the coefficients

$$a_i, \qquad b_i$$

in (3.84) are nonnegative for all $i \in \{1, \dots, n\}$. Therefore,

$$w_i a_i \geq 0, \qquad w_i b_i \geq 0,$$

for all $w \in \mathbb{Z}^n_{++}$, which proves that both ${}^{\wedge}h^c_{p,w}$ and ${}^{\vee}h^c_{p,w}$ are nondecreasing in each argument.

■

Proposition 3.18 (Locality & masking)**.** *Consider the logical GMSR operators* ${}^{\wedge}h^c_{p,w}$ *and* ${}^{\vee}h^c_{p,w}$*. Whenever one of these operators has a negative value, its derivative with respect to every argument that contributes to that negative value is positive. Furthermore, whenever the derivative with respect to an argument is nonzero, its magnitude is either directly or inversely proportional to the magnitude of that argument, up to the corresponding positive weight.*

Proof. We prove the result using the derivative formulas derived above for the logical GMSR operators.

If ${}^{\wedge}h^c_{p,w}(y) < 0$, then there exists at least one index $i \in \{1, \dots, n\}$ such that $y_i < 0$, and the negative branch of ${}^{\wedge}h^c_{p,w}$ is active. Therefore, the corresponding coefficient multiplying the gradient contribution of that variable is positive, which implies that the derivative of ${}^{\wedge}h^c_{p,w}$ with respect to every active negative-valued variable is positive.

Furthermore:

1. If ${}^{\wedge}h^c_{p,w}(y) > 0$, then the positive branch is active and

$$\frac{\partial_{y_i} {}^{\wedge}h^c_{p,w}(y)}{\partial_{y_j} {}^{\wedge}h^c_{p,w}(y)} \propto \frac{w_i y_j}{w_j y_i}, \qquad \forall\, i, j \in \{1, \dots, n\}.$$

   Thus, for equal weights, smaller positive margins receive larger gradient emphasis.

2. If ${}^{\wedge}h^{c}_{p,w}(y) < 0$, then

$$\frac{\partial_{y_i} {}^{\wedge}h^{c}_{p,w}(y)}{\partial_{y_j} {}^{\wedge}h^{c}_{p,w}(y)} \propto \frac{w_i |y_i|^{2p-1}}{w_j |y_j|^{2p-1}}, \qquad \forall\, i, j \in \{k : y_k < 0\},$$

and

$$\partial_{y_i} {}^{\wedge}h^{c}_{p,w}(y) = 0 \qquad \forall\, i \in \{k : y_k \geq 0\}.$$

Similarly, if ${}^{\vee}h^{c}_{p,w}(y) < 0$, then all entries of y are negative and the corresponding active-branch coefficients are positive, so the derivatives with respect to all contributing variables are positive.

Furthermore:

3. If ${}^{\vee}h^{c}_{p,w}(y) < 0$, then

$$\frac{\partial_{y_i} {}^{\vee}h^{c}_{p,w}(y)}{\partial_{y_j} {}^{\vee}h^{c}_{p,w}(y)} \propto \frac{w_i |y_j|}{w_j |y_i|}, \qquad \forall\, i, j \in \{1, \ldots, n\}.$$

Thus, for equal weights, among negative entries, those closest to zero receive larger gradient emphasis.

4. If ${}^{\vee}h^{c}_{p,w}(y) > 0$, then

$$\frac{\partial_{y_i} {}^{\vee}h^{c}_{p,w}(y)}{\partial_{y_j} {}^{\vee}h^{c}_{p,w}(y)} \propto \frac{w_i y_i^{2p-1}}{w_j y_j^{2p-1}}, \qquad \forall\, i, j \in \{k : y_k > 0\},$$

and

$$\partial_{y_i} {}^{\vee}h^{c}_{p,w}(y) = 0 \qquad \forall\, i \in \{k : y_k \leq 0\}.$$

■

These properties precisely explain why GMSR avoids the extreme localization and masking behavior of standard min / max-based robustness and of their sharp smooth approximations. The practical implication for trajectory optimization is that D-SSR may focus the optimizer on a single extremizing node or time instant, while D-GMSR continues to propagate gradient information to multiple relevant

Example 3.19 (Trajectory-level illustration of locality and masking)**.** *Consider the trajectory optimization problem*

$$
\begin{aligned}
\underset{x_k,u_k}{\textit{maximize}} \quad & \tilde{\rho}^{\boldsymbol{F}_{[1:K]}\varphi}(x,0) \quad \textit{or} \quad \Gamma^{\boldsymbol{F}_{[1:K]}\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,0) \\
\textit{subject to} \quad & x_{k+1} = x_k + u_k, \qquad k \in \{1,\dots,K-1\}, \\
& x_1 = (0,0), \\
& x_K = (8,0), \\
& \|u_k\| \le 1.5, \qquad k \in \{1,\dots,K\},
\end{aligned}
\tag{3.85}
$$

with $K = 9$, where the goal is to **`eventually`** *reach a circular target set centered at p_w with radius r_w. The predicate is*

$$f(x) \coloneqq r_w - \|x - p_w\|, \qquad \varphi \coloneqq (f(x) \ge 0).$$

The D-SSR case uses $\kappa = 25$, whereas the D-GMSR case uses

$$c = 10^{-8}, \qquad p = 1, \qquad w = \mathbf{1}.$$

The initial and converged trajectories, together with the corresponding predicate values and robustness values over the iterations, are shown in Figure 3.5.

For the chosen initial trajectory, the second node is the closest one to the target set. Consequently, in the D-SSR case,

$$\nabla_{x_k}\tilde{\rho}^{\boldsymbol{F}_{[1:K]}\varphi}(x,0) \approx 0, \qquad k \in \{1,\dots,K\} \setminus \{2\},$$

so the optimizer effectively acts only on the second node. As illustrated by the red trajectory in Figure 3.5, the control bound prevents that node from entering the target set, even though other nodes could do so. The optimization therefore converges to a trajectory that still violates the STL specification.

By contrast, in the D-GMSR case,

$$\nabla_{x_k}\Gamma^{\boldsymbol{F}_{[1:K]}\varphi}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(x,0) = a_k\, \nabla_{x_k} f(x_k),$$

with

$$a_2 > a_k \gg 0, \qquad k \in \{1, \ldots, K\} \setminus \{2\}.$$

Thus, the optimizer still prioritizes the second node, but it also continues to move the other nodes toward the target set. As illustrated by the blue trajectory in Figure 3.5, one of the other nodes eventually reaches the target region, and the specification becomes satisfied. This example makes the trajectory-level consequences of locality and masking explicit [286].

Continuous-time extensions. The same gradient logic carries over directly to the continuous-time constructions, since both realizations are built from the same weighted GMSR operators applied either to dense-time samples or to continuous-time aggregates.

For the exact dense-time realization, the same coefficient structure appears after replacing the vector of logical subformula values by a vector of dense-time robustness values. In particular, for conjunction-type aggregation, one has

$$\left|\frac{\partial \,{}^{\wedge}h^c_{p,w}}{\partial y_i}\right| \propto \frac{w_i}{y_i}, \qquad y_i > 0, \tag{3.86}$$

$$\left|\frac{\partial \,{}^{\wedge}h^c_{p,w}}{\partial y_i}\right| \propto w_i |y_i|^{2p-1}, \qquad y_i < 0, \tag{3.87}$$

whereas for disjunction-type aggregation, one has

$$\left|\frac{\partial \,{}^{\vee}h^c_{p,w}}{\partial y_i}\right| \propto \frac{w_i}{|y_i|}, \qquad y_i < 0, \tag{3.88}$$

$$\left|\frac{\partial \,{}^{\vee}h^c_{p,w}}{\partial y_i}\right| \propto w_i |y_i|^{2p-1}, \qquad y_i > 0. \tag{3.89}$$

Thus, `always` behaves as a smooth worst-case aggregation over dense-time, `eventually` behaves as a smooth best-candidate aggregation, and `until` combines the two through its outer witness search and inner prefix aggregation [284, 285].

For the augmentation-based realization, the same interpretation persists in continuous form. If $y(t)$ is the active signal and $\omega(t)$ is the temporal weighting density, then the conjunction-type surrogate ${}^{\wedge}\mathcal{H}^{c}_{p,\omega,\varepsilon}$ satisfies

$$\frac{\delta\, {}^{\wedge}\mathcal{H}^{c}_{p,\omega,\varepsilon}}{\delta y(t)} \propto \omega(t)\frac{y(t)}{y(t)^2+\varepsilon}, \qquad y(t) > 0 \quad \text{geometric-integral branch,} \tag{3.90}$$

$$\frac{\delta\, {}^{\wedge}\mathcal{H}^{c}_{p,\omega,\varepsilon}}{\delta y(t)} \propto \omega(t)|y(t)|^{2p-1}, \qquad y(t) < 0 \quad \text{integral branch.} \tag{3.91}$$

and the disjunction-type surrogate has the dual behavior. Therefore, the augmentation-based realization also distributes gradient information across an interval rather than collapsing it to a single critical time. The only difference from the exact dense-time realization is that the inverse-proportional geometric branch is recovered only approximately because of the unavoidable ε-regularization [284, 285].

Thus, the continuous-time extensions do not introduce a fundamentally different gradient mechanism; rather, they lift the same GMSR structure from discrete collections of predicate values to dense-time samples and continuous-time aggregates.

Taken together, the preceding analysis explains the numerical behavior observed in the DT-STL and CT-STL studies. D-SSR and related smooth min/max relaxations inherit gradient concentration from the underlying extremum operators, which can lead to locality and masking and can prevent the optimizer from finding a satisfying trajectory even when one exists. GMSR replaces this behavior with a distributed gradient structure that remains smooth, exact, and monotone while emphasizing the most relevant active margins. In the dense-time CT-STL realization, the same mechanism is distributed across dense-time samples; in the augmentation-based realization, it is distributed across intervals through integral and geometric-integral surrogates. This yields a substantially more favorable optimization landscape for trajectory generation with temporal logic mission specifications.

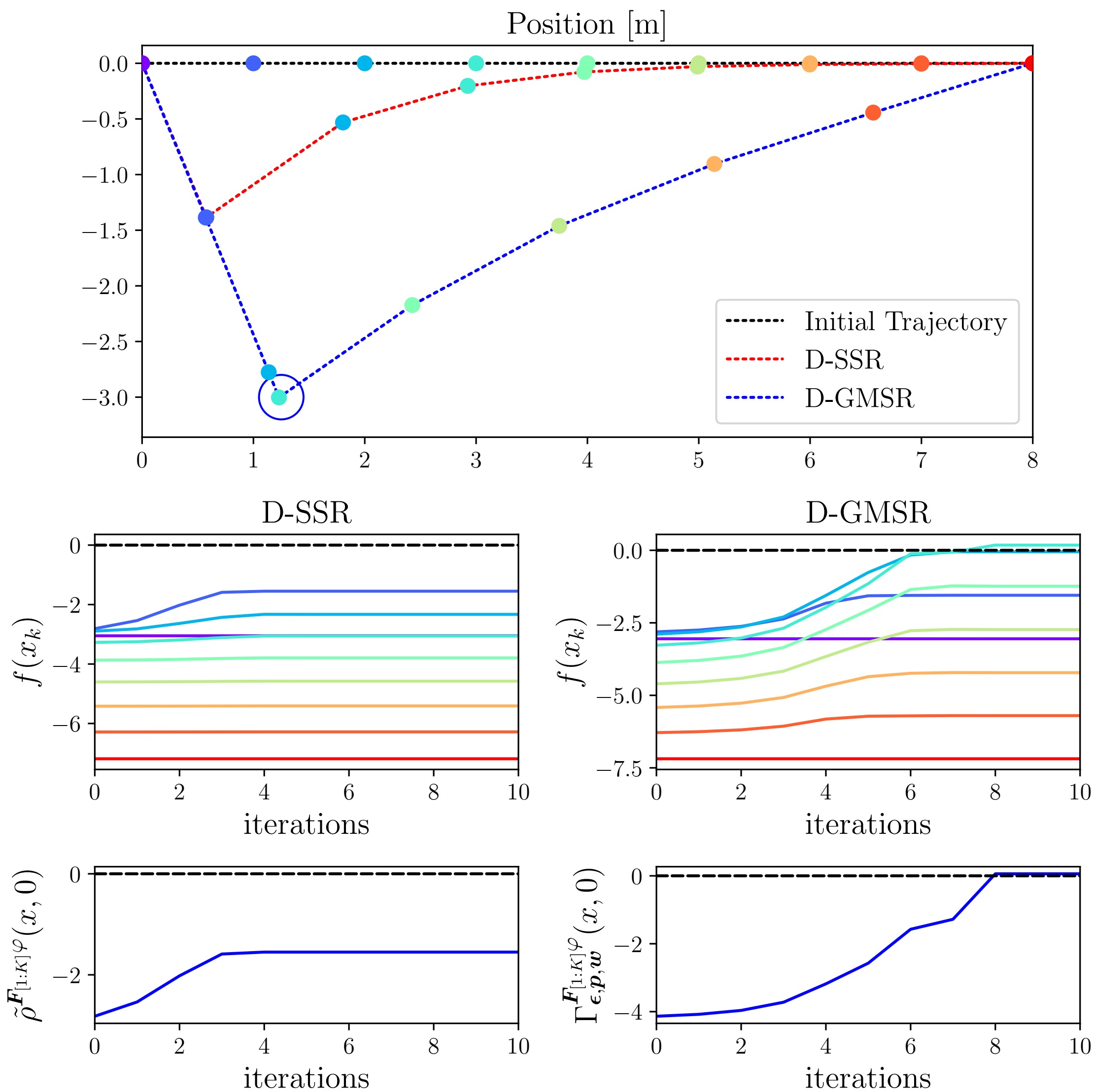


Figure 3.5: Trajectory-level illustration of locality and masking. The black dotted curve denotes the initial trajectory, while the red and blue dotted curves denote the converged trajectories obtained using D-SSR and D-GMSR, respectively. The accompanying predicate-value and robustness plots illustrate how D-SSR concentrates gradient information on a single near-witness node, whereas D-GMSR continues to propagate gradient information to multiple relevant nodes.

Chapter 4

PROX-CONVEX: A STRUCTURE-PRESERVING SEQUENTIAL CONVEX PROGRAMMING ALGORITHM FOR NONCONVEX OPTIMIZATION

4.1 Introduction

The previous chapters established two ingredients that are central to the remainder of the dissertation. First, Chapter 2 developed a numerical optimal-control backbone based on time-dilation, finite-dimensional control parameterization, multiple-shooting discretization, and exact-penalty finite-dimensional formulations. Second, Chapter 3 introduced smooth and exact modeling tools for mission-level specifications, culminating in GMSR-based parameterizations of temporal logic requirements in both discrete and continuous-time. When these two ingredients are combined, the resulting finite-dimensional trajectory-generation problems often exhibit a structured form of nonconvexity. Smooth nonlinear features may appear inside convex penalties, while convex, possibly nonsmooth features may be coupled through smooth outer maps. This chapter develops an optimization algorithm tailored precisely to that structure.

Motivated by this structure, we consider composite problems

$$\min_{x\in\mathbb{R}^m} \; F(x) \;=\; g(x) \;+\; h(C(x)) \;+\; s(R(x)), \tag{4.1}$$

where $g : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is proper, closed, and convex, $h : \mathbb{R}^d \to \mathbb{R}$ is finite-valued, closed, and convex, $C : \mathbb{R}^m \to \mathbb{R}^d$, $s : \mathbb{R}^n \to \mathbb{R}$ are $\mathcal{C}^1$-smooth, and $R : \mathbb{R}^m \to \mathbb{R}^n$, $R(x) = (r_1(x), \ldots, r_n(x))$, with each $r_i : \mathbb{R}^m \to \mathbb{R}$ finite-valued, closed, and convex. For the convergence analysis developed later in this chapter, we further impose mild technical regularity and Lipschitz assumptions, which are stated in the assumptions subsection of Section 4.2. This class generalizes well-studied *convex-composite* problems because of the term $s(R(x))$, which is an essential feature for the motivating examples considered in this dissertation. In particular, the finite-dimensional exact-penalty formulations of Chapter 2 naturally give rise to terms of the form $h(C(x))$, while the smooth and exact mission-specification parameterizations of Chapter 3 naturally give rise to terms of the form $s(R(x))$ when smooth robustness maps act on convex predicate functions.

Remark 4.1 (Modeling breadth). The decomposition (4.1) covers a wide range of structured objectives. Here, g collects convex objectives and indicator functions of convex constraints; $h(C(x))$ captures convex penalties applied to smooth (possibly nonlinear) feature maps; and $s(R(x))$ models smooth couplings of convex (possibly nonsmooth) features such as predicate functions used in temporal logic specifications.

Importantly, the split between R and s is *not unique* and can be chosen to preserve convex structure. Only those features that we want to keep as convex coordinates need to appear inside R. Any additional smooth dependence on x (even if not convex) can be absorbed into the smooth outer map by augmenting the inner map with the identity. Concretely, if a modeling term has the form $\tilde{s}(\bar{R}(x), q(x))$, where $\bar{R}$ is convex (possibly nonsmooth) and q is smooth, then it can be written as $s(R(x))$ by setting $R(x) = (\bar{R}(x), x)$ and defining $s(y, z) = \tilde{s}(y, q(z))$. Thus, one may assume without loss of modeling generality that R collects the convex building blocks, while all smooth components are handled by s.

Representative instances covered by (4.1) are summarized in Table 4.1.

For this broad class of problems, this chapter develops a prox-convex algorithm that *linearizes only the smooth maps* while *preserving all convex structure*. At a current point x_k, we form a convex model $F(x; x_k)$ and solve the strongly convex subproblem

$$x_{k+1} = \arg\min_x \left\{ F(x; x_k) + \tfrac{1}{2}\|x - x_k\|_{Q_k}^2 \right\},$$

with the positive-definite proximal metric

$$Q_k := \mu_k I + H_k^+ \succ 0.$$

Here $\mu_k > 0$ is chosen by a predicted/actual reduction ratio test to enforce sufficient decrease, and $H_k^+ \succeq 0$ optionally injects curvature from $\mathcal{C}^2$ components of the smooth maps, with

Component	Modeling role	Typical instances
$g(x)$	**Convex objectives**	$\frac{1}{2}\|Ax-b\|_2^2,\ \|Ax-b\|_2,\ a^\top x,\ \lambda\|x\|_1,\ \lambda\|x\|_2$, TV, nuclear norm $\|X\|_*$
	Convex constraints via indicator	$\delta_{\mathcal{X}}(x)$ (compact convex sets: boxes/balls/polytopes/simplices) $\delta_{\mathcal{K}}(x)$ (SOC/SDP cones), indicator of affine equalities/dynamics
h(C(x))	**Composition of Convex loss with smooth features**	Nonlinear regression: $\frac{1}{2}\|C(x)\|_2^2,\ \|C(x)\|_1,\ \|C(x)\|_\infty$ Robust fitting: Huber loss on $C(x)$
	Exact penalization of smooth (possibly nonconvex) constraints	For sufficiently large w: Equality constraints $\big(C(x)=0\big)$: $w\|C(x)\|_1$ Inequality constraints $\big(C(x)\le 0\big)$: $w\sum_i (C_i(x))_+$, where $(t)_+=\max\{0,t\}$
s(R(x))	**Temporal Logic Specifications** with convex predicate functions **Disjunction (OR)** $\bigvee_{i=1}^{n}\big(r_i(x)\le 0\big)$ **Implication** $\big(r_1(x)<0\big)\Rightarrow\big(r_2(x)\le 0\big)$ **Compound** $\big(r_1(x)<0\vee r_2(x)<0\big)\Rightarrow$ $\big(r_3(x)\le 0\wedge r_4(x)\le 0\big)$	**Smooth and Exact Parameterizations of the Specifications** [286] **Disjunction (OR)** $-{}^{\vee}h^c_{p,w}\big(-r_1(x),\dots,-r_n(x)\big)\le 0 \iff \prod_{i=1}^n \big(r_i(x)\big)_+^2=0$ **Implication** $-{}^{\vee}h^c_{p,w}\big(r_1(x),\,-r_2(x)\big)\le 0 \iff \big(-r_1(x)\big)_+^2\big(r_2(x)\big)_+^2=0$ **Compound** $-{}^{\vee}h^c_{p,w}\Big({}^{\wedge}h^c_{p,w}\big(r_1(x),r_2(x)\big),\,{}^{\wedge}h^c_{p,w}\big(-r_3(x),-r_4(x)\big)\Big)\le 0 \iff$ $\Big(\big(-r_1(x)\big)_+^2+\big(-r_2(x)\big)_+^2\Big)\Big(\big(r_3(x)\big)_+^2+\big(r_4(x)\big)_+^2\Big)=0$ where $(t)_+=\max\{0,t\}$

Table 4.1: Modeling examples covered by the decomposition $F(x) = g(x)+h(C(x))+s(R(x))$.

$H_k^+ = 0$ when such information is not available. This design keeps g and h exact and retains the convex inner map $R(\cdot)$ inside the linearized s term, while the quadratic regularizer ensures stability and strong convexity of the subproblem.

Under standard boundedness and Lipschitz model-error conditions, accepted steps of prox-convex monotonically decrease F and drive the metric prox-gradient

$$\mathcal{G}_{Q_k}(x_k) \ := \ Q_k\,(x_k - x_{k+1})$$

to zero. Moreover,

$$\min_{0\le j<N} \|\mathcal{G}_{Q_j}(x_j)\| = O(N^{-1/2}),$$

so that $O(\varepsilon^{-2})$ accepted steps suffice to obtain

$$\|\mathcal{G}_{Q_k}(x_k)\| \le \varepsilon.$$

A local error-bound condition further yields a metric step-size error bound and thus local Q-linear convergence of the function values.

The Taylor-like framework of Drusvyatskiy, Ioffe, and Lewis [93] transfers this residual control to stationarity: every cluster point is *limiting* stationary and, by the subdifferential-regularity result developed later in this chapter, also *Fréchet* stationary. The same framework also justifies robustness to inexact subproblem solves and a model-decrease stopping rule. Curvature terms H_k^+ provide local acceleration without weakening global guarantees.

At a high level, the developments of this chapter are: (i) a structure-preserving prox-convex subproblem that avoids unnecessary linearization of convex components, (ii) an adaptive proximal metric with optional second-order information to improve conditioning and local progress, and (iii) a convergence theory with explicit quantitative guarantees. Additional details about these developments are placed in context through the related-work discussion below.

Related works. The development of algorithms for nonsmooth, nonconvex optimization has a rich history, evolving from classical methods for nonlinear programming to modern techniques designed for the large-scale, structured problems common in data science. The prox-convex algorithm developed in this chapter builds upon several key lines of research, which we now survey to place the chapter's contribution in context.

The core idea of iteratively minimizing a simplified model of a difficult problem has deep roots in numerical optimization. For nonlinear least-squares,

$$\min_x \tfrac{1}{2}\|f(x)\|^2,$$

the **Gauss–Newton method** generalizes Newton's method [215, Section 3.3] by linearizing the inner function to create a tractable subproblem [215, Section 10.3]. It is a classical workhorse for nonlinear least-squares and nonlinear regression, widely used for parameter estimation in statistical models and inverse problems [32].

The **Levenberg–Marquardt method** [165, 189] (see also [215, Section 10.3]) improves robustness by minimizing the regularized model

$$\min_{d} \tfrac{1}{2}\|f(x_k) + J_k d\|^2 + \tfrac{\mu_k}{2}\|d\|^2$$

to stabilize poorly conditioned steps, a concept that was later analyzed and extended [54, 212]. In practice, Levenberg–Marquardt underlies standard nonlinear least-squares libraries such as MINPACK and Ceres, and is widely used for curve fitting, parameter estimation, and geometric calibration across engineering and the physical sciences [9, 205, 283].

These ideas are generalized into the powerful **trust-region framework** for smooth objectives,

$$\min_{x} f(x),$$

which at each iteration solves

$$\min_{\|d\|\le\Delta_k} m_k(d)$$

and accepts or rejects by predicted/actual reduction agreement [73, 116]; early analyses established convergence even for nonsmooth objectives [309].

In parallel, **proximal methods** emerged as a unifying abstraction for nonsmooth structures. The classic **proximal point algorithm**, introduced by Martinet [190] and developed by Rockafellar [241], addresses

$$\min_{x} h(x)$$

via regularized subproblems

$$\min_{x} h(x) + \tfrac{1}{2t}\|x - x^k\|^2.$$

While often too expensive to apply directly, the proximal-point method serves as the conceptual basis for many practical splitting schemes, including Douglas–Rachford, ADMM, and Chambolle–Pock, which form the algorithmic core of modern large-scale signal processing, imaging, and distributed optimization methods [47, 63, 72, 98].

This idea specializes to the influential **proximal-gradient** (forward–backward) method for convex "smooth + simple" objectives,

$$\min_x f(x) + h(x),$$

with accelerated variants such as FISTA improving worst-case rates [25, 213]. Proximal-gradient and its accelerated variants are central to modern data science, underpinning large-scale sparse learning, such as Lasso and structured sparsity, and ℓ_1-type regularized inverse problems in imaging and compressed sensing [25, 62, 72, 75, 280]. Comprehensive treatments and unifying perspectives appear in [24, 90, 225].

To achieve the fast local convergence rates, whether superlinear or quadratic, characteristic of second-order methods, proximal algorithms are extended to incorporate curvature information [163]. For additive convex composites,

$$\min_x f(x) + h(x),$$

proximal Newton (PN) at x_k solves the local problem

$$\min_d \ \nabla f(x_k)^\top d + \tfrac{1}{2} d^\top H_k d + h(x_k + d),$$

with

$$H_k \approx \nabla^2 f(x_k),$$

while *proximal quasi-Newton* (PQN) replaces H_k by a curvature approximation B_k. Under standard local assumptions, such as strong convexity near the solution and a Lipschitz-continuous Hessian for PN, or Dennis–Moré-type conditions for PQN, these methods attain local quadratic and superlinear convergence rates, respectively, mirroring smooth Newton/quasi-Newton behavior [163].

The proximal-gradient framework has also been significantly extended to handle fully nonsmooth **nonconvex** problems,

$$\min_x f(x) + h(x).$$

A major theoretical breakthrough in the analysis of nonconvex algorithms came with the application of the **Kurdyka–Łojasiewicz (K–Ł) inequality**. The work of Attouch, Bolte, and Svaiter [18] provided a unified framework for proving convergence of a vast class of descent methods for nonsmooth, nonconvex functions that satisfy the K–Ł property. This framework guarantees that if an algorithm produces a sequence that satisfies a sufficient decrease and a relative error condition, its iterates will converge to a single stationary point.

Bolte, Sabach, and Teboulle [37] provided a prime example of this framework's power by using it to prove the convergence of the widely used Proximal Alternating Linearized Minimization (PALM) algorithm for block problems

$$\min_{x,y} f(x) + g(y) + H(x, y).$$

A complementary strategy replaces direct descent on

$$\min_{x} f(x) + h(x)$$

with a smooth model. Themelis, Stella, and Patrinos [279] introduced the **forward–backward envelope (FBE)**, an exact, strictly continuous penalty for the original problem. Performing a line search on the FBE yields global convergence guarantees for proximal-gradient schemes; moreover, when the search directions satisfy a Dennis–Moré condition, for example through quasi-Newton updates, one obtains *superlinear* convergence to a critical point [279].

Second-order information can also be incorporated in the nonconvex regularized setting. For fully nonconvex regularized objectives,

$$\min_{x} f(x) + h(x),$$

proximal quasi-Newton trust-region (PQNTR) methods minimize the trust-region model

$$\min_{\|d\| \le \Delta_k} \nabla f(x_k)^\top d + \tfrac{1}{2} d^\top B_k d + h(x_k + d),$$

using predicted/actual reduction ratio tests and adaptive radii to secure global convergence to first-order points with worst-case complexity $\mathcal{O}(\varepsilon^{-2})$ [16]. In nonsmooth regularized least squares,

$$\min_x \tfrac{1}{2}\|f(x)\|^2 + h(x),$$

Levenberg–Marquardt ideas yield regularized subproblems

$$\min_d \ \tfrac{1}{2}\|f(x_k) + J_k d\|^2 + \tfrac{\lambda_k}{2}\|d\|^2 + h(x_k + d),$$

which deliver robust global behavior alongside fast local progress [17].

The prox-linear (ProxDescent) algorithm extends the proximal-gradient method to the more general composite setting

$$\min_x g(x) + h(C(x)),$$

by, at each iteration, minimizing the local linearized model

$$\min_x g(x) + h\Big(C(x_k) + \nabla C(x_k)(x - x_k)\Big) + \tfrac{1}{2t}\|x - x_k\|^2.$$

This approach has a long history, with foundational ideas appearing in the 1980s [50, 115, 309], and has been the subject of intense recent study and analysis [61, 92, 167].

Prox-linear and related composite-model methods have been successfully applied to structured nonlinear inverse problems, including robust phase retrieval and quadratic sensing [94], robust blind deconvolution and bilinear sensing [67], and nonconvex constrained trajectory generation via successive convexification [104, 147, 286].

A unifying viewpoint for such composite problems is provided by the *Taylor-like model* framework [93], which studies algorithms driven by first-order models with two-sided quadratic approximation error. Within this framework, a *slope error bound* plays a central role. Under two-sided quadratic model error bounds, a local slope error bound for F implies a *step-size error bound*, which in turn yields local linear convergence of model-based methods [91, 93, 138]. In particular, for broad classes of prox-regular functions, slope error bounds are equivalent, up to constants, to subdifferential error bounds and to the K–Ł property with exponent $1/2$; in the convex setting, quadratic growth, slope error bounds, subdifferential error bounds, K–Ł exponent $1/2$, and step-size error bounds become equivalent notions of local regularity

[38, 91, 93]. Thus, in practice, any one of these conditions is sufficient to obtain a step-size error bound and hence a local linear rate.

Second-order theory for the composite problem

$$\min_x g(x) + h(C(x))$$

is also well developed. Classical local convergence theory for Newton and quasi-Newton methods for this composite problem uses second-order epi-derivatives to establish optimality conditions and sensitivity results [242, 243]. More recent analyses frame the problem using generalized equations [240] and establish *strong metric (sub)regularity* of the associated KKT mapping, particularly for piecewise linear convex functions [70]. The work of Burke et al. [53] extends this framework to piecewise linear-quadratic structures, see also [15, 243], by augmenting the generalized-equations approach with techniques from partial smoothness [168], thereby enabling Newton-style steps with superlinear or quadratic local rates under appropriate regularity [52, 53]. When higher-order derivatives of the smooth component are accessible, higher-order composite methods can further sharpen oracle-complexity bounds and have been shown to substantially reduce iteration counts in practice [86].

Relative to the prox-linear method, prox-convex explicitly accommodates an additional smooth–convex composite term $s(R(x))$, thereby allowing smooth couplings of convex, and possibly nonsmooth, features. Such terms naturally arise when signal temporal logic specifications are encoded via smooth robustness maps acting on convex predicate functions [89, 286]. In powered descent guidance [289], implication-type specifications, that is, state-triggered constraints, give rise to objectives and constraints of the form $s(R(x))$. Likewise, eventual satisfaction requirements for convex range, field-of-view, and direction constraints in perception-aware motion planning for quadrotor flight are modeled through the same smooth–convex composite structure [290]. These are precisely the kinds of structures that arise once the modeling machinery of Chapter 3 is combined with the finite-dimensional transcription and exact-penalty backbone of Chapter 2.

Such structures have also been studied by Ochs et al. in a line of work on nonconvex, nonsmooth optimization [216, 217]. In [216], they propose iteratively reweighted majorization–minimization schemes for problems of the form

$$F_1(x) + F_2(G(x)),$$

with convex F_1, coordinatewise convex G, and nonconvex but coordinatewise nondecreasing F_2. A key contribution is the design of convex majorizers for popular nonconvex penalties, leading to convex subproblems. This framework is flexible within those classes, but it relies on constructing explicit convex models for F_2, which becomes difficult outside the specific penalty families treated there. Moreover, the composite structure $F_2 \circ G$ must still be amenable to efficient convex solvers.

In follow-up work, Ochs et al. [217] propose a unifying Bregman-proximal line-search framework: at each iteration, a convex model together with a Bregman distance generates a trial point, and an Armijo-type condition on the true objective enforces descent. This abstract scheme covers, among other cases, composite objectives $F_2 \circ G$ with smooth, coordinatewise nondecreasing outer functions F_2 and convex G, while more general smooth but sign-indefinite F_2 are not treated in detail. Their analysis, based on the K–Ł property, unifies a broad class of algorithms and guarantees convergence to critical points under standard assumptions. The price of this generality is a mainly asymptotic theory, without explicit iteration-complexity bounds or quantitative links between model decrease, inexact model solves, and ε-stationarity.

Main contributions of this chapter. This chapter develops prox-convex, an extension of the prox-linear method to

$$F(x) = g(x) + h(C(x)) + s(R(x)),$$

that *linearizes only the smooth maps* C and s while *preserving the convex structure* in g, h, and in the convex inner map R. Compared with fully linearizing $s \circ R$, this structure-preserving design is especially effective when R is *highly curved, ill-conditioned, or nonsmooth*: the local model is tighter, accepted steps tend to be larger, the required proximal regularization is milder, and progress per solve improves.

Adaptive globalization. At iterate x_k, prox-convex minimizes a convex model plus a quadratic term $\frac{1}{2}\|x - x_k\|^2_{Q_k}$ with

$$Q_k = \mu_k I + H_k^+ \succ 0.$$

The scalar proximal weight μ_k is chosen by a predicted/actual reduction ratio test that balances model error bounds and step length, yielding monotone decrease on accepted steps while avoiding both instability and overly conservative progress [52, 73].

Second-order injection. When available, the algorithm *injects curvature* via $H_k^+ \succeq 0$ using $\mathcal{C}^2$ components of C and s, which tightens the model, improves conditioning, and accelerates local convergence [51].

Theory backbone and guarantees.

1. *Subdifferential regularity:* under the monotone-or-smooth condition, F is regular so $\widehat{\partial} F = \partial F$ and stationarity is unambiguous [71, 243].
2. *Convergence and complexity with an adaptive proximal metric:* under mild Lipschitz/smoothness constants, a spectral threshold ensures acceptance, only finitely many rejections can occur, and the proximal metrics remain uniformly conditioned; accepted steps yield sufficient decrease and an $O(\varepsilon^{-2})$ bound, in accepted steps, to drive $\|\mathcal{G}_{Q_k}(x_k)\| \leq \varepsilon$.
3. *Linear convergence from error bounds:* building on the error-bound analysis of the prox-linear method with a fixed scalar proximal weight [91], the theory is extended in three directions: matrix-valued, iteration-dependent proximal metrics Q_k; asymmetric quadratic model-error bounds; and corresponding matrix-valued gradient inequalities for the metric prox-gradient. Under a *metric* step-size error bound, this yields local Q-linear convergence of the function values for prox-convex; in the special case of a fixed scalar metric $Q_k = t^{-1}I$ and symmetric model error, the resulting rate constant reduces to that of the prox-linear algorithm.
4. *Taylor-like framework and stationarity:* combining quadratic model-error bounds with boundedness, asymptotic regularity $\|x_{k+1} - x_k\| \to 0$, and the Taylor-like model framework [93], every cluster point of the accepted iterates is limiting stationary. Under

the regularity conditions above, this further implies Fréchet stationarity. The same framework also covers inexact subproblem solves and provides a model-decrease stopping rule.

Numerical validation and implementation. Numerical experiments on representative examples illustrate the practical benefits of preserving convex inner structure, including tighter local models, faster convergence, improved robustness to initialization and proximal-weight selection, and markedly better scaling on ill-conditioned problems. The implementation is available at `https://github.com/UW-ACL/prox-convex`.

The remainder of this chapter is organized as follows. Section 4.2 presents the prox-convex method, including the structural assumptions, the structure-preserving convex local model, the update rule, and the adaptive proximal metric with optional curvature injection. Section 4.3 develops the theoretical analysis, including subdifferential regularity, global descent, complexity, and local linear convergence guarantees. Finally, Section 4.4 presents numerical examples illustrating model quality, convergence behavior, robustness to initialization and proximal-weight selection, and the practical benefits of preserving convex inner structure. Supplementary technical results supporting several model-comparison and Hessian-augmented bounds are collected at the end of the chapter.

4.2 The prox-convex method

This section presents the prox-convex method. After collecting the standing assumptions, we define the convex local subproblem solved at each iteration and then describe an adaptive variable-metric choice $Q_k \succ 0$, with optional curvature injection to improve conditioning and local progress.

4.2.1 Assumptions

We begin by stating a structural condition that rules out "concave-through-a-kink" pathologies, for example $x \mapsto -\|x\|_1$, and ensures regularity.

Informally, for each coordinate i, we require that either the outer map s is locally nondecreasing in the i-th direction, or the inner map r_i is smooth near the point of interest.

Assumption 4.2 (Monotone-or-smooth per coordinate). *For each $\bar{x} \in \operatorname{dom} F$, there exists a neighborhood W of $R(\bar{x})$ such that, for every index $i \in \{1, \ldots, n\}$, the following holds: if r_i is not $\mathcal{C}^1$ on any neighborhood of $\bar{x}$, then*

$$\nabla_i s(z) \;\geq\; 0 \qquad \text{for all } z \in W.$$

In particular, whenever

$$\nabla_i s(R(\bar{x})) < 0,$$

Assumption 4.2 guarantees that r_i is $\mathcal{C}^1$ near $\bar{x}$, so $\nabla r_i(\bar{x})$ is well defined and can be used to build the convex model on coordinates where

$$\nabla_i s(R(\bar{x})) < 0.$$

We next record mild Lipschitz and smoothness conditions on the constituent mappings, which will be used to control model errors, descent estimates, and conditioning of the proximal metric in the theoretical analysis.

Assumption 4.3 (Standing assumptions). *Let $x_0 \in \operatorname{dom} F$ be the starting point and define the initial level set*

$$\mathcal{X}_0 := \{x \in \operatorname{dom} F : F(x) \leq F(x_0)\}.$$

Assume $\mathcal{X}_0$ is nonempty and compact. Moreover, assume:

- *$h : \mathbb{R}^d \to \mathbb{R}$ is convex and L_h-Lipschitz;*
- *$C : \mathbb{R}^m \to \mathbb{R}^d$ is $\mathcal{C}^1$ with β_C-Lipschitz Jacobian;*
- *$s : \mathbb{R}^n \to \mathbb{R}$ is $\mathcal{C}^1$ with β_s-Lipschitz Jacobian;*
- *each $r_i : \mathbb{R}^m \to \mathbb{R}$ is convex and L_{r_i}-Lipschitz, so that $R : \mathbb{R}^m \to \mathbb{R}^n$ is L_R-Lipschitz with respect to $\|\cdot\|_2$, where*

$$L_R := \|(L_{r_1}, \ldots, L_{r_n})\|_2.$$

If there exists a point $x \in \mathcal{X}_0$ and an index i such that

$$\nabla_i s(R(x)) < 0,$$

then preserving convexity of the local model may require linearizing the corresponding component r_i at such points. Therefore, we additionally assume:

- $$\|\nabla s(y)\| \le L_s \qquad \textit{for all } y \in \{R(x) : F(x) \le F(x_0)\};$$
- *any component r_i that is ever linearized on the level set $\mathcal{X}_0$ is $\mathcal{C}^1$ with a β_{r_i}-Lipschitz gradient. Let $\mathcal{I}^-$ denote the set of all such indices, and define*

$$\beta_R := \left(\sum_{i \in \mathcal{I}^-} \beta_{r_i}^2 \right)^{1/2}.$$

4.2.2 Convex local model and prox-convex update

Given an iterate x_k, we construct a convex model of (4.1) by: (i) linearizing the smooth inner map C inside the convex function h, as in classical convex-composite modeling, and (ii) linearizing s while treating each convex coordinate map r_i according to the sign of $\nabla_i s(R(x_k))$ so that convexity is preserved.

We define the convex model of (4.1) at iteration k by

$$F(x; x_k) = g(x) + h\Big(C(x_k) + \nabla C(x_k)(x - x_k)\Big) + s\Big(R(x_k)\Big) + \sum_{i=1}^{n} \nabla_i s\Big(R(x_k)\Big)\, \Phi_i(x; x_k),$$

where

$$\Phi_i(x; x_k) = \begin{cases} r_i(x) - r_i(x_k), & \text{if } \nabla_i s(R(x_k)) \ge 0, \\ \langle \nabla r_i(x_k),\, (x - x_k) \rangle, & \text{if } \nabla_i s(R(x_k)) < 0. \end{cases}$$

We denote the set of *linearized channels* at x_k by

$$\mathcal{I}_k^- := \Big\{ i \in \{1, \ldots, n\} : \nabla_i s(R(x_k)) < 0 \Big\}.$$

Remark 4.4. If $\nabla_i s(R(x_k)) \geq 0$, then the contribution

$$x \;\mapsto\; \nabla_i s(R(x_k))\Big(r_i(x) - r_i(x_k)\Big)$$

is convex, since it is a nonnegative scalar times a convex function, so we keep the convex structure intact in the subproblem.

If $\nabla_i s(R(x_k)) < 0$, then

$$x \;\mapsto\; \nabla_i s(R(x_k))\Big(r_i(x) - r_i(x_k)\Big)$$

is *concave*. To preserve convexity of the model, we replace this concave term by its affine first-order expansion

$$x \;\mapsto\; \nabla_i s(R(x_k))\,\langle \nabla r_i(x_k),\, (x - x_k)\rangle,$$

which is a global convex *majorant* of that concave channel. Thus, $x \mapsto F(x; x_k)$ is convex.

Next, we define the proximal model at iteration k as

$$F_{Q_k}(x; x_k) = F(x; x_k) + \frac{1}{2}\|x - x_k\|^2_{Q_k},$$

where $Q_k = \mu_k I + H_k^+$. Here Q_k is a positive definite matrix that may vary across iterations. Each Q_k combines a scalar proximal weight $\mu_k > 0$ with a curvature approximation $H_k^+ \succeq 0$ that captures local second-order information. Since $F(\cdot; x_k)$ is convex and $Q_k \succ 0$, the function

$$x \mapsto F_{Q_k}(x; x_k)$$

is $\lambda_{\min}(Q_k)$-strongly convex.

The prox-convex update at x_k is the unique minimizer of the strongly convex model $F_{Q_k}(x; x_k)$:

$$x_{k+1} = \arg\min_x F_{Q_k}(x; x_k).$$

This subproblem can be solved efficiently either by interior-point methods [150, 199, 214] or by first-order primal-dual schemes [47, 63, 308], depending on the structure of g and the r_i.

Remark 4.5 (Choosing inner vs. outer linearization). For composite functions, when one mapping is convex and both mappings are $\mathcal{C}^1$, linearizing only the nonconvex component while keeping the convex component intact yields a convex subproblem; under stronger smoothness and Lipschitz assumptions, Proposition 4.22 shows that this single-component linearization is uniformly tighter than linearizing both components.

When both mappings are convex and $\mathcal{C}^1$, a practical guideline is to keep the more curved component exact and linearize the less curved one. Even in the special case where the outer mapping is nondecreasing, so that the composite is convex, see [42, (3.10)], a linearized formulation may still be preferable for compatibility with standard convex solvers.

4.2.3 Adaptive proximal metric

This subsection specifies the proximal metric $Q_k \succ 0$. We describe a practical construction that combines a scalar damping term with optional curvature information, and we show how to adapt the scalar weight online using predicted-versus-actual decrease.

Construction of the Hessian Approximation

When C or s are twice continuously differentiable, one can construct H_k^+ from their Hessians to enhance curvature adaptation. If such Hessian information is unavailable or computationally intractable, H_k^+ can be safely set to zero.

Curvature for $h(C(x))$ (inner-only). Since h is kept *exact* in the subproblem, we do not include any outer curvature. When C is $\mathcal{C}^2$ and tractable, we add the inner curvature weighted by a subgradient of h:

$$H_{C,k} \;:=\; \sum_{j=1}^{d} y_j \, \nabla^2 C_j(x_k), \qquad y \in \partial h\Big(C(x_k)\Big).$$

If h is smooth, we take

$$y = \nabla h(C(x_k)).$$

If C is not $\mathcal{C}^2$ or Hessians are intractable, we set $H_{C,k} = 0$.

Curvature for $s(R(x))$ (outer pullback + inner compensation on linearized channels). When s is $\mathcal{C}^2$, we include the outer pullback. When, in addition, a linearized channel $i \in \mathcal{I}_k^-$ has $r_i \in \mathcal{C}^2$ and a tractable Hessian, we also include its inner curvature, which is negative semidefinite once weighted by the negative outer component:

$$H_{s,k} \;:=\; \underbrace{G_R(x_k)^\top \nabla^2 s(R(x_k))\, G_R(x_k)}_{\text{outer pullback (if } s \in \mathcal{C}^2)} \;+\; \underbrace{\sum_{i \in \mathcal{I}_k^-} \Big[\nabla s(R(x_k))\Big]_i \nabla^2 r_i(x_k)}_{\text{inner compensation on linearized channels (if } r_i \in \mathcal{C}^2)} \quad .$$

Here

$$G_R(x_k) := \begin{bmatrix} g_{1,k}^\top \\ \vdots \\ g_{n,k}^\top \end{bmatrix}, \qquad g_{i,k} \in \partial r_i(x_k).$$

When each r_i is smooth,

$$g_{i,k} = \nabla r_i(x_k) \qquad \text{and} \qquad G_R(x_k) = J_R(x_k).$$

For nonsmooth r_i, the stacked subgradients form an affine first-order model of $R(x)$. For channels with

$$[\nabla s(R(x_k))]_i \geq 0,$$

we keep r_i *exact* in the subproblem and therefore do not add its positive curvature. If s is not $\mathcal{C}^2$ or the required Hessians are unavailable, we set $H_{s,k} = 0$.

Final metric and bounds. We assemble the curvature as

$$H_k^+ \;:=\; \Pi_{\mathbb{S}_+}\Big(H_{C,k} \;+\; H_{s,k}\Big), \qquad Q_k = \mu_k I + H_k^+,$$

where $\Pi_{\mathbb{S}_+}$ denotes the projection onto the positive semidefinite cone.

Remark 4.6 (Justification of the Hessian construction). Under the local second-order smoothness conditions assumed for the Hessian-based model, in particular bounded or Lipschitz Hessians, see Assumption 4.23, the unprojected Hessian model is locally third-order accurate. After PSD projection, the only second-order discrepancy is the nonpositive quadratic "projection gap", while the remaining error terms are third order or higher, see Corollary 4.26.

Adaptive scalar proximal weight

The local error bounds of each model are governed by two varying quantities, namely the outer and inner curvatures of the composite function, so any *static* proximal weight inevitably becomes either unstable, when too small in highly curved regions, or overly conservative, when too large in flat regions. Our adaptive prox-convex scheme therefore tunes the proximal metric

$$Q_k = \mu_k I + H_k^+$$

from iteration to iteration using the agreement between *predicted* and *actual* decrease: when the model is accurate, we decrease the scalar proximal weight μ_k to take bolder steps; when it is not, we increase μ_k to restore reliability, implicitly tracking the unknown local constants without ever computing them.

Whenever C or s is $\mathcal{C}^2$ and Hessian evaluation is tractable, we enrich Q_k with PSD curvature blocks $H_k^+ \succeq 0$, which tighten the model, improve conditioning, and sharpen the local decrease rates. Otherwise, we fall back to first-order damping while preserving $Q_k \succ 0$. This combination yields robust step acceptance with larger effective steps in benign regions and safe contraction near strong inner curvature, translating into faster and more reliable convergence in practice.

The trust-region idea dates back to the 1970s [229] and is classically used to update the radius [73, Ch. 6]; analogous ideas update scalar proximal weights [61]. Here we adopt an adaptive scheme that scales the *proximal metric* in the quadratic term rather than a scalar alone. The resulting adaptive prox-convex algorithm is presented in Algorithm 1.

4.3 Theoretical Analysis

This section establishes global convergence and complexity guarantees for prox-convex. We proceed as follows:

1. We first prove subdifferential regularity

$$\widehat{\partial} F(\bar{x}) = \partial F(\bar{x})$$

Algorithm 1 Adaptive prox-convex with proximal metric update

Input x_0, $\mu_0 > 0$, $\mu_{\min} > 0$, thresholds $0 < \alpha_1 < \alpha_2 < 1$, factors $\nu_{\text{inc}} > 1 > \nu_{\text{dec}} > 0$
for $k = 0, 1, 2, \ldots$ **do**
  Build $H_k^+ \succeq 0$;
  repeat ▷ inner loop (retries on rejection)
    Set $Q_k \leftarrow \mu_k I + H_k^+$;
    $x_k^+ = \arg\min_x \left\{ F(x; x_k) + \frac{1}{2}\|x - x_k\|_{Q_k}^2 \right\}$;
    $\text{Pred}_k = F(x_k) - F_{Q_k}(x_k^+; x_k)$; $\text{Act}_k = F(x_k) - F(x_k^+)$ ▷ predicted and actual decreases
    if $\text{Pred}_k = 0$ or $\|Q_k(x_k - x_k^+)\| \le \epsilon_{\text{term}}$ **then** ▷ termination criteria
      return x_k
    end if
    $\rho_k = \text{Act}_k / \text{Pred}_k$ ▷ acceptance ratio
    if $\rho_k < \alpha_1$ **then** ▷ poor agreement
      $\mu_k \leftarrow \nu_{\text{inc}} \mu_k$
    end if
  until $\rho_k \ge \alpha_1$ ▷ acceptance criteria
  $x_{k+1} \leftarrow x_k^+$
  if $\rho_k > \alpha_2$ **then** ▷ excellent agreement
    $\mu_{k+1} \leftarrow \max\{\mu_{\min}, \nu_{\text{dec}} \mu_k\}$
  else ▷ moderate agreement
    $\mu_{k+1} \leftarrow \mu_k$
  end if
end for

in Theorem 4.7, under the monotone-or-smooth condition of Assumption 4.2.

2. For the actual prox-convex model, we derive two-sided quadratic error bounds with

$$L_U := L_h \beta_C + L_R^2 \beta_s, \qquad L_L := L_U + L_s \beta_R,$$

   in Lemma 4.8.

3. We analyze the acceptance mechanism: a simple spectral threshold guarantees acceptance (Lemma 4.10); at most finitely many consecutive rejections occur (Lemma 4.11); and the adaptive proximal metrics Q_k enjoy uniform spectral bounds (Lemma 4.12).

4. As a consequence, accepted steps achieve a sufficient decrease and an $O(\varepsilon^{-2})$ bound, in accepted steps, for reducing the norm of the metric prox-gradient $\|\mathcal{G}_{Q_k}(x_k)\|$ below ε (Theorem 4.13).

5. We then establish *linear convergence* under local error bounds: extending the error-bound analysis of the prox-linear method with a fixed *scalar* proximal term in [91] to our *iteration-dependent matrix* metric Q_k and asymmetric two-sided model constants (L_L, L_U), we derive matrix-valued gradient inequalities for the metric prox-gradient and show that a *metric* step-size error bound implies local Q-linear decay of function values (Theorem 4.18); in the fixed-metric scalar case $Q_k = t^{-1}I$ with symmetric model constants, the resulting rate constant reduces to that of the prox-linear algorithm (Remark 4.19).

6. Finally, combining our quadratic model error bounds with sufficient decrease, hence asymptotic regularity $\|x_{k+1} - x_k\| \to 0$, and boundedness of the accepted iterates, and invoking the Taylor-like model framework of Drusvyatskiy–Ioffe–Lewis [93], we prove that every cluster point of the accepted iterates is limiting-stationary (Theorem 4.20). Under the regularity conditions of Theorem 4.7, this further implies Fréchet stationarity. The same framework also justifies a model-decrease termination rule and robustness to inexact subproblem solves.

4.3.1 Subdifferential Regularity

We begin by ensuring that the various subdifferential notions agree at the points of interest, so that "stationarity" can be stated unambiguously for limit points of the iterates.

For a proper, lower semicontinuous function F, one typically distinguishes the Fréchet (regular) subdifferential $\widehat{\partial} F$, the limiting (Mordukhovich) subdifferential ∂F, and, for locally Lipschitz F, the Clarke subdifferential $\partial^\circ F$ [71, 243]. These coincide for smooth or convex functions, but in the nonsmooth, nonconvex setting the inclusions

$$\widehat{\partial} F(\bar{x}) \;\subset\; \partial F(\bar{x}) \;\subset\; \partial^\circ F(\bar{x})$$

may be strict.

A simple example is $F(x) = -\|x\|_1$ at $x = 0$, where $\widehat{\partial} F(0) = \emptyset$ $\partial F(0) = \{-1, 1\}$ in one dimension, or the set of sign vectors in higher dimensions, and $\partial^\circ F(0) = [-1, 1]$.

The following result shows that, under our monotone-or-smooth condition, F is *subdifferentially regular* at $\bar{x}$, so that

$$\widehat{\partial} F(\bar{x}) = \partial F(\bar{x}).$$

In particular, all standard first-order stationarity conditions agree at $\bar{x}$.

Theorem 4.7 (Subdifferential regularity)**.** *Suppose Assumption 4.2 holds at $\bar{x}$. Let W be the corresponding neighborhood of $R(\bar{x})$ and define*

$$\mathcal{I}_{\bar{x}} := \Big\{ i \in \{1, \ldots, n\} : \nabla_i s(z) \geq 0 \textit{ for all } z \in W \Big\},$$

so that for every $i \notin \mathcal{I}_{\bar{x}}$, the channel r_i is $\mathcal{C}^1$ on a neighborhood of $\bar{x}$.

Then F is subdifferentially regular at $\bar{x}$; in particular, its Fréchet and limiting subdifferentials coincide:

$$\widehat{\partial} F(\bar{x}) = \partial F(\bar{x}).$$

Moreover,

$$\partial F(\bar{x}) = \partial g(\bar{x}) + \nabla C(\bar{x})^\top \partial h(C(\bar{x})) + \sum_{i \in \mathcal{I}_{\bar{x}}} \nabla_i s(R(\bar{x}))\, \partial r_i(\bar{x}) + \sum_{i \notin \mathcal{I}_{\bar{x}}} \nabla_i s(R(\bar{x}))\, \nabla r_i(\bar{x}),$$

where the sums are Minkowski sums.

Proof. By convex analysis, g is regular with

$$\widehat{\partial} g(\bar{x}) = \partial g(\bar{x}).$$

Since h is convex and C is $\mathcal{C}^1$, the basic chain rule in [243, Thm. 10.6] yields

$$\widehat{\partial}(h(C(\bar{x}))) = \partial(h(C(\bar{x}))) = \nabla C(\bar{x})^\top \partial h(C(\bar{x})).$$

Since s is $\mathcal{C}^1$, it is regular at $R(\bar{x})$ and

$$\partial s(R(\bar{x})) = \{\nabla s(R(\bar{x}))\}.$$

Moreover, each r_i is convex finite, hence R is locally Lipschitz (strictly continuous) near $\bar{x}$.

Thus the equality case of the extended chain rule [243, Thm. 10.49] applies provided $y^\top R$ is regular for $y = \nabla s(R(\bar{x}))$. By Assumption 4.2, for each i, either $y_i \geq 0$, so $y_i r_i$ is convex and therefore regular, or r_i is $\mathcal{C}^1$ near $\bar{x}$, so $y_i r_i$ is smooth and therefore regular. Hence $y^\top R$ is regular and

$$\partial(s(R(\bar{x}))) = D^\star R(\bar{x})\Big[\partial s(R(\bar{x}))\Big] = \sum_{i \in \mathcal{I}_{\bar{x}}} \nabla_i s(R(\bar{x}))\, \partial r_i(\bar{x}) + \sum_{i \notin \mathcal{I}_{\bar{x}}} \nabla_i s(R(\bar{x}))\, \nabla r_i(\bar{x}),$$

where $D^\star R(\bar{x})$ denotes the limiting coderivative of the mapping R at $\bar{x}$. Therefore, $s \circ R$ is regular at $\bar{x}$.

Finally, $h \circ C$ and $s \circ R$ are locally Lipschitz, hence

$$\partial^\infty (h \circ C)(\bar{x}) = \partial^\infty (s \circ R)(\bar{x}) = \{0\},$$

so the horizon qualification in [243, Cor. 10.9] is automatic. Since g, $h \circ C$, and $s \circ R$ are regular at $\bar{x}$, the addition rule yields

$$\partial F(\bar{x}) = \partial g(\bar{x}) + \partial(h(C(\bar{x}))) + \partial(s(R(\bar{x}))),$$

and F is regular at $\bar{x}$, i.e.,

$$\widehat{\partial} F(\bar{x}) = \partial F(\bar{x}).$$

∎

4.3.2 Global descent and complexity bound

We show that the prox-convex model satisfies explicit two-sided quadratic approximation bounds, and therefore fits the Taylor-like model framework.

Lemma 4.8 (Quadratic Model Error)**.** *For all x, the model error is bounded by*

$$-\frac{L_L}{2}\|x - x_k\|^2 \;\leq\; F(x) - F(x; x_k) \;\leq\; \frac{L_U}{2}\|x - x_k\|^2,$$

where

$$L_L := L_U + L_s\beta_R, \qquad L_U := L_h\beta_C + L_R^2\beta_s.$$

Proof. The proof strategy is to decompose the error $F(x) - F(x; x_k)$ into its two main composite parts and then establish both an upper and a lower bound for each component. Let $d_x := x - x_k$. By construction, the terms involving $g(x)$ cancel, leaving

$$F(x) - F(x; x_k) = \underbrace{\Big[h(C(x)) - h\Big(C(x_k) + \nabla C(x_k)d_x\Big)\Big]}_{\text{Term (I): Error from } h\circ C} + \underbrace{\Big[s(R(x)) - s(R(x_k)) - \nabla s(R(x_k))^\top \Phi(x; x_k)\Big]}_{\text{Term (II): Error from } s\circ R}.$$

Bounding Term (I). This term represents the error from linearizing the inner map C. Let

$$\Delta_C := C(x) - C(x_k) - \nabla C(x_k)d_x.$$

Since ∇C is β_C-Lipschitz, we have

$$\|\Delta_C\| \leq \frac{\beta_C}{2}\|d_x\|^2.$$

The L_h-Lipschitz continuity of h then provides the symmetric quadratic bound

$$-\frac{L_h\beta_C}{2}\|d_x\|^2 \;\leq\; \text{(I)} \;\leq\; \frac{L_h\beta_C}{2}\|d_x\|^2.$$

Bounding Term (II). This term represents the error from the prox-convex approximation of the composition $s \circ R$. Let

$$\Delta_R := R(x) - R(x_k).$$

We rewrite Term (II) as

$$\text{(II)} = \underbrace{\Big[s(R(x)) - s(R(x_k)) - \nabla s(R(x_k))^\top \Delta_R\Big]}_{\text{Smoothness Error}} + \underbrace{\nabla s(R(x_k))^\top (\Delta_R - \Phi(x; x_k))}_{\text{Model Design Error}}.$$

The standard inequality for the β_s-smooth function s bounds the *Smoothness Error*. Furthermore, by construction of our model, the *Model Design Error* is always nonpositive.

For each component i,

$$r_i(x) - r_i(x_k) - \Phi_i(x; x_k) = \begin{cases} 0, & \text{if } \nabla_i s\Big(R(x_k)\Big) \geq 0, \\ r_i(x) - r_i(x_k) - \langle \nabla r_i(x_k),\, d_x \rangle, & \text{otherwise.} \end{cases}$$

By convexity of r_i,

$$r_i(x) - r_i(x_k) - \langle \nabla r_i(x_k),\, d_x \rangle \geq 0.$$

This implies

$$\nabla s(R(x_k))^\top (\Delta_R - \Phi(x; x_k)) = \sum_{i=1}^{n} \nabla_i s\Big(R(x_k)\Big)\Big(r_i(x) - r_i(x_k) - \Phi_i(x; x_k)\Big) \leq 0.$$

This immediately gives the upper bound

$$\text{(II)} \leq \frac{\beta_s}{2} \|\Delta_R\|^2 + 0 \leq \frac{\beta_s L_R^2}{2} \|d_x\|^2.$$

For the lower bound, we use the other side of the smoothness inequality together with Cauchy–Schwarz:

$$\text{(II)} \geq -\frac{\beta_s}{2} \|\Delta_R\|^2 - \|\nabla s(R(x_k))\| \, \|\Delta_R - \Phi(x; x_k)\|.$$

The error vector $\Delta_R - \Phi$ has nonzero components only if $\nabla_i s\Big(R(x_k)\Big) < 0$. For these components, our additional assumption provides the bound

$$|r_i(x) - r_i(x_k) - \langle \nabla r_i(x_k),\, d_x \rangle| \leq \frac{\beta_{r_i}}{2} \|d_x\|^2.$$

Therefore,

$$\|\Delta_R - \Phi\| \leq \frac{\beta_R}{2}\|d_x\|^2.$$

Substituting this into the lower bound for Term (II), and using

$$\|\nabla s(R(x_k))\| \leq L_s, \qquad \|\Delta_R\| \leq L_R\|d_x\|,$$

we obtain

$$(\mathrm{II}) \geq -\frac{\beta_s L_R^2}{2}\|d_x\|^2 - \frac{L_s\beta_R}{2}\|d_x\|^2.$$

Adding the bounds for (I) and (II) gives

$$-\frac{L_L}{2}\|d_x\|^2 \leq F(x) - F(x; x_k) \leq \frac{L_U}{2}\|d_x\|^2,$$

where

$$L_L := L_U + L_s\beta_R, \qquad L_U := L_h\beta_C + L_R^2\beta_s.$$

■

Remark 4.9 (Model constants). As will be shown, L_U controls how large Q_k must be to ensure acceptance, and hence directly governs the effective step size, whereas L_L is needed to translate model progress into stationarity guarantees and to establish the linear convergence rate.

Key points.

1. We linearize a channel r_i only when the outer directional influence is negative, so the contribution is locally concave; the affine term is a global majorant and preserves convexity.
2. Curvature of such linearized inner channels enters only through an additive term in L_L, via the design-error bound, and does *not* affect L_U; hence step-size selection depends solely on L_U.
3. If no channel is linearized, that is, $\mathcal{I}^- = \emptyset$, then the bound is symmetric and $L_L = L_U$.

We connect model error bounds to the predicted/actual reduction ratio test, proving that a minimal amount of proximal metric guarantees acceptance.

Lemma 4.10 (Sufficient condition for step acceptance)**.** *Let* $Q_k \succ 0$ *and write* $\sigma_k := \lambda_{\min}(Q_k)$. *If*

$$\sigma_k \;\geq\; \frac{L_U}{2-\alpha_1},$$

then the trial point x_k^+ *is accepted, i.e.,*

$$\rho_k := \mathrm{Act}_k/\mathrm{Pred}_k \geq \alpha_1.$$

Proof. By strong convexity of $x \mapsto F_{Q_k}(x; x_k)$ and optimality of x_k^+,

$$F(x_k) = F_{Q_k}(x_k; x_k) \;\geq\; F_{Q_k}(x_k^+; x_k) + \tfrac{1}{2}\|x_k^+ - x_k\|_{Q_k}^2,$$

and therefore

$$\mathrm{Pred}_k := F(x_k) - F_{Q_k}(x_k^+; x_k) \;\geq\; \tfrac{1}{2}\|x_k^+ - x_k\|_{Q_k}^2. \tag{4.2}$$

Moreover, by expanding F_{Q_k} at x_k^+,

$$\mathrm{Pred}_k = F(x_k) - F(x_k^+; x_k) - \tfrac{1}{2}\|x_k^+ - x_k\|_{Q_k}^2,$$

hence

$$F(x_k) - F(x_k^+; x_k) = \mathrm{Pred}_k + \tfrac{1}{2}\|x_k^+ - x_k\|_{Q_k}^2.$$

By the upper bound in Lemma 4.8 at $x = x_k^+$,

$$\mathrm{Act}_k := F(x_k) - F(x_k^+) \;\geq\; F(x_k) - F(x_k^+; x_k) - \tfrac{L_U}{2}\|x_k^+ - x_k\|^2$$

and therefore

$$\mathrm{Act}_k \;\geq\; \mathrm{Pred}_k + \tfrac{1}{2}\|x_k^+ - x_k\|_{Q_k}^2 - \tfrac{L_U}{2}\|x_k^+ - x_k\|^2.$$

If $x_k^+ - x_k = 0$, then

$$\mathrm{Act}_k = \mathrm{Pred}_k = 0,$$

and the trial point is accepted trivially. Assume instead that $x_k^+ - x_k \neq 0$. Dividing by $\mathrm{Pred}_k > 0$ yields

$$\rho_k = \frac{\mathrm{Act}_k}{\mathrm{Pred}_k} \;\geq\; 1 + \frac{\tfrac{1}{2}\|x_k^+ - x_k\|_{Q_k}^2 - \tfrac{L_U}{2}\|x_k^+ - x_k\|^2}{\mathrm{Pred}_k}.$$

If

$$\tfrac{1}{2}\|x_k^+ - x_k\|_{Q_k}^2 - \tfrac{L_U}{2}\|x_k^+ - x_k\|^2 \geq 0,$$

then $\rho_k \geq 1$, and hence $\rho_k \geq \alpha_1$ since $\alpha_1 < 1$.

Otherwise,

$$\tfrac{1}{2}\|x_k^+ - x_k\|_{Q_k}^2 - \tfrac{L_U}{2}\|x_k^+ - x_k\|^2 < 0,$$

and using the lower bound (4.2) gives

$$\rho_k \;\geq\; 1 + \frac{\tfrac{1}{2}\|x_k^+ - x_k\|_{Q_k}^2 - \tfrac{L_U}{2}\|x_k^+ - x_k\|^2}{\tfrac{1}{2}\|x_k^+ - x_k\|_{Q_k}^2} = 2 - L_U \frac{\|x_k^+ - x_k\|^2}{\|x_k^+ - x_k\|_{Q_k}^2}.$$

Finally,

$$\|x_k^+ - x_k\|_{Q_k}^2 \geq \sigma_k \|x_k^+ - x_k\|^2$$

implies

$$\rho_k \;\geq\; 2 - \frac{L_U}{\sigma_k} \;\geq\; \alpha_1 \qquad \text{whenever} \qquad \sigma_k \geq \frac{L_U}{2 - \alpha_1}.$$

■

Thus, once the trial curvature exceeds the model smoothness threshold L_U, discounted by the acceptance ratio $2 - \alpha_1$, the predicted/actual reduction ratio test always passes.

We next bound the number of consecutive rejections via geometric growth of the scalar proximal weight, thereby ensuring progress of the inner loop.

Lemma 4.11 (Finite rejections)**.** *Assume the setting of Lemma 4.10. Let* $Q_k = \mu_k I + H_k^+$ *with* $H_k^+ \succeq 0$ *fixed within the inner loop, and update* $\mu_k \leftarrow \nu_{\text{inc}}\mu_k$ *on each rejection, with* $\nu_{\text{inc}} > 1$*. Writing*

$$\psi := \frac{L_U}{2 - \alpha_1},$$

after at most

$$N_k \;\leq\; \Big\lceil \log\!\big(\psi/\mu_k\big) \big/ \log(\nu_{\text{inc}}) \Big\rceil_+$$

rejections we have

$$\lambda_{\min}(Q_k) \geq \psi,$$

so the step is accepted. In particular, if $\mu_k \geq \psi$*, then* $N_k = 0$*. Here* $\lceil \cdot \rceil_+ := \max\{0, \lceil \cdot \rceil\}$.

Proof. Within the inner loop the curvature block H_k^+ is fixed and only the scalar μ is updated. After j consecutive rejections,

$$Q^{(j)} = \mu^{(j)} I + H_k^+, \qquad \mu^{(j)} = \nu_{\mathrm{inc}}^{j}\, \mu_k.$$

Since $H_k^+ \succeq 0$,

$$\lambda_{\min}\Big(Q^{(j)}\Big) \geq \mu^{(j)}.$$

By Lemma 4.10, acceptance is guaranteed once

$$\lambda_{\min}(Q^{(j)}) \geq \psi,$$

which is implied by $\mu^{(j)} \geq \psi$. Using $\mu^{(j)} = \nu_{\mathrm{inc}}^{j} \mu_k$ gives

$$j \;\geq\; \frac{\log(\psi/\mu_k)}{\log(\nu_{\mathrm{inc}})}.$$

Taking the smallest integer j satisfying the inequality and clipping at 0 yields the claim. If $\mu_k \geq \psi$, then $N_k = 0$ and the first trial is accepted. ■

We next show that the adaptive proximal metrics remain uniformly well-conditioned, which is crucial for descent and complexity.

Lemma 4.12 (Uniform spectral bounds for Q_k)**.** *Run Algorithm 1 with*

$$0 < \alpha_1 < \alpha_2 < 1, \qquad \nu_{\mathrm{inc}} > 1, \qquad \nu_{\mathrm{dec}} \in (0,1), \qquad \mu_{\min} > 0.$$

Let

$$Q_k = \mu_k I + H_k^+$$

where H_k^+ is constructed as in Section 4.2.3. Then there exist constants

$$0 < \underline{q} \leq \overline{q} < \infty$$

independent of k, such that

$$\underline{q}\, I \;\preceq\; Q_k \;\preceq\; \overline{q}\, I \qquad (\forall k), \tag{4.3}$$

with

$$\underline{q} := \min\{\mu_0, \mu_{\min}\}, \qquad \overline{q} := \max\Big\{\mu_0,\ \frac{\nu_{\mathrm{inc}} L_U}{2-\alpha_1}\Big\} + \overline{h},$$

and the curvature cap

$$\boxed{\overline{h} := \beta_C L_h + \beta_s L_R^2 + L_s\,\beta_R}.$$

Proof. Step 1 (Spectral bound for H_k^+). Because the projection $\Pi_{\mathbb{S}_+}$ is nonexpansive,

$$\|H_k^+\| \le \|H_{C,k} + H_{s,k}\|.$$

Assumption 4.3 and standard Lipschitz–Hessian bounds yield

$$\Big\|\sum_j y_j\,\nabla^2 C_j(x_k)\Big\| \ \le\ \beta_C\,\|y\|_2 \ \le\ \beta_C\,L_h,$$

and

$$\Big\|G_R(x_k)^\top \nabla^2 s(R(x_k))\,G_R(x_k)\Big\| \ \le\ \beta_s\,L_R^2.$$

If each linearized channel

$$i \in \mathcal{I}_k^- \subseteq \mathcal{I}^-$$

satisfies

$$r_i \in \mathcal{C}^1 \qquad \text{with} \qquad \|\nabla^2 r_i(x)\| \le \beta_{r_i},$$

then using

$$\|\nabla s(R(x_k))\| \le L_s,$$

we have

$$\Bigg\|\sum_{i\in\mathcal{I}_k^-}\Big[\nabla s(R(x_k))\Big]_i\,\nabla^2 r_i(x_k)\Bigg\| \ \le\ \|\nabla s(R(x_k))\|\,\beta_R \ \le\ L_s\,\beta_R.$$

Therefore,

$$0 \ \le\ \lambda_{\min}(H_k^+) \ \le\ \lambda_{\max}(H_k^+) \ \le\ \overline{h} := \beta_C L_h + \beta_s L_R^2 + L_s\beta_R.$$

Step 2 (Lower spectral bound for Q_k). By the update rules in Algorithm 1,

$$\mu_{k+1} = \begin{cases} \nu_{\mathrm{inc}}\mu_k, & \text{if rejected,}\\[2ex] \max\{\mu_{\min},\ \nu_{\mathrm{dec}}\mu_k\}, & \text{if accepted with excellent agreement,}\\[2ex] \mu_k, & \text{if accepted with moderate agreement,} \end{cases}$$

so by induction

$$\mu_k \geq \min\{\mu_0, \mu_{\min}\} \qquad \text{for all } k.$$

Since $H_k^+ \succeq 0$, we have

$$\lambda_{\min}(Q_k) \geq \mu_k \geq \underline{q},$$

hence

$$\underline{q}\, I \preceq Q_k.$$

Step 3 (Upper spectral bound for Q_k). Let

$$\psi := \frac{L_U}{2 - \alpha_1}.$$

By Lemma 4.11, within any inner loop the step is accepted once $\mu \geq \psi$, and during rejections we only multiply μ by ν_{inc}. Therefore the scalar at acceptance obeys

$$\mu^{\text{acc}} \leq \nu_{\text{inc}}\psi = \frac{\nu_{\text{inc}} L_U}{2 - \alpha_1},$$

and during that inner loop μ never exceeds this cap. Across outer iterations, acceptance either keeps or shrinks μ, hence

$$\mu_k \;\leq\; \max\Big\{\mu_0,\; \frac{\nu_{\text{inc}} L_U}{2 - \alpha_1}\Big\} \qquad (\forall k).$$

Finally,

$$\lambda_{\max}(Q_k) \leq \mu_k + \lambda_{\max}(H_k^+) \leq \max\Big\{\mu_0, \frac{\nu_{\text{inc}} L_U}{2 - \alpha_1}\Big\} + \overline{h} =: \overline{q}.$$

Thus $Q_k \preceq \overline{q}\, I$, completing (4.3). ■

With acceptance and conditioning in place, we derive a per-step decrease and a global $O(\varepsilon^{-2})$ bound in accepted steps. Throughout, let $\mathcal{S} \subset \mathbb{N}$ denote the set of accepted indices produced by Algorithm 1, and write

$$\Delta_F := F(x_0) - \inf F < \infty.$$

Theorem 4.13 (Global descent and $O(\varepsilon^{-2})$ complexity on accepted steps)**.** *Let F be bounded below and let $\{x_k\}_{k\in\mathcal{S}}$ be the accepted iterates of Algorithm 1 with ratio threshold $\alpha_1 \in (0,1)$ and metrics Q_k satisfying Lemma 4.12. Then for every $k \in \mathcal{S}$,*

$$F(x_k) - F(x_{k+1}) \;\geq\; \frac{\alpha_1}{2}\,\|x_{k+1}-x_k\|^2_{Q_k} \;\geq\; \frac{\alpha_1\,\underline{q}}{2}\,\|x_{k+1}-x_k\|^2. \tag{4.4}$$

Consequently,

$$\sum_{k\in\mathcal{S}} \|x_{k+1}-x_k\|^2_{Q_k} < \infty,$$

hence

$$\|x_{k+1}-x_k\|_{Q_k} \to 0.$$

By Lemma 4.12,

$$\|\mathcal{G}_{Q_k}(x_k)\| = \|Q_k(x_k - x_{k+1})\| \leq \overline{q}\,\|x_{k+1}-x_k\| \leq \frac{\overline{q}}{\sqrt{\underline{q}}}\,\|x_{k+1}-x_k\|_{Q_k} \to 0$$

along $\mathcal{S}$.

Moreover, with $\Delta_F := F(x_0) - \inf F$ and for the first N accepted indices $\mathcal{S}_N$,

$$\frac{1}{N}\sum_{k\in\mathcal{S}_N}\|\mathcal{G}_{Q_k}(x_k)\|^2 \;\leq\; \frac{2\,\overline{q}\,\Delta_F}{\alpha_1\,N}, \qquad \min_{k\in\mathcal{S}_N}\|\mathcal{G}_{Q_k}(x_k)\| \;\leq\; \sqrt{\frac{2\,\overline{q}\,\Delta_F}{\alpha_1\,N}}. \tag{4.5}$$

Hence $N = O(\varepsilon^{-2})$ accepted steps suffice to reach

$$\min_{k\in\mathcal{S}_N}\|\mathcal{G}_{Q_k}(x_k)\| \leq \varepsilon.$$

Proof. For accepted k,

$$\mathrm{Act}_k \geq \alpha_1\,\mathrm{Pred}_k.$$

Strong convexity of $x \mapsto F_{Q_k}(x;x_k)$ and optimality of x_{k+1} give

$$\mathrm{Pred}_k \geq \tfrac{1}{2}\|x_{k+1}-x_k\|^2_{Q_k},$$

yielding the first inequality in (4.4); the second follows from $Q_k \succeq \underline{q}I$.

Summing over accepted indices and using $\inf F > -\infty$ gives

$$\sum_{k\in\mathcal{S}} \|x_{k+1}-x_k\|^2_{Q_k} < \infty.$$

For (4.5), note that $Q_k^{-1} \succeq \overline{q}^{-1} I$, so

$$F(x_k) - F(x_{k+1}) \;\geq\; \frac{\alpha_1}{2}\,\|\mathcal{G}_{Q_k}(x_k)\|^2_{Q_k^{-1}} \;\geq\; \frac{\alpha_1}{2\overline{q}}\,\|\mathcal{G}_{Q_k}(x_k)\|^2.$$

Summing over $\mathcal{S}_N$, dividing by N, and using min $\leq$ average gives the claim. ■

Remark 4.14 (Effect of rejections). If at most $j_{\max}$ consecutive rejections occur between accepted steps, cf. Lemma 4.11, then at most

$$(1 + j_{\max})N$$

subproblem solves are needed to obtain N accepted steps. Thus, the overall evaluation complexity remains $O(\varepsilon^{-2})$ up to a constant factor.

Remark 4.15 (Unconditional decrease without the ratio test). Combining the same inequalities used in the proof of Lemma 4.10 yields

$$F(x_k) - F(x_{k+1}) \;\geq\; \tfrac{1}{2}\Big(2 - \frac{L_U}{\sigma_k}\Big)\|x_{k+1} - x_k\|^2_{Q_k}, \qquad \sigma_k := \lambda_{\min}(Q_k).$$

Thus, whenever

$$\sigma_k > \tfrac{1}{2} L_U,$$

we have descent even *without* invoking the ratio test.

4.3.3 Linear rate of convergence

Conceptually, this subsection mirrors the prox-linear analysis with a fixed *scalar* proximal weight in [91], but in a more general setting: our prox-convex scheme uses an *iteration-dependent matrix metric* Q_k and relies on the *asymmetric* two-sided model bounds of Lemma 4.8, with distinct constants L_U and L_L rather than a single symmetric constant.

We first derive matrix-valued gradient inequalities for the metric prox-gradient $\mathcal{G}_{Q_k}$, a matrix analogue of [91, Lem. 5.1]. Combined with a *metric* step-size error bound in the sense of [91, Def. 5.4], these estimates yield local Q-linear convergence of function values following the template of [91, Thm. 5.5]. In the fixed-metric scalar case

$$Q_k = t^{-1} I \qquad \text{and} \qquad L_L = L_U,$$

the resulting rate constant reduces to that of the prox-linear algorithm.

Lemma 4.16 (Gradient inequality for prox-convex)**.** *Let*

$$x_{k+1} = \arg\min_x \left\{ F(x; x_k) + \tfrac{1}{2}\|x - x_k\|^2_{Q_k} \right\}$$

and define the metric prox-gradient mapping

$$\mathcal{G}_{Q_k}(x_k) \;:=\; Q_k(x_k - x_{k+1}), \qquad Q_k^{-1}\mathcal{G}_{Q_k}(x_k) = x_k - x_{k+1}.$$

Then for all $y \in \mathbb{R}^m$, the following hold.

$$F(y; x_k) \;\geq\; F_{Q_k}(x_{k+1}; x_k) + \langle \mathcal{G}_{Q_k}(x_k),\, y - x_k\rangle + \tfrac{1}{2}\langle Q_k^{-1}\mathcal{G}_{Q_k}(x_k),\, \mathcal{G}_{Q_k}(x_k)\rangle. \tag{4.6}$$

If Lemma 4.8 holds with constants L_U, L_L and $\sigma_k := \lambda_{\min}(Q_k)$, then

$$F(y) \;\geq\; F(x_{k+1}) + \langle \mathcal{G}_{Q_k}(x_k),\, y - x_k\rangle + \left(1 - \tfrac{L_U}{2\sigma_k}\right)\langle Q_k^{-1}\mathcal{G}_{Q_k}(x_k),\, \mathcal{G}_{Q_k}(x_k)\rangle - \tfrac{L_L}{2}\,\|y - x_k\|^2. \tag{4.7}$$

In particular, setting $y = x_k$ gives the sufficient decrease estimate

$$F(x_k) - F(x_{k+1}) \;\geq\; \left(1 - \tfrac{L_U}{2\sigma_k}\right)\langle Q_k^{-1}\mathcal{G}_{Q_k}(x_k),\, \mathcal{G}_{Q_k}(x_k)\rangle = \tfrac{1}{2}\left(2 - \tfrac{L_U}{\sigma_k}\right)\|x_{k+1} - x_k\|^2_{Q_k}. \tag{4.8}$$

When $Q_k = t^{-1}I$ and $L_L = L_U = L$, the symmetric-error-bounds case of [91], (4.7) reduces to

$$F(y) \;\geq\; F(x_{k+1}) + \langle \mathcal{G}_{t^{-1}}(x_k), y - x_k\rangle + \tfrac{t}{2}(2 - Lt)\,\|\mathcal{G}_{t^{-1}}(x_k)\|^2 - \tfrac{L}{2}\|y - x_k\|^2,$$

which matches [91, Lem. 5.1] in the scalar-stepsize case.

Proof. For brevity let $d_k := x_{k+1} - x_k$ and

$$\mathcal{G}_{Q_k} := \mathcal{G}_{Q_k}(x_k) = Q_k(x_k - x_{k+1}) = -Q_k d_k.$$

Consider the strongly convex function $F_{Q_k}(x; x_k)$. Since x_{k+1} is its unique minimizer, for all y,

$$F_{Q_k}(y; x_k) \;\geq\; F_{Q_k}(x_{k+1}; x_k) + \tfrac{1}{2}\|y - x_{k+1}\|^2_{Q_k}.$$

Therefore,

$$F(y;x_k) + \tfrac{1}{2}\|y - x_k\|_{Q_k}^2 \;\geq\; F_{Q_k}(x_{k+1};x_k) + \tfrac{1}{2}\|y - x_{k+1}\|_{Q_k}^2.$$

Write

$$y - x_{k+1} = (y - x_k) + Q_k^{-1}\mathcal{G}_{Q_k}$$

and expand:

$$\|y - x_{k+1}\|_{Q_k}^2 = \|y - x_k\|_{Q_k}^2 + 2\langle \mathcal{G}_{Q_k},\, y - x_k\rangle + \langle Q_k^{-1}\mathcal{G}_{Q_k}, \mathcal{G}_{Q_k}\rangle.$$

Substituting this into the previous inequality and cancelling $\frac{1}{2}\|y - x_k\|_{Q_k}^2$ from both sides yields (4.6).

For the function values, Lemma 4.8 gives, for all y,

$$F(y) \;\geq\; F(y;x_k) - \tfrac{L_L}{2}\|y - x_k\|^2.$$

Combining this with (4.6) yields

$$F(y) \;\geq\; F_{Q_k}(x_{k+1};x_k) + \langle \mathcal{G}_{Q_k}, y - x_k\rangle + \tfrac{1}{2}\langle Q_k^{-1}\mathcal{G}_{Q_k}, \mathcal{G}_{Q_k}\rangle - \tfrac{L_L}{2}\|y - x_k\|^2. \tag{4.9}$$

Lemma 4.8 also gives, at x_{k+1},

$$F(x_{k+1};x_k) \;\geq\; F(x_{k+1}) - \tfrac{L_U}{2}\|d_k\|^2.$$

Thus

$$F_{Q_k}(x_{k+1};x_k) = F(x_{k+1};x_k) + \tfrac{1}{2}\|d_k\|_{Q_k}^2$$

satisfies

$$F_{Q_k}(x_{k+1};x_k) \;\geq\; F(x_{k+1}) + \tfrac{1}{2}\|d_k\|_{Q_k}^2 - \tfrac{L_U}{2}\|d_k\|^2.$$

Since $Q_k \succeq \sigma_k I$, we have

$$\|d_k\|^2 \leq \sigma_k^{-1}\|d_k\|_{Q_k}^2,$$

so

$$F_{Q_k}(x_{k+1};x_k) \;\geq\; F(x_{k+1}) + \left(\tfrac{1}{2} - \tfrac{L_U}{2\sigma_k}\right)\|d_k\|_{Q_k}^2$$

and therefore

$$F_{Q_k}(x_{k+1}; x_k) \;\geq\; F(x_{k+1}) + \left(\tfrac{1}{2} - \tfrac{L_U}{2\sigma_k}\right)\langle Q_k^{-1}\mathcal{G}_{Q_k}, \mathcal{G}_{Q_k}\rangle.$$

Combining this with (4.9) yields

$$F(y) \;\geq\; F(x_{k+1}) + \langle \mathcal{G}_{Q_k}, y - x_k\rangle + \left(1 - \tfrac{L_U}{2\sigma_k}\right)\langle Q_k^{-1}\mathcal{G}_{Q_k}, \mathcal{G}_{Q_k}\rangle - \tfrac{L_L}{2}\|y - x_k\|^2,$$

which is (4.7). Setting $y = x_k$ removes the last term and gives (4.8). ■

Definition 4.17 (Metric step-size error bound)**.** *Let $\{x_k\}$ be generated by Algorithm 1, and let $x^\star$ be a limit point of $\{x_k\}$. We say that $\{x_k\}$ satisfies a* metric step-size error bound at *$x^\star$ if there exist a constant $\kappa > 0$, a neighborhood $\mathcal{U}$ of $x^\star$, and an index K such that, for all $k \geq K$,*

$$x_k \in \mathcal{U}, \qquad \|x_k - x^\star\| \;\leq\; \kappa \left\|\mathcal{G}_{Q_k}(x_k)\right\|. \tag{4.10}$$

Relation to existing error-bound notions and implications for prox-convex. When

$$Q_k = (1/t)I,$$

condition (4.10) reduces exactly to the scalar step-size error bound of [91, Def. 5.4]. More generally, if the metrics are uniformly well-conditioned,

$$\sigma_{\min} I \preceq Q_k \preceq \sigma_{\max} I,$$

then the metric step-size error bound (4.10) is equivalent, up to constants depending only on $(\sigma_{\min}, \sigma_{\max})$, to the corresponding scalar step-size error bound by norm equivalence. Combined with the Taylor-like model error bounds of $F_{Q_k}(\cdot; x_k)$, any local slope error bound, or in the convex/composite case, any of its equivalent forms such as a K–Ł exponent 1/2 or quadratic growth, therefore yields a step-size error bound of the form (4.10) [93, Thm. 3.5], which is exactly the ingredient used below to establish Q-linear convergence of the prox-convex iterates.

Theorem 4.18 (Q-linear convergence under a metric step-size error bound)**.** *Assume Definition 4.17 holds at a limit point $x^\star$ of the sequence $\{x_k\}$ generated by Algorithm 1, and the uniform lower spectral bound satisfies*

$$\underline{q} > \frac{L_U}{2-\alpha_1}.$$

Set

$$F^\star := F(x^\star).$$

Then all steps are accepted, by Lemma 4.10, and

$$F(x_{k+1}) - F^\star \;\le\; q^\star \Big(F(x_k) - F^\star\Big) \qquad \text{for all } k \ge K, \tag{4.11}$$

with contraction factor

$$q^\star := 1 - \min\left\{1,\; \frac{2 - \frac{L_U}{\underline{q}}}{\overline{q}\Big(2 + L_L\Big)\,\bar{\kappa}^{\,2}}\right\} \in [0,1), \qquad \bar{\kappa} := \max\{1, \kappa\}. \tag{4.12}$$

Consequently, $\{F(x_k)\}$ converges Q-linearly to $F(x^\star)$.

Proof. Let

$$\mathcal{G}_{Q_k} := \mathcal{G}_{Q_k}(x_k), \qquad \sigma_k := \lambda_{\min}(Q_k).$$

Applying Lemma 4.16 with $y = x^\star$, and using

$$\langle Q_k^{-1}\mathcal{G}_{Q_k}, \mathcal{G}_{Q_k}\rangle \ge \overline{q}^{-1}\|\mathcal{G}_{Q_k}\|^2 \qquad \text{and} \qquad \sigma_k \ge \underline{q},$$

gives

$$F(x_{k+1}) - F(x^\star) \;\le\; \|\mathcal{G}_{Q_k}\|\,\|x_k - x^\star\| + \frac{L_L}{2}\|x_k - x^\star\|^2 + \frac{L_U/\underline{q} - 2}{2\overline{q}}\,\|\mathcal{G}_{Q_k}\|^2. \tag{4.13}$$

Define

$$\kappa_k := \max\left\{1,\; \frac{\|x_k - x^\star\|}{\|\mathcal{G}_{Q_k}\|}\right\},$$

with the convention that the ratio is 0 when $\mathcal{G}_{Q_k} = 0$. Then

$$\|x_k - x^\star\| \le \kappa_k \|\mathcal{G}_{Q_k}\|$$

and

$$\|x_k - x^\star\|^2 \le \kappa_k^2 \|\mathcal{G}_{Q_k}\|^2.$$

Therefore, from (4.13),

$$F(x_{k+1}) - F(x^\star) \;\le\; \left[\kappa_k + \frac{L_L}{2}\kappa_k^2 + \frac{L_U/\underline{q} - 2}{2\overline{q}}\right]\|\mathcal{G}_{Q_k}\|^2,$$

and hence

$$F(x_{k+1}) - F(x^\star) \;\le\; \left[\left(1 + \frac{L_L}{2}\right)\kappa_k^2 + \frac{L_U/\underline{q} - 2}{2\overline{q}}\right]\|\mathcal{G}_{Q_k}\|^2.$$

On the other hand, the sufficient decrease estimate (4.8) and the bound

$$\langle Q_k^{-1}\mathcal{G}_{Q_k}, \mathcal{G}_{Q_k}\rangle \ge \overline{q}^{-1}\|\mathcal{G}_{Q_k}\|^2$$

yield

$$F(x_k) - F(x_{k+1}) \;\ge\; \left(1 - \frac{L_U}{2\underline{q}}\right)\langle Q_k^{-1}\mathcal{G}_{Q_k}, \mathcal{G}_{Q_k}\rangle \;\ge\; \frac{2 - \frac{L_U}{\underline{q}}}{2\overline{q}}\,\|\mathcal{G}_{Q_k}\|^2.$$

Combining the last two displays, we obtain

$$\frac{F(x_{k+1}) - F^\star}{F(x_k) - F(x_{k+1})} \;\le\; \frac{\left(1 + \frac{L_L}{2}\right)\kappa_k^2 + \frac{L_U/\underline{q} - 2}{2\overline{q}}}{\frac{2 - \frac{L_U}{\underline{q}}}{2\overline{q}}} = \frac{2\overline{q}\left(1 + \frac{L_L}{2}\right)\kappa_k^2}{2 - \frac{L_U}{\underline{q}}} - 1.$$

Hence

$$F(x_{k+1}) - F^\star \;\le\; \hat{q}_k^\star\left(F(x_k) - F(x_{k+1})\right),$$

where

$$\hat{q}_k^\star := \frac{2\overline{q}\left(1 + \frac{L_L}{2}\right)\kappa_k^2}{2 - \frac{L_U}{\underline{q}}} - 1.$$

Rewriting this as

$$F(x_{k+1}) - F^\star \le \frac{\hat{q}_k^\star}{1 + \hat{q}_k^\star}\left(F(x_k) - F^\star\right),$$

we see that the per-iteration contraction factor is

$$q_k^\star := \frac{\hat{q}_k^\star}{1 + \hat{q}_k^\star} = 1 - \frac{2 - \frac{L_U}{\underline{q}}}{2\overline{q}\left(1 + \frac{L_L}{2}\right)\kappa_k^2}.$$

Now invoke the metric step-size error bound (4.10). For all sufficiently large k,

$$\|x_k - x^\star\| \le \kappa \|\mathcal{G}_{Q_k}\|,$$

and therefore

$$\kappa_k \le \bar{\kappa} := \max\{1, \kappa\}.$$

Hence, for all sufficiently large k,

$$q_k^\star \;\le\; 1 - \frac{2 - \frac{L_U}{\underline{q}}}{2\,\overline{q}\Big(1 + \frac{L_L}{2}\Big)\,\bar{\kappa}^{\,2}} = 1 - \frac{2 - \frac{L_U}{\underline{q}}}{\overline{q}\Big(2 + L_L\Big)\,\bar{\kappa}^{\,2}}$$

and thus

$$q_k^\star \;\le\; 1 - \min\left\{1,\; \frac{2 - \frac{L_U}{\underline{q}}}{\overline{q}\Big(2 + L_L\Big)\,\bar{\kappa}^{\,2}}\right\} = q^\star \in [0, 1).$$

This gives (4.11) and completes the proof. ∎

Remark 4.19 (Comparison with prox-linear). In the scalar case $Q_k = t^{-1} I$, the prox-convex step coincides with the prox-linear step with parameter t from [91], and

$$\mathcal{G}_{Q_k}(x_k) = \mathcal{G}_{t^{-1}}(x_k).$$

The metric error bound (4.10) then specializes exactly to the scalar step-size error bound of [91, Def. 5.4]:

$$\|x_k - x^\star\| \;\le\; \kappa\,\|\mathcal{G}_{t^{-1}}(x)\|.$$

In this scalar setting we have

$$\underline{q} = \overline{q} = t^{-1} \qquad \text{and} \qquad L_U = L_L = L,$$

where L is the symmetric-error-bounds constant of [91]. The general contraction factor (4.12) therefore becomes

$$q^\star \;\le\; 1 - \frac{t(2 - Lt)}{(2 + L)\,\bar{\kappa}^{\,2}}, \qquad \bar{\kappa} = \max\{1, \kappa\}.$$

Choosing the standard stepsize $t = 1/L$ yields

$$q^\star \;\le\; 1 - \frac{1}{L(2 + L)\,\bar{\kappa}^{\,2}},$$

which *exactly matches* the prox-linear linear rate constant in [91, Thm. 5.5].

4.3.4 Taylor-like framework and stationarity

We now complete the argument using the Taylor-like model framework of Drusvyatskiy–Ioffe–Lewis [93]. Their key estimate [93, Thm. 3.1], proved via Ekeland's variational principle [99], shows that a uniform quadratic model-error bound yields a nearby point with small stationarity measure; see also [93, Cor. 3.2].

Building on this, [93, Cor. 3.3] implies subsequence stationarity under exact model minimization, asymptotic regularity

$$\|x_{k+1} - x_k\| \to 0,$$

and boundedness. In our setting, these requirements are ensured by

1. two-sided quadratic model error bounds, Lemma 4.8;
2. sufficient decrease and asymptotic regularity along accepted steps, Theorem 4.13;
3. boundedness of the accepted subsequence.

Therefore, every cluster point of the accepted iterates is first-order stationary; combined with Theorem 4.7, stationarity holds for both the limiting and Fréchet subdifferentials.

Theorem 4.20 (Stationary cluster points)**.** *Let $\{x_k\}_{k\in\mathcal{S}}$ be the accepted iterates of Algorithm 1. Assume F is proper, closed, bounded below, $\{x_k\}_{k\in\mathcal{S}}$ is bounded, and Lemma 4.8 holds. Then every cluster point $\bar{x}$ of $\{x_k\}_{k\in\mathcal{S}}$ is first-order stationary:*

$$0 \in \partial F(\bar{x}).$$

Proof. By Theorem 4.13, we have

$$\|x_{k+1} - x_k\| \to 0$$

along $\mathcal{S}$.

Moreover, Lemma 4.8 and the spectral upper bound

$$\lambda_{\max}(Q_k) \le \overline{q}$$

imply that for all y and all accepted indices $k \in \mathcal{S}$,

$$\left|F_{Q_k}(y; x_k) - F(y)\right| \le \frac{L_L + \lambda_{\max}(Q_k)}{2} \|y - x_k\|^2 \le \omega(\|y - x_k\|),$$

where

$$\omega(r) := \tfrac{1}{2}(L_L + \overline{q})r^2.$$

Thus the growth condition of [93, Cor. 3.3] holds.

Since $\{x_k\}_{k\in\mathcal{S}}$ is bounded, it has a cluster point $\bar{x}$; take a subsequence $x_{k_i} \to \bar{x}$. By accepted-step monotonicity and boundedness below, $F(x_k)$ converges along $\mathcal{S}$, hence $F(x_{k_i})$ converges as well; in particular,

$$(x_{k_i}, F(x_{k_i})) \to (\bar{x}, F(\bar{x})).$$

Applying [93, Cor. 3.3] yields that $\bar{x}$ is stationary for F, i.e., $0 \in \partial F(\bar{x})$. ■

Remark 4.21 (Inexact solves and model-decrease termination). By the Taylor-like framework of [93], our two-sided model bounds imply robustness to *inexact* subproblem solves. If

$$F_{Q_k}(x_{k+1}; x_k) \ \le \ \inf_y F_{Q_k}(y; x_k) + \varepsilon_k,$$

then [93, Cor. 5.2] guarantees a nearby point $\hat{x}_k$ with

$$|\nabla F|(\hat{x}_k) \ \le \ \sqrt{12\,\eta_k\,\varepsilon_k} \ + \ 3\eta_k\|x_{k+1} - x_k\|, \qquad \eta_k := L_L + \lambda_{\max}(Q_k).$$

Consequently, if

$$\varepsilon_k \to 0 \qquad \text{and} \qquad \|x_{k+1} - x_k\| \to 0$$

along accepted steps, the stationarity conclusions remain valid.

Moreover, the *model decrease*

$$\Delta_k := F(x_k) - \inf_y F_{Q_k}(y; x_k)$$

is a practical stopping criterion: [93, Cor. 5.5] yields, for any $\eta \ge \eta_k$, the existence of $\hat{x}$ with

$$|\nabla F|(\hat{x}) \ \le \ \sqrt{12\,\eta\,\Delta_k}.$$

4.4 Numerical Experiments

We illustrate the practical behavior of prox-convex through three progressively richer examples. Section 4.4.1 uses a one-dimensional problem to visualize how the two decomposition strategies, $h(C(x))$ and $s(R(x))$, produce different convex models and why the $s(R(x))$ formulation yields a tighter surrogate. Section 4.4.2 treats a log-of-quadratic objective that admits *both* an $h(C(x))$ and an $s(R(x))$ decomposition, and compares prox-gradient, prox-linear, and prox-convex with and without adaptive proximal weights. Section 4.4.3 presents the main benchmark, a disjunctive feasibility problem in which either r_1 or r_2 is required to be nonpositive and whose GMSR parameterization reduces to a product-of-quadratics objective, together with an extensive study of convergence behavior, sensitivity to initialization and proximal weight, and dependence on the condition number of the inner maps.

Throughout, all methods share identical trust-region parameters $(\alpha_1, \alpha_2, \nu_{\text{inc}}, \nu_{\text{dec}})$ unless explicitly noted. The initial proximal weight μ_0 is specified separately for each experiment. Termination is declared when the absolute objective decrease $|F(x_k) - F(x_{k+1})|$ falls below 10^{-8} or the step size $\|x_{k+1} - x_k\|$ falls below 10^{-10}. Subproblems are solved with QOCO [65]; all timing measurements are wall-clock times on a single core. The implementation is available at `https://github.com/UW-ACL/prox-convex`.

4.4.1 Model quality: one-dimensional illustration

We begin with the one-dimensional example

$$f(x) \;=\; \left|\tfrac{1}{2}x^2 + x\right| \;=\; |x|\left|1 + \tfrac{x}{2}\right|, \tag{4.14}$$

whose global minimizers are $x = 0$ and $x = -2$, both satisfying $f(x) = 0$. This example is adapted from [93], where it is used to illustrate that, for nonsmooth problems, convergence to a local minimum does not necessarily require the gradient to converge to zero.

The function admits two natural decompositions.

- $h(C(x))$ **(prox-linear):** set $h(z) = |z|$ and $C(x) = \frac{1}{2}x^2 + x$. Here h is convex and kept exact, while the smooth inner map C is linearized. The resulting convex model at x_0 is

$$m^{\mathrm{pl}}(x;x_0) = \Big|C(x_0) + C'(x_0)(x - x_0)\Big|,$$

that is, the absolute value of an affine approximation of the parabola.

- $s(R(x))$ **(prox-convex):** set

  $$s(z_1, z_2) = z_1 z_2, \qquad R(x) = \Big(|x|,\, |1 + \tfrac{x}{2}|\Big).$$

  In this representation, the smooth outer map s is linearized, while the convex inner components are kept exact. Since

  $$\nabla s(R(x_0)) = \Big(|1 + \tfrac{x_0}{2}|,\, |x_0|\Big) \geq 0,$$

  the convex model becomes

  $$m^{\mathrm{pcx}}(x;x_0) = |1 + \tfrac{x_0}{2}|\,|x| + |x_0|\,|1 + \tfrac{x}{2}| - |x_0|\,|1 + \tfrac{x_0}{2}|.$$

Figure 4.1 compares these two convex models at two representative expansion points. The prox-linear model m^{pl} is built from a single tangent line to the smooth parabola $C(x) = \frac{1}{2}x^2 + x$, so it only captures the behavior of the objective through that local affine approximation and yields a relatively loose surrogate near the kinks at $x = 0$ and $x = -2$. By contrast, the prox-convex model m^{pcx} preserves the full piecewise-linear geometry of $|x|$ and $|1 + x/2|$, and therefore automatically captures both kinks regardless of the expansion point. As a result, the prox-convex model is uniformly tighter in this example.

This tightness has a particularly simple consequence here. For the $s(R(x))$ decomposition, the minimizer of the prox-convex subproblem is also a minimizer of the original objective. Therefore, once the problem is represented in this form, convergence can be achieved in a single iteration. This makes the example a useful visual illustration of the central idea of the paper: preserving convex inner structure can yield a much more informative local model than fully linearizing the composite objective.

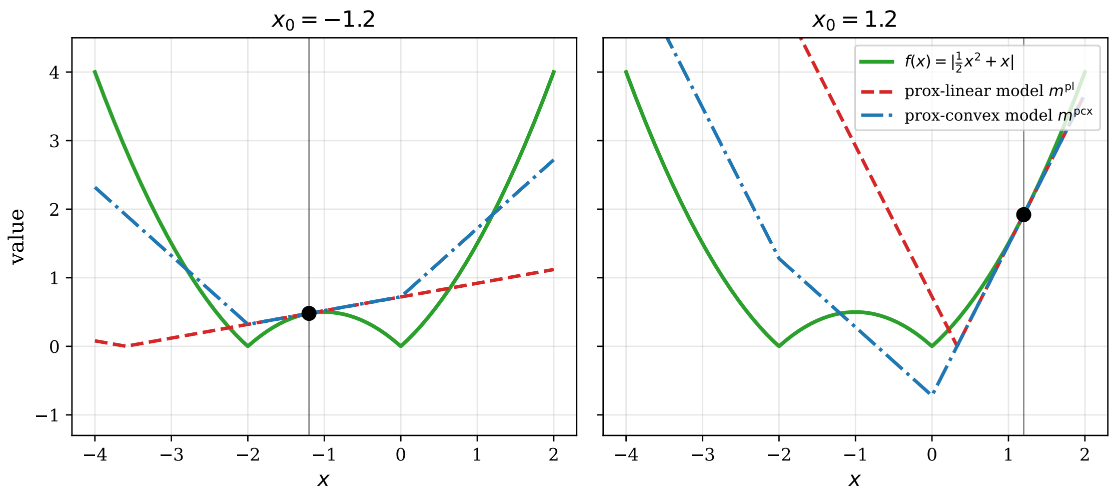


Figure 4.1: Convex models for the one-dimensional problem (4.14) at two expansion points $x_0 \in \{-1.2, 1.2\}$. *Solid*: true objective $f(x)$. *Dashed*: prox-linear model m^{pl}, which linearizes $C(x) = \frac{1}{2}x^2 + x$ and keeps $h(z) = |z|$ exact. *Dash-dotted*: prox-convex model m^{pcx}, which linearizes $s(z_1, z_2) = z_1 z_2$ and keeps $R(x) = \big(|x|, |1 + x/2|\big)$ exact. The prox-convex model preserves both kinks and tracks the true objective more tightly at both expansion points.

4.4.2 Dual decomposition of a log-of-quadratic

We next consider an objective that admits *both* decompositions simultaneously, allowing a direct comparison between the prox-linear and prox-convex subproblems on identical ground. Define

$$F(x) \;=\; \log\!\Big(Q(x)\Big) + 8, \qquad Q(x) := \|A(x - \bar{x})\|^2 + c, \tag{4.15}$$

where $A = \mathrm{diag}(\sqrt{5}, 1)$, $\bar{x} = (5, 2)$, and $c = e^{-8}$. The function Q is a strictly convex quadratic, so F is smooth, nonconvex (concave composite of convex), with its unique global minimizer at $x^\star = \bar{x}$ where $F^\star = \log(c) + 8 = 0$.

Two decompositions.

- **Prox-linear** ($h(C)$ form): set $h(z) = -\log(z) + 8$ (convex) and $C(x) = 1/Q(x)$ (smooth, nonconvex). The model linearizes C; the high curvature $|C''| \propto Q^{-3}$ near the optimum makes the linearization increasingly poor.

- **Prox-convex** ($s(R)$ form): set $s(z) = \log(z) + 8$ (smooth, concave) and $R(x) = Q(x)$ (convex quadratic). The model linearizes s; since s is concave increasing and R is convex, the linearized model is a *global majorizer* of F:

$$F(x) = s(R(x)) \;\leq\; s(R(x_k)) + \nabla s(R(x_k))\Big(R(x) - R(x_k)\Big) = F(x; x_k). \tag{4.16}$$

Non-adaptive regime. Because (4.16) holds globally, the prox-convex model guarantees descent for *any* $\mu > 0$, no matter how small; moreover, since the model's unique minimizer approaches $x^\star = \bar{x}$ as $\mu \to 0$, prox-convex converges in a *single accepted iteration* from any starting point when μ is chosen sufficiently small.

In sharp contrast, prox-linear and prox-gradient become increasingly conservative as the iterates approach the minimizer. Near $x^\star$, the reciprocal map $C(x) = 1/Q(x)$ becomes highly curved, so convergence in a non-adaptive scheme requires a very large fixed proximal weight in order to keep the iterates stable over the entire trajectory. In our experiments, the smallest fixed values that still yielded convergence from $x_0 = (1.5, -1.5)$ were $\mu^{\mathrm{pg}} = 2260$ for prox-gradient and $\mu^{\mathrm{pl}} = 2110$ for prox-linear. Even then, the required iteration counts were extremely large: approximately 27,700 for prox-gradient and 25,900 for prox-linear to reach $|F(x_k) - F^\star| < 10^{-7}$. Prox-convex avoids this issue by keeping the quadratic map Q exact and linearizing only the outer logarithm. For this example, the minimizer of the prox-convex subproblem coincides with the minimizer of the original objective, and convergence can therefore be obtained in a single iteration.

Adaptive regime. With the adaptive trust-region scheme of Algorithm 1 enabled ($\alpha_1 = 0.1$, $\alpha_2 = 0.75$, $\nu_{\mathrm{inc}} = \nu_{\mathrm{dec}} = 1.2$), all three methods converge reliably from $x_0 = (1.5, -1.5)$. For prox-gradient and prox-linear, we use $\mu_0 = 0.1$. For prox-convex, we use $\mu_0 = 10^{-8}$, which is safe in this example because the linearized $s(R(x))$ model is a global majorizer of the objective. Figure 4.2 compares their convergence history and Figure 4.3 shows the iterate trajectories. Prox-convex converges in just 2 accepted iterations (with 2 total subproblem solves), reaching $F \approx 1.7 \times 10^{-14}$ in 6.0 ms. Prox-gradient requires 37 accepted iterations

(96 solves, 66.0 ms) and prox-linear 55 accepted iterations (93 solves, 64.1 ms). Prox-convex maintains the smallest proximal weight μ_k throughout, while prox-linear has more accepted iterations than prox-gradient in this instance because its inner map $C(x) = 1/Q(x)$ becomes highly curved near the optimum, which leads to more rejected trials.

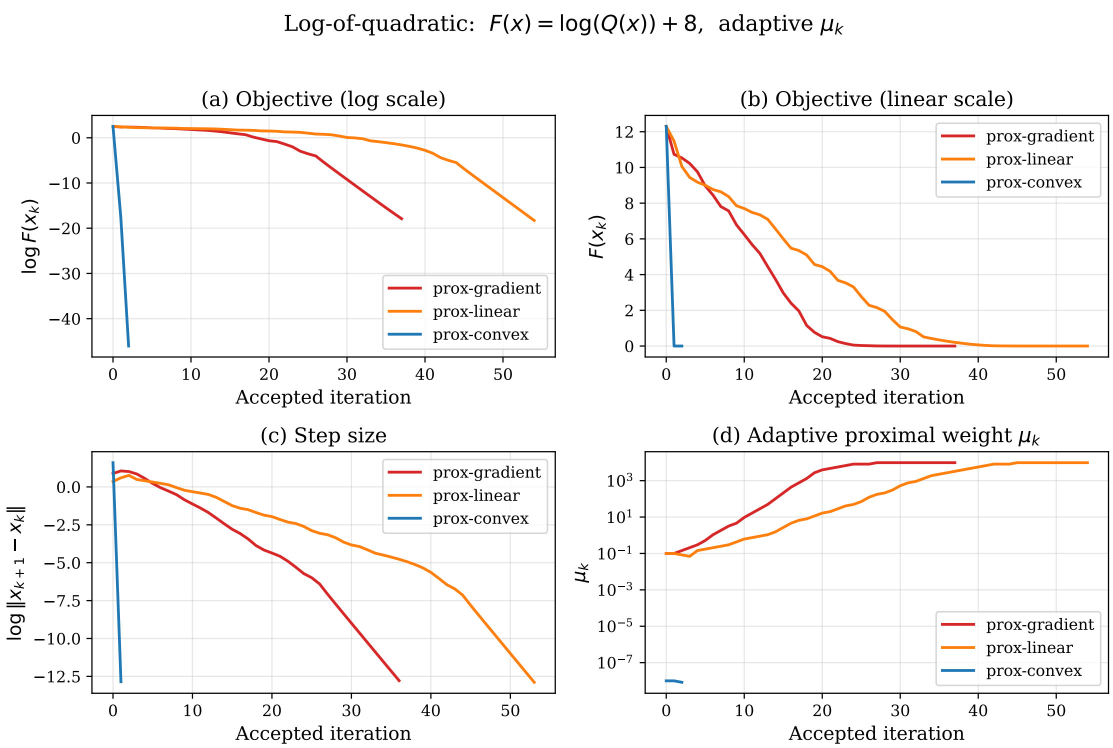


Figure 4.2: Adaptive convergence on the log-of-quadratic problem (4.15) from $x_0 = (1.5, -1.5)$. Prox-gradient and prox-linear use $\mu_0 = 0.1$, whereas prox-convex uses $\mu_0 = 10^{-8}$. **(a)** Log objective $\log F(x_k)$ vs. accepted iteration. **(b)** Objective $F(x_k)$ (linear scale). **(c)** Log step size $\log \|x_{k+1} - x_k\|$. **(d)** Adaptive proximal weight μ_k. All three methods use identical trust-region parameters. Prox-convex (blue) converges in 2 accepted iterations, while prox-gradient (red, 37 iters) and prox-linear (orange, 55 iters) require substantially more iterations and higher proximal weights.

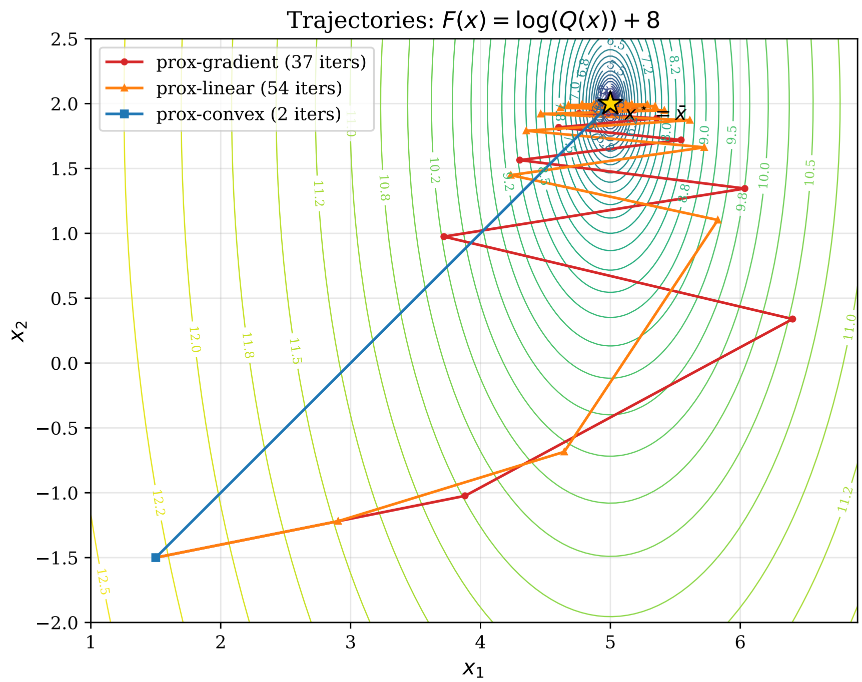


Figure 4.3: Iterate trajectories over the level sets of $F(x) = \log(Q(x)) + 8$. The gold star marks the unique minimizer $x^\star = \bar{x} = (5, 2)$. Prox-convex (blue squares, 2 iters) takes a nearly direct path to $x^\star$, while prox-gradient (red circles, 37 iters) and prox-linear (orange triangles, 55 iters) take many small steps, oscillating as the adaptive proximal weight adjusts to the rapidly increasing curvature near $x^\star$.

4.4.3 Product of quadratics

We now turn to the main benchmark, a disjunctive feasibility problem in which at least one of two nonnegative quadratic functions is required to be nonpositive. Via the GMSR parameterization of disjunction, this condition is represented by requiring a product-of-quadratics objective to vanish. Specifically, we consider

$$F(x) \;=\; r_1(x)\, r_2(x), \qquad r_i(x) := \tfrac{1}{2}(x - c_i)^\top P_i (x - c_i), \tag{4.17}$$

with centers $c_1 = (5, 2)$, $c_2 = (2, 1)$ and matrices $P_1 = \mathrm{diag}(5,\, 5/\kappa)$, $P_2 = \mathrm{diag}(5/\kappa,\, 5)$, where $\kappa \geq 1$ is the condition number. Because each $r_i(x) \geq 0$, the condition $F(x) = 0$ is equivalent to requiring that at least one of $r_1(x)$ or $r_2(x)$ vanish, and hence the global minimizers are $x = c_1$ and $x = c_2$.

Decomposition. For this benchmark, we do not use a nontrivial $h(C(x))$ representation. Instead, the natural structure-preserving formulation is the $s(R)$ decomposition

$$s(z_1, z_2) = z_1 z_2, \qquad R(x) = \big(r_1(x),\, r_2(x)\big).$$

Since

$$\nabla s(R(x_k)) = (r_2(x_k),\, r_1(x_k)) \geq 0,$$

because both quadratic channels are nonnegative, the prox-convex model takes the form

$$F(x; x_k) \;=\; r_2(x_k)\, r_1(x) \;+\; r_1(x_k)\, r_2(x) \;-\; r_1(x_k)\, r_2(x_k). \tag{4.18}$$

This model is convex, being a nonnegative weighted sum of convex quadratics minus a constant, and it preserves the full quadratic geometry of both r_1 and r_2. By contrast, the fully linearized baseline replaces F by a single tangent hyperplane and discards this curvature information. Accordingly, for this problem the main practical comparison is between prox-gradient, which fully linearizes the objective, and prox-convex, which preserves the convex quadratic channels.

Model quality and conditioning. For $s(z) = z_1 z_2$, the outer smoothness is fixed, with Hessian

$$\nabla^2 s(z) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

so the curvature of the outer map does not change with κ. What changes with κ is the anisotropy of the quadratic channels r_1 and r_2: as κ increases, their level sets become increasingly elongated and the geometry of the product objective becomes more ill-conditioned. In this regime, the difference between the two local models becomes more pronounced. The fully linearized prox-gradient model ignores the exact quadratic structure and therefore becomes increasingly conservative in practice, typically requiring more accepted iterations and larger proximal weights. By contrast, prox-convex retains the exact quadratic channels in the subproblem, so the local model continues to reflect the geometry of the original objective much more faithfully. This is precisely the effect observed in the condition-number sweep: as κ increases, the iteration advantage of prox-convex grows substantially, while its accepted iteration count remains comparatively stable.

Setup. All experiments use Algorithm 1 with $\alpha_1 = 0.1$, $\alpha_2 = 0.75$, $\nu_{\text{inc}} = \nu_{\text{dec}} = 1.5$.

Single-run comparison

From the starting point $x_0 = (0, 0)$ with $\kappa = 10$, Figure 4.4 shows the convergence history and Figure 4.5 displays the iterate trajectories over the level sets of F. Prox-convex converges in 7 accepted iterations (7 subproblem solves, 8.8 ms) while prox-gradient requires 43 accepted iterations (72 solves, 53.5 ms)—a 6.1× reduction in iterations and 6.1× speedup in wall-clock time. Prox-convex maintains a "consistently" lower proximal weight μ_k throughout. The trajectory plot reveals that prox-convex follows a more direct path toward $c_2 = (2, 1)$, the nearer minimizer, while prox-gradient takes many small, conservative steps.

Robustness across initializations

To verify that the advantage of prox-convex is not an artifact of a particular starting point, we run both methods from $N = 50$ random starting points drawn uniformly from $[-2, 8] \times [-2, 5]$, all with $\kappa = 10$ and $\mu_0 = 10$. All 50 runs of both methods converge within the iteration limit. Figure 4.6 shows the distribution of accepted iteration counts and wall-clock solve times. The median iteration count is 46.5 for prox-gradient vs. 7 for prox-convex (a 6.69× ratio), and the median wall-clock time is 52.1 ms vs. 9.4 ms (a 5.39× ratio). Prox-convex exhibits both a lower median and a tighter interquartile range in both metrics, indicating faster and more predictable convergence across diverse initializations.

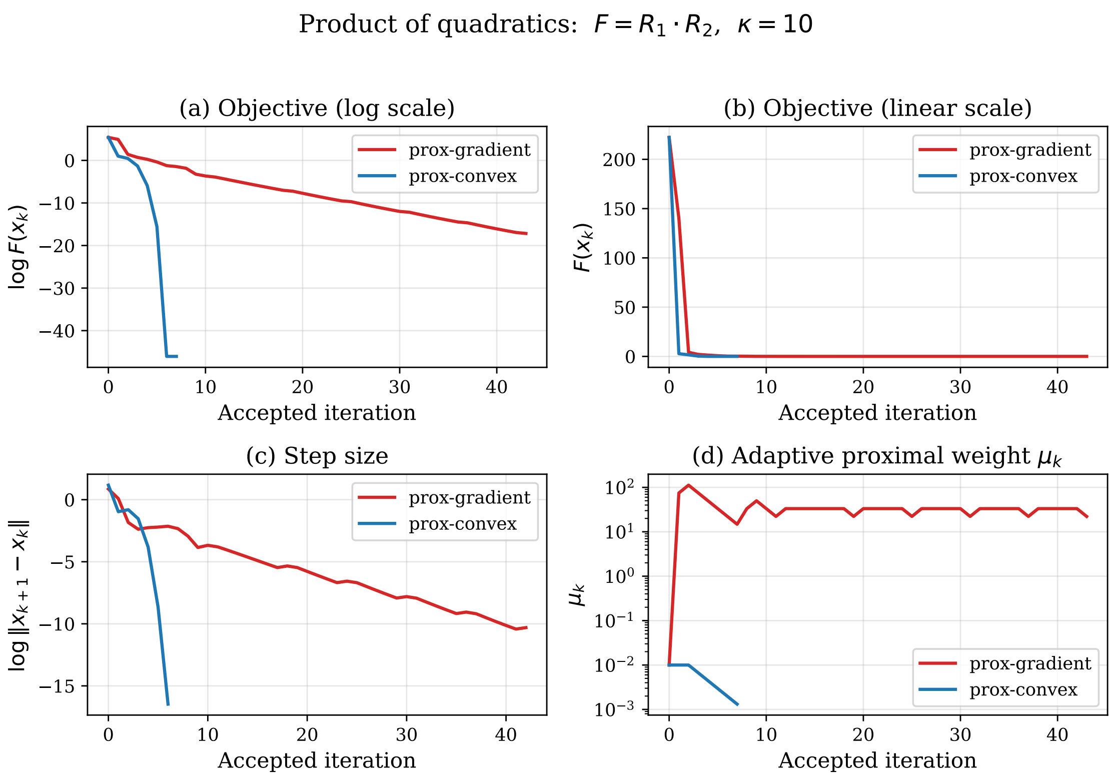


Figure 4.4: Convergence comparison on the product-of-quadratics problem (4.17) with $\kappa = 10$, $x_0 = (0, 0)$, $\mu_0 = 10$. **(a)** Log objective vs. accepted iteration. **(b)** Objective (linear scale). **(c)** Log step size $\log \|x_{k+1} - x_k\|$. **(d)** Adaptive proximal weight μ_k (log scale). Prox-convex (blue, 7 iters) converges faster and operates with lower μ_k than prox-gradient (red, 43 iters), reflecting its tighter model that preserves the quadratic geometry of r_1 and r_2.

Effect of inner curvature

We next investigate how the relative performance changes as the quadratic channels become more anisotropic. To this end, we sweep the condition number

$$\kappa \in \{1, 2, 5, 10, 20, 50, 100, 200, 500\},$$

and, for each value of κ, run both methods from $N = 50$ common random starting points. Table 4.2 and Figure 4.7 report the results.

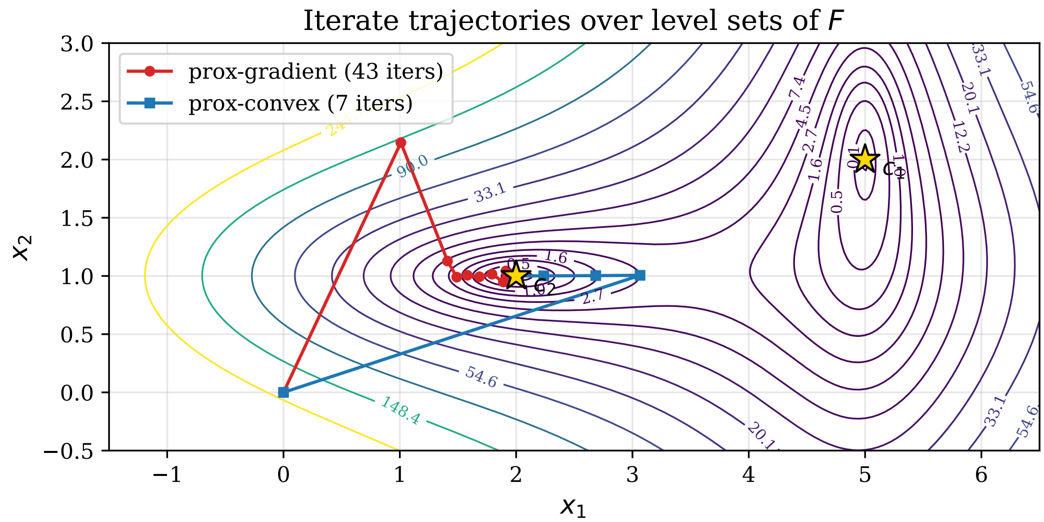


Figure 4.5: Iterate trajectories over the level sets of $F(x) = r_1(x)r_2(x)$ with $\kappa = 10$. Gold stars mark the two global minimizers $c_1 = (5, 2)$ and $c_2 = (2, 1)$. Prox-convex (blue squares, 7 iters) reaches c_2 via a direct path, while prox-gradient (red circles, 43 iters) takes many small steps. The prox-convex subproblem (4.18) "sees" both ellipsoidal sublevel sets simultaneously, whereas prox-gradient sees only a single tangent plane.

The empirical trend is clear: as κ increases, the advantage of prox-convex grows substantially. The median iteration ratio (prox-gradient / prox-convex) rises from $2.31\times$ at $\kappa = 1$ to $83.33\times$ at $\kappa = 500$. A similar effect is observed in wall-clock time, where the median time ratio grows from $3.38\times$ to $46.38\times$ over the same range. At the same time, the accepted iteration count for prox-convex remains remarkably stable, with medians between 6 and 14 across the entire sweep, whereas the prox-gradient median grows steadily from 14 to 500.

This behavior is consistent with the geometry of the benchmark. As κ increases, the quadratic channels become increasingly elongated, and the fully linearized prox-gradient model becomes much more conservative in practice. By contrast, prox-convex retains the exact quadratic structure of both channels in the subproblem, so its local model continues to reflect the geometry of the original objective much more faithfully. This effect becomes particularly

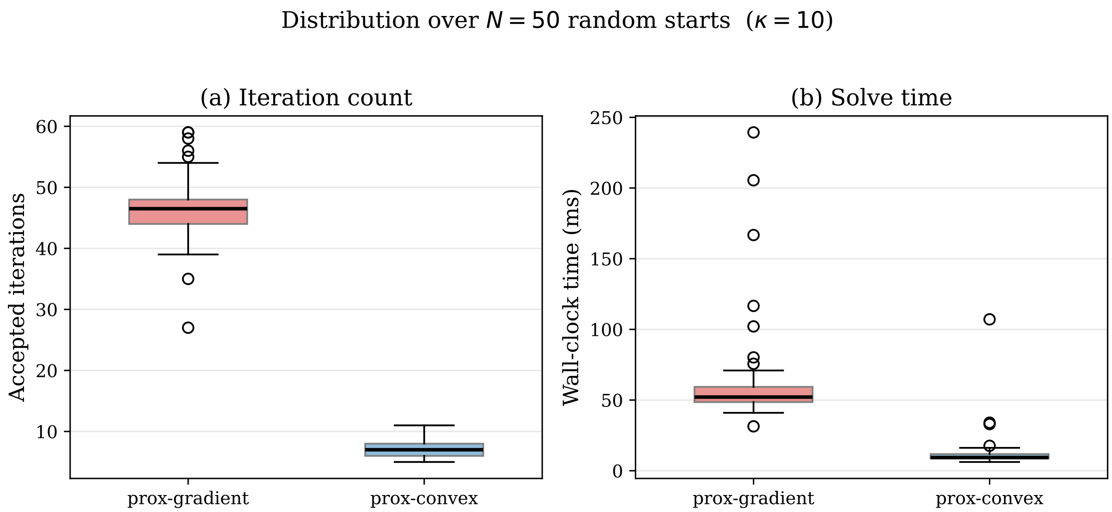


Figure 4.6: Distribution over $N = 50$ random starting points ($\kappa = 10$, $\mu_0 = 10$). Boxes span the interquartile range (25th–75th percentile); the interior line is the median; whiskers extend to 1.5× IQR. **(a)** Accepted iterations: median 46.5 (prox-gradient) vs. 7 (prox-convex). **(b)** Wall-clock time: median 52.1 ms vs. 9.4 ms. Prox-convex has a lower median and narrower spread in both metrics, confirming that the structural advantage persists across initializations.

pronounced for large κ: at $\kappa = 200$ and $\kappa = 500$, only 33/50 and 21/50 prox-gradient runs converge within the iteration limit, whereas all 50/50 prox-convex runs converge throughout.

Sensitivity to the initial proximal weight

We next examine the sensitivity of the adaptive trust-region scheme to the initial proximal weight μ_0. To do so, we sweep

$$\mu_0 \in \{10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\},$$

and, for each value of μ_0, run both methods from $N = 20$ common random starting points with $\kappa = 10$. Table 4.3 reports the median accepted iterations, total subproblem solves, and wall-clock time.

Table 4.2: Median accepted iterations and wall-clock time over $N = 50$ random starting points for each condition number κ. All prox-convex runs converge; prox-gradient convergence rates are reported in the last column.

	prox-gradient		prox-convex		ratio		
κ	iters	time (ms)	iters	time (ms)	iters	time	pg conv.
1	14	27.9	6	8.1	2.31	3.38	50/50
2	19	28.2	6	7.6	3.07	3.82	50/50
5	27	35.7	6	9.0	4.00	4.12	50/50
10	47	48.8	7	9.0	6.46	5.49	50/50
20	83	81.0	8	10.2	9.88	8.20	50/50
50	176	159.3	14	13.7	14.82	12.47	50/50
100	258	225.8	10	10.5	27.20	20.60	50/50
200	408	351.9	6	8.5	60.83	36.99	33/50
500	500	396.4	6	7.6	83.33	46.38	21/50

The adaptive mechanism makes both methods reasonably robust to the choice of μ_0, but the difference between them remains substantial across the entire sweep. For prox-gradient, the median accepted iteration count stays in the range 44–48, while for prox-convex it ranges from 7 to 22. The corresponding iteration ratios vary from $6.86\times$ at $\mu_0 = 10^{-2}$ to $2.13\times$ at $\mu_0 = 10^3$. Wall-clock time shows the same qualitative pattern, with median time ratios between $2.13\times$ and $5.78\times$.

The trends are also intuitive. When μ_0 is too small, the methods incur a few extra rejected trials before the trust region expands to a reliable scale. When μ_0 is too large, the initial steps become overly conservative, which increases the accepted iteration count, especially for prox-convex. Nevertheless, the relative advantage of prox-convex persists uniformly across four orders of magnitude of μ_0. This indicates that the observed performance gap is not a fragile consequence of tuning, but rather a structural consequence of preserving the exact

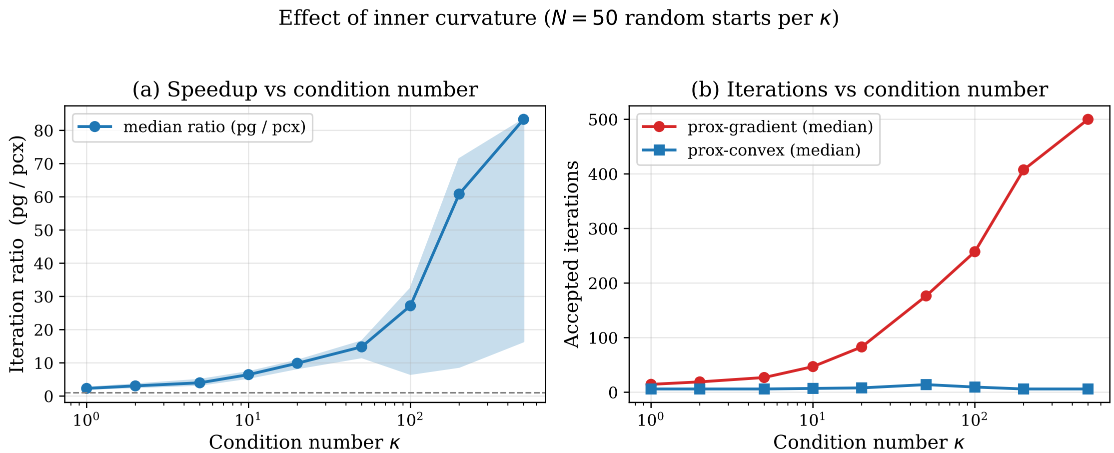


Figure 4.7: Effect of increasing anisotropy on relative performance. **(a)** Median iteration ratio (prox-gradient / prox-convex) versus condition number κ; the shaded band shows the interquartile range over $N = 50$ random starts. The ratio grows rapidly as κ increases. **(b)** Absolute median iteration counts. Prox-convex iterations remain comparatively stable across the sweep, while prox-gradient iterations grow sharply and eventually hit the iteration limit on a significant fraction of runs.

quadratic channels in the local model.

4.5 Supplementary technical results

The results in this section collect technical comparison and Hessian-augmented model bounds used to justify remarks in the main text.

4.5.1 Model errors

In composite problems, the effective level of linearization depends on the roles of the outer and inner maps. If the outer map is smooth and the inner components are convex, linearizing the *outer* layer preserves convexity; if the outer is convex and the inner is smooth but nonconvex, linearizing the *inner* layer is natural. When *both* s and the r_i are convex

Table 4.3: Sensitivity to the initial proximal weight μ_0 ($\kappa = 10$, $N = 20$ random starts per μ_0). All values are medians. "Iters" denotes accepted iterations, and "solves" includes rejected trials.

	prox-gradient			prox-convex			ratio	
μ_0	iters	solves	time (ms)	iters	solves	time (ms)	iters	time
10^{-2}	48	74	54.2	7	7	9.4	6.86	5.78
10^{-1}	44	66	46.9	8	8	9.7	5.93	4.82
10^{0}	45	66	47.8	10	10	11.1	4.50	4.30
10^{1}	47	58	41.7	14	14	14.2	3.48	2.93
10^{2}	46	52	39.3	17	17	15.7	2.71	2.50
10^{3}	48	62	44.0	22	22	20.6	2.13	2.13

and $\mathcal{C}^1$-smooth, the choice is less obvious. We therefore compare three models: (i) *full* linearization F^{all}, (ii) *inner-only* linearization F^{in}, and (iii) *outer-only* linearization F^{out}, and derive clean error bounds that reveal the dominant constants.

To compare the three linearization levels on equal footing, we strengthen Assumption 4.3 for this subsection as follows: in addition to the original standing assumptions, we assume

$$\|\nabla s(y)\| \le L_s$$

on the relevant image of the level set, and that every r_i is $\mathcal{C}^1$ with β_{r_i}-Lipschitz gradient. We then write

$$\beta_R := \left(\sum_{i=1}^{n} \beta_{r_i}^2\right)^{1/2}.$$

Proposition 4.22 (Modeling error bounds)**.** *Fix x_k, write*

$$d_x := x - x_k, \qquad R_k := R(x_k), \qquad J_k := J_R(x_k), \qquad \Delta_R := R(x) - R_k.$$

Define

$$F^{\text{all}}(x; x_k) := s(R_k) + \nabla s(R_k)^\top (J_k d_x),$$
$$F^{\text{in}}(x; x_k) := s(R_k + J_k d_x), \qquad\qquad E^{(\cdot)}(x; x_k) := s(R(x)) - F^{(\cdot)}(x; x_k).$$
$$F^{\text{out}}(x; x_k) := s(R_k) + \nabla s(R_k)^\top (\Delta_R),$$

Under the assumptions above,

$$|E^{\text{all}}(x; x_k)| \le \left(\frac{\beta_s L_R^2}{2} + \frac{L_s \beta_R}{2} \right) \|d_x\|^2,$$
$$|E^{\text{in}}(x; x_k)| \le \frac{L_s \beta_R}{2} \|d_x\|^2,$$
$$|E^{\text{out}}(x; x_k)| \le \frac{\beta_s L_R^2}{2} \|d_x\|^2.$$

Proof. Let the inner Taylor remainder be

$$e_R(x) := R(x) - \Big(R_k + J_k d_x \Big).$$

By componentwise β_{r_i}-smoothness and stacking,

$$\|e_R(x)\| \le \frac{\beta_R}{2} \|d_x\|^2. \tag{4.19}$$

Lipschitz continuity of R yields

$$\|\Delta_R\| \le L_R \|d_x\|, \qquad \|J_k d_x\| \le L_R \|d_x\|. \tag{4.20}$$

For s we use the descent lemma and a gradient-bound shift:

$$\Big| s(u) - s(v) - \nabla s(v)^\top (u - v) \Big| \le \frac{\beta_s}{2} \|u - v\|^2, \qquad \Big| s(u) - s(v) \Big| \le L_s \|u - v\|. \tag{4.21}$$

All:

$$\begin{aligned}
|E^{\text{all}}| &= \Big| s(R_k + \Delta_R) - s(R_k) - \nabla s(R_k)^\top (J_k d_x) \Big| \\
&\le \underbrace{\Big| s(R_k + \Delta_R) - s(R_k + J_k d_x) \Big|}_{\le L_s \|e_R(x)\|} + \underbrace{\Big| s(R_k + J_k d_x) - s(R_k) - \nabla s(R_k)^\top (J_k d_x) \Big|}_{\le \frac{\beta_s}{2} \|J_k d_x\|^2} \\
&\le \frac{L_s \beta_R}{2} \|d_x\|^2 + \frac{\beta_s L_R^2}{2} \|d_x\|^2,
\end{aligned}$$

by (4.19), (4.20), and (4.21).

In:

$$|E^{\text{in}}| = \Big|s(R_k + \Delta_R) - s(R_k + J_k d_x)\Big| \le L_s \|e_R(x)\| \le \frac{L_s \beta_R}{2} \|d_x\|^2.$$

Out:

$$|E^{\text{out}}| = \Big|s(R_k + \Delta_R) - s(R_k) - \nabla s(R_k)^\top \Delta_R\Big| \le \frac{\beta_s}{2} \|\Delta_R\|^2 \le \frac{\beta_s L_R^2}{2} \|d_x\|^2.$$

■

The outer-only model is favored when

$$\beta_s L_R^2 \ll L_s \beta_R,$$

that is, under strong inner curvature but gently curved outer geometry, while the inner-only model is preferable in the opposite regime. The full linearization combines both penalties and is typically dominated. In prox-convex, the preserved-structure model is then paired with an adaptive proximal metric and, when tractable, second-order curvature blocks, further tightening local models and improving observed progress.

4.5.2 Bounds for Hessian-augmented models

In this subsection, in addition to Assumption 4.3, we assume the following higher-order smoothness conditions to provide bounds for Hessian-augmented models supporting Section 4.2.3.

Assumption 4.23 (Additional smoothness assumptions)**.** *In the level set $\mathcal{X}_0$, we assume:*

- $h : \mathbb{R}^d \to \mathbb{R}$ *is L_h-Lipschitz and $\mathcal{C}^1$ with β_h-Lipschitz Jacobian;*
- $C : \mathbb{R}^m \to \mathbb{R}^d$ *is L_C-Lipschitz and $\mathcal{C}^2$ with*

$$\|\nabla^2 C_i(u) - \nabla^2 C_i(v)\| \le \gamma_C \|u - v\|;$$

- $s : \mathbb{R}^n \to \mathbb{R}$ *is $\mathcal{C}^2$ with*

$$\|\nabla^2 s(u) - \nabla^2 s(v)\| \le \gamma_s \|u - v\|;$$

- *$R : \mathbb{R}^m \to \mathbb{R}^n$ is $\mathcal{C}^1$ with β_R-Lipschitz Jacobian.*

Proposition 4.24 (Inner curvature cancellation)**.** *Let*

$$J_k := J_C(x_k), \qquad y := \nabla h(C(x_k)).$$

Define

$$H_{C,k} := \sum_{i=1}^{d} y_i \, \nabla^2 C_i(x_k),$$

and the model

$$F^C_{H_{C,k}}(x; x_k) := h\Big(C(x_k) + J_k d\Big) + \frac{1}{2} d^\top H_{C,k} d, \qquad d := x - x_k.$$

Then, for x near x_k,

$$|h(C(x)) - F^C_{H_{C,k}}(x; x_k)| \le M^3_C \|d\|^3 + M^4_C \|d\|^4,$$

where

$$M^3_C = \frac{L_h \gamma_C}{6} + \frac{\beta_h L_C \beta_C}{2}, \qquad M^4_C = \frac{\beta_h \beta_C^2}{8} + \frac{\beta_h L_C \gamma_C}{6}.$$

Proof. Let $q(d) \in \mathbb{R}^d$ with

$$[q(d)]_i = \langle \nabla^2 C_i(x_k) d, d \rangle,$$

and let r_3 be the cubic remainder in

$$C(x_k + d) = C(x_k) + J_k d + \frac{1}{2} q(d) + r_3, \qquad \|q(d)\| \le \beta_C \|d\|^2, \qquad \|r_3\| \le \frac{\gamma_C}{6} \|d\|^3.$$

Let

$$z := C(x_k) + J_k d, \qquad \delta := \frac{1}{2} q(d) + r_3,$$

so that

$$C(x) = z + \delta.$$

By Taylor's theorem with β_h-Lipschitz gradient,

$$h(z + \delta) = h(z) + \langle \nabla h(z), \delta \rangle + \frac{1}{2} \delta^\top \Big(\nabla^2 h(z + \tau \delta) \Big) \delta, \qquad \|\nabla^2 h(z + \tau \delta)\| \le \beta_h.$$

Add and subtract $\nabla h(C(x_k))$:

$$h(z+\delta) - h(z) = \underbrace{\langle \nabla h(C(x_k)), \delta\rangle}_{=\frac{1}{2}\langle y, q(d)\rangle + \langle y, r_3\rangle} + \underbrace{\langle \nabla h(z) - \nabla h(C(x_k)), \delta\rangle}_{\mathrm{I}} + \underbrace{\frac{1}{2}\delta^\top \nabla^2 h(\cdot)\,\delta}_{\mathrm{II}}.$$

Since

$$\frac{1}{2} d^\top H_{C,k} d = \frac{1}{2}\langle y, q(d)\rangle,$$

subtracting $F^C_{H_{C,k}}$ leaves

$$h(C(x)) - F^C_{H_{C,k}}(x; x_k) = \langle y, r_3\rangle + \mathrm{I} + \mathrm{II}.$$

Bounds:

$$\|y\| = \|\nabla h(C(x_k))\| \le L_h$$

gives

$$|\langle y, r_3\rangle| \le \frac{L_h \gamma_C}{6}\|d\|^3.$$

Moreover,

$$\|\nabla h(z) - \nabla h(C(x_k))\| \le \beta_h \|z - C(x_k)\| = \beta_h \|J_k d\| \le \beta_h L_C \|d\|,$$

and

$$\|\delta\| \le \frac{1}{2}\|q(d)\| + \|r_3\| \le \frac{1}{2}\beta_C \|d\|^2 + \frac{\gamma_C}{6}\|d\|^3,$$

hence

$$|\mathrm{I}| \le \frac{\beta_h L_C \beta_C}{2}\|d\|^3 + \frac{\beta_h L_C \gamma_C}{6}\|d\|^4.$$

Finally,

$$|\mathrm{II}| \le \frac{1}{2}\beta_h \|\delta\|^2 \le \frac{\beta_h \beta_C^2}{8}\|d\|^4 + O(\|d\|^5).$$

Collecting the cubic and quartic terms and absorbing higher-order terms gives the stated bound. ■

Proposition 4.25 (Outer curvature cancellation). *Fix x_k and write*

$$d := x - x_k, \qquad \Delta := R(x) - R(x_k), \qquad J_k := J_R(x_k), \qquad e_R(x) := R(x) - R(x_k) - J_k d.$$

Set

$$\tilde{H}_{s,k} := \nabla^2 s(R(x_k)), \qquad H_{s,k} := J_k^\top \tilde{H}_{s,k} J_k,$$

and define

$$F^s_{H_{s,k}}(x; x_k) := s(R(x_k)) + \nabla s(R(x_k))^\top \Big(R(x) - R(x_k)\Big) + \frac{1}{2} d^\top H_{s,k} d.$$

Then, for x near x_k,

$$|s(R(x)) - F^s_{H_{s,k}}(x; x_k)| \le M_s^3 \|d\|^3 + M_s^4 \|d\|^4,$$

where

$$M_s^3 = \frac{\beta_s L_R \beta_R}{2} + \frac{\gamma_s L_R^3}{6}, \qquad M_s^4 = \frac{\beta_s \beta_R^2}{8}.$$

Proof. Apply Taylor's theorem with third-order remainder to s at $R(x_k)$:

$$s(R(x)) = s(R(x_k)) + \nabla s(R(x_k))^\top \Delta + \frac{1}{2} \Delta^\top \tilde{H}_{s,k} \Delta + R_3,$$

with

$$|R_3| \le \frac{\gamma_s}{6} \|\Delta\|^3.$$

With

$$\Delta = J_k d + e, \qquad e := e_R(x),$$

we have

$$\frac{1}{2} \Delta^\top \tilde{H}_{s,k} \Delta = \frac{1}{2} d^\top H_{s,k} d + (J_k d)^\top \tilde{H}_{s,k} e + \frac{1}{2} e^\top \tilde{H}_{s,k} e.$$

Subtracting $F^s_{H_{s,k}}$ yields

$$s(R(x)) - F^s_{H_{s,k}}(x; x_k) = (J_k d)^\top \tilde{H}_{s,k} e + \frac{1}{2} e^\top \tilde{H}_{s,k} e + R_3.$$

Using

$$\|\tilde{H}_{s,k}\| \le \beta_s$$

gives

$$\pm (J_k d)^\top \tilde{H}_{s,k} e \le \beta_s \|J_k d\| \, \|e\|, \qquad \pm \frac{1}{2} e^\top \tilde{H}_{s,k} e \le \frac{1}{2} \beta_s \|e\|^2,$$

while

$$|R_3| \le \frac{\gamma_s}{6}\|\Delta\|^3.$$

Thus

$$|s(R(x)) - F^s_{H_{s,k}}(x;x_k)| \le \beta_s\|J_k d\|\,\|e_R(x)\| + \frac{1}{2}\beta_s\|e_R(x)\|^2 + \frac{\gamma_s}{6}\|\Delta\|^3.$$

Substituting the auxiliary estimates

$$\|J_k d\| \le L_R\|d\|, \qquad \|e_R(x)\| \le \frac{\beta_R}{2}\|d\|^2, \qquad \|\Delta\| \le L_R\|d\|$$

yields the claimed bound. ■

Corollary 4.26 (Projection after summation)**.** *Adopt Assumption 4.23 and the notation of Propositions 4.24 and 4.25. Let*

$$H_k := H_{C,k} + H_{s,k}, \qquad H_k^+ := \Pi_{\mathbb{S}_+}(H_k), \qquad H_k^- := H_k^+ - H_k \succeq 0.$$

Define the joint Hessian-augmented model

$$\begin{aligned} F_{H_k^+}(x;x_k) := \;& g(x) + \underbrace{h\Big(C(x_k) + J_C(x_k)(x - x_k)\Big)}_{\textit{inner first order}} \\ & + \underbrace{s(R(x_k)) + \nabla s(R(x_k))^\top\Big(R(x) - R(x_k)\Big)}_{\textit{outer first order}} \\ & + \frac{1}{2}(x - x_k)^\top H_k^+(x - x_k). \end{aligned}$$

Then, for x near x_k and $d := x - x_k$,

$$\begin{aligned} &\text{(Upper bound)} \quad F(x) - F_{H_k^+}(x;x_k) \le \Big(M_s^3 + M_C^3\Big)\|d\|^3 + \Big(M_s^4 + M_C^4\Big)\|d\|^4, \\ &\text{(Lower bound)} \quad F(x) - F_{H_k^+}(x;x_k) \ge -\frac{1}{2}d^\top H_k^- d - \Big(M_s^3 + M_C^3\Big)\|d\|^3 - \Big(M_s^4 + M_C^4\Big)\|d\|^4, \end{aligned}$$

where the block constants M_s^3, M_s^4 and M_C^3, M_C^4 are those in Propositions 4.24 and 4.25. Equivalently, using

$$d^\top H_k^- d \le \|H_k^-\|\,\|d\|^2,$$

the lower bound can be written with $\|H_k^-\|\,\|d\|^2/2$.

Proof. Apply the inner and outer curvature expansions of Propositions 4.24 and 4.25 at x_k and add the resulting identities. The exact second-order terms sum to

$$\frac{1}{2}d^\top (H_{C,k} + H_{s,k})d = \frac{1}{2}d^\top H_k d.$$

Subtracting the model's quadratic

$$\frac{1}{2}d^\top H_k^+ d$$

leaves

$$-\frac{1}{2}d^\top (H_k^+ - H_k)d = -\frac{1}{2}d^\top H_k^- d,$$

which is the sole second-order projection gap. The remaining terms are exactly the cubic and quartic remainders already bounded in the two propositions, and they add linearly, yielding the stated upper and lower bounds. Using

$$d^\top H_k^- d \le \|H_k^-\| \, \|d\|^2$$

gives the norm form. ■

Chapter 5

TRAJECTORY OPTIMIZATION WITH MISSION-LEVEL SPECIFICATIONS

5.1 Introduction

This chapter demonstrates how the numerical optimal-control foundation of Chapter 2, the mission-specification modeling framework of Chapter 3, and the structure-preserving optimization machinery developed in Chapter 4 come together in trajectory optimization problems with mission-level specifications. We begin with simple illustrative CT-STL examples that isolate the `always`, `eventually`, and `until` operators. The later sections of the chapter consider richer trajectory generation problems, including receding-horizon obstacle avoidance, perception-constrained motion planning, rocket landing with compound state-triggered constraints, and quadrotor flight with CT-STL specifications.

5.2 Illustrative CT-STL trajectory optimization examples

This section presents three fixed-final-time double-integrator examples constructed to illustrate the `always`, `eventually`, and `until` operators in isolation. In all cases, the primary objective is specification satisfaction. To keep the presentation focused, the exact code-level scaling and smoothing constants used in the implementation are omitted here.

In our implementation, `JAX` is used to integrate the augmented dynamics and to compute the first-order derivative information of the smooth residual maps required for linearization [48]. Each convex subproblem is modeled in `CVXPY` [81] and solved using `QOCO` [65]. The implementation is available at `https://github.com/UW-ACL/TrajOpt_CT-STL`.

Remark 5.1 (Satisfaction-oriented surrogates). In some of the examples below, the objective is only to generate trajectories that satisfy the target specification. Accordingly, the auxiliary-state constructions are chosen to capture the relevant satisfaction mechanism of each operator, rather than to reproduce the full CT-GMSR robustness exactly. We therefore denote the terminal objective quantities by $\widehat{\Gamma}$ rather than by Γ. These surrogates are sign- and satisfaction-oriented, but need not coincide with the full CT-GMSR robustness value.

5.2.1 Double-integrator examples

Base dynamics and common continuous-time path constraints

For the three double-integrator examples, let

$$x := (r, v), \qquad r = (r_x, r_y, r_z), \qquad v = (v_x, v_y, v_z), \qquad u = (u_x, u_y, u_z),$$

with dynamics

$$\dot{r}(t) = v(t), \qquad \dot{v}(t) = u(t) - g_0 e_3, \qquad g_0 = 9.806 \text{ m/s}^2.$$

All three examples are subject to the same continuous-time tilt, thrust, and speed constraints,

$$u_x^2 + u_y^2 \leq (\cos\theta_{\max} u_z)^2, \qquad \|u\|_2 \leq T_{\max}, \qquad \|v\|_2 \leq v_{\max},$$

with $\theta_{\max} = 45°$, $T_{\max} = 1.75\, g_0$, and $v_{\max} = 6$ m/s. Defining the predicates

$$\begin{aligned}
\mu_\theta(u) &:= \Big((\cos\theta_{\max} u_z)^2 - u_x^2 - u_y^2 \geq 0\Big), \\
\mu_T(u) &:= \Big(T_{\max}^2 - u_x^2 - u_y^2 - u_z^2 \geq 0\Big), \\
\mu_v(v) &:= \Big(v_{\max}^2 - v_x^2 - v_y^2 - v_z^2 \geq 0\Big),
\end{aligned}$$

the common path-constraint formula is

$$\varphi_{\text{path}} := \boldsymbol{G}_{[0,t_{\text{f}}]}\Big(\mu_\theta \wedge \mu_T \wedge \mu_v\Big).$$

In the implementation, these continuous-time `always` constraints are enforced through an accumulated path-penalty state η_{p},

$$\dot{\eta}_{\text{p}}(t) = \chi_\theta(u(t)) + \chi_T(u(t)) + \chi_v(v(t)), \qquad \eta_{\text{p}}(0) = 0,$$

where

$$\chi_\theta(u) := \left[(\cos\theta_{\max} u_z)^2 - u_x^2 - u_y^2\right]_-^2,$$
$$\chi_T(u) := \left[T_{\max}^2 - u_x^2 - u_y^2 - u_z^2\right]_-^2,$$
$$\chi_v(v) := \left[v_{\max}^2 - v_x^2 - v_y^2 - v_z^2\right]_-^2,$$

and $[s]_+ := \max(s,0)$, $[s]_- := \min(s,0)$. Hence,

$$\eta_{\mathrm{p}}(t_{\mathrm{f}}) = 0$$

certifies satisfaction of the common continuous-time path constraints.

To avoid repeating the boundary conditions, define

$$\mathcal{X}_{\mathrm{i}}(r_{\mathrm{i}}) := \left\{(r,v) \in \mathbb{R}^6 : r = r_{\mathrm{i}},\ v = 0\right\},$$
$$\mathcal{X}_{\mathrm{f}}(r_{\mathrm{f}}) := \left\{(r,v) \in \mathbb{R}^6 : r = r_{\mathrm{f}},\ v = 0\right\},$$

and let

$$u_{\mathrm{h}} := (0, 0, g_0)$$

denote the gravity-compensating input. Then the common endpoint conditions are

$$x(0) \in \mathcal{X}_{\mathrm{i}}(r_{\mathrm{i}}), \qquad x(t_{\mathrm{f}}) \in \mathcal{X}_{\mathrm{f}}(r_{\mathrm{f}}), \qquad u(0) = u(t_{\mathrm{f}}) = u_{\mathrm{h}}.$$

The three examples below differ only in the additional mission-level requirement imposed on top of the common continuous-time path-constraint formula φ_{path}.

Example 1: `always` obstacle avoidance

The first example uses

$$r_{\mathrm{i}} = (-5, 0, 0), \qquad r_{\mathrm{f}} = (5, 0, 0), \qquad t_{\mathrm{f}} = 7 \text{ s}, \qquad K = 6,$$

and requires the vehicle to always avoid two implication-type obstacle regions. The obstacle specifications are

$$\varphi_{\mathrm{obs},1}(r) := (a_1 \leq r_x \leq b_1 \ \Rightarrow\ r_y \leq c_1),$$
$$\varphi_{\mathrm{obs},2}(r) := (a_2 \leq r_x \leq b_2 \ \Rightarrow\ r_y \geq c_2),$$

with

$$(a_1, b_1, c_1) = (-3.25, -0.25, -2), \qquad (a_2, b_2, c_2) = (0.25, 3.25, 2).$$

Thus, the overall task is

$$\varphi_{\text{alw}} := \varphi_{\text{path}} \wedge \boldsymbol{G}_{[0,t_{\text{f}}]}\Big(\varphi_{\text{obs},1} \wedge \varphi_{\text{obs},2}\Big).$$

We augment the state as

$$x^{\varphi_{\text{alw}}} := (x, \eta_{\text{p}}, \xi_1, \xi_2),$$

where η_{p} handles the common path constraints and ξ_1, ξ_2 accumulate obstacle violations:

$$\begin{aligned}
\dot{\xi}_1(t) &= [a_1 - r_x]_-^2 \, [r_x - b_1]_-^2 \, [c_1 - r_y]_-^2, \quad \xi_1(0) = 0, \\
\dot{\xi}_2(t) &= [a_2 - r_x]_-^2 \, [r_x - b_2]_-^2 \, [r_y - c_2]_-^2, \quad \xi_2(0) = 0.
\end{aligned}$$

We therefore define the terminal surrogate

$$\widehat{\Gamma}_{\text{alw}} := -\Big(w_\eta \eta_{\text{p}}(t_{\text{f}}) + w_1 \xi_1(t_{\text{f}}) + w_2 \xi_2(t_{\text{f}})\Big),$$

with positive weights w_η, w_1, w_2. Maximizing $\widehat{\Gamma}_{\text{alw}}$, or equivalently minimizing its negative, favors trajectories that satisfy both the common path constraints and the obstacle-avoidance requirement.

The resulting continuous-time problem is

$$\boxed{\begin{array}{ll}
\underset{x^{\varphi_{\text{alw}}}(\cdot),u(\cdot)}{\text{maximize}} & \widehat{\Gamma}_{\text{alw}} \\
\text{subject to} & \dot{r}(t) = v(t), \qquad \dot{v}(t) = u(t) - g_0 e_3, \\
& \dot{\eta}_{\text{p}}(t) = \chi_\theta(u(t)) + \chi_T(u(t)) + \chi_v(v(t)), \\
& \dot{\xi}_1(t) = [a_1 - r_x]_-^2 \, [r_x - b_1]_-^2 \, [c_1 - r_y]_-^2, \\
& \dot{\xi}_2(t) = [a_2 - r_x]_-^2 \, [r_x - b_2]_-^2 \, [r_y - c_2]_-^2, \\
& x(0) \in \mathcal{X}_{\text{i}}(r_{\text{i}}), \qquad x(t_{\text{f}}) \in \mathcal{X}_{\text{f}}(r_{\text{f}}), \\
& u(0) = u(t_{\text{f}}) = u_{\text{h}}, \qquad \eta_{\text{p}}(0) = 0, \\
& \xi_1(0) = 0, \qquad \xi_2(0) = 0.
\end{array}}$$

Figure 5.1 shows the optimized trajectory for the `always` obstacle-avoidance example. The vehicle departs from the straight start-to-goal path, bends around the two implication-type forbidden regions, and passes through the admissible corridor before reaching the terminal state. The color-coded speed profile along the trajectory also indicates a smooth acceleration and deceleration pattern, with no abrupt maneuvers required to satisfy the obstacle-avoidance task. Figure 5.2 confirms this behavior quantitatively. The plotted obstacle-margin signals remain in the safe region for the entire horizon, so the trajectory never enters either forbidden set. The common continuous-time tilt, thrust, and speed constraints are also satisfied throughout the maneuver.

Example 2: ***`eventually`*** *visit three waypoints*

The second example uses

$$r_{\mathrm{i}} = (-10, 0, 0), \qquad r_{\mathrm{f}} = (6, 0, 0), \qquad t_{\mathrm{f}} = 12 \text{ s}, \qquad K = 7,$$

and requires the vehicle to visit three waypoint regions centered at

$$p_{w,1} = (-8, -5, 0), \quad p_{w,2} = (-6, 5, 0), \quad p_{w,3} = (-4, -5, 0),$$

all with radius $r_w = 0.5$. Define the waypoint margins

$$\rho_i(r) := r_w^2 - \|r - p_{w,i}\|_2^2, \qquad i = 1, 2, 3,$$

and the corresponding predicates

$$\mu_{w,i}(r) := (\rho_i(r) \geq 0), \qquad i = 1, 2, 3.$$

Then the temporal task is

$$\varphi_{\mathrm{evt}} := \varphi_{\mathrm{path}} \wedge \boldsymbol{F}_{[0,t_{\mathrm{f}}]}\mu_{w,1} \wedge \boldsymbol{F}_{[0,t_{\mathrm{f}}]}\mu_{w,2} \wedge \boldsymbol{F}_{[0,t_{\mathrm{f}}]}\mu_{w,3}.$$

We augment the state as

$$x^{\varphi_{\mathrm{evt}}} := (x, \eta_{\mathrm{p}}, y_1, z_1, y_2, z_2, y_3, z_3),$$

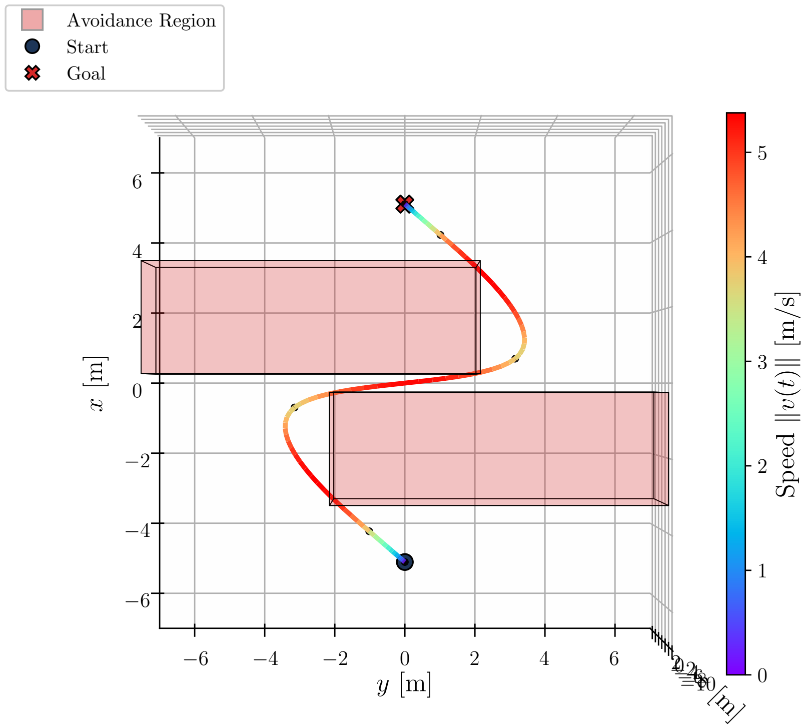


Figure 5.1: Optimized trajectory for the `always` obstacle-avoidance example. The vehicle departs from the direct start-to-goal path, passes through the admissible corridor between the two forbidden rectangular regions, and reaches the terminal state without entering either avoidance region. The trajectory is color-coded by speed.

where η_{p} again enforces the common path constraints, and for each waypoint we introduce a geometric-integral state y_i and an additive state z_i:

$$\dot{y}_i(t) = \frac{y_i(t)}{t_{\mathrm{f}}} \log\Big(\varepsilon_{\mathrm{evt}} + [\rho_i(r)]_-^2\Big), \qquad y_i(0) = 1,$$
$$\dot{z}_i(t) = \frac{1}{t_{\mathrm{f}}}[\rho_i(r)]_+^2, \qquad z_i(0) = 0,$$

The state y_i acts as a geometric-integral measure of how strongly the trajectory stays outside waypoint i, while z_i accumulates only when the vehicle is inside waypoint i. We define the

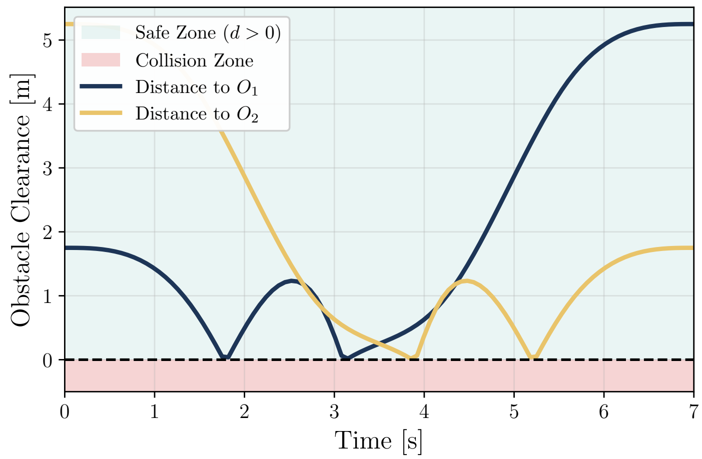


Figure 5.2: Obstacle-clearance histories for the `always` example. Both clearance signals remain in the safe region throughout the maneuver, confirming continuous-time satisfaction of the two obstacle-avoidance constraints.

corresponding waypoint-wise terminal surrogates

$$\widehat{\Gamma}_{w,i} := \left(c_{\mathrm{evt}}^2 + \alpha_i z_i(t_{\mathrm{f}})\right)^{1/2} - \left(c_{\mathrm{evt}}^2 + \beta_i y_i(t_{\mathrm{f}})\right)^{1/2},$$

for $i = 1, 2, 3$ with positive constants $c_{\mathrm{evt}}, \alpha_i, \beta_i$. Thus, entering waypoint i increases $\widehat{\Gamma}_{w,i}$ through z_i, whereas remaining outside decreases it through y_i.

The resulting problem is

$$\boxed{\begin{aligned}
\underset{x^{\varphi_{\mathrm{evt}}}(\cdot),u(\cdot)}{\text{maximize}}\quad & -w_\eta \eta_{\mathrm{p}}(t_{\mathrm{f}}) + \sum_{i=1}^{3} \widehat{\Gamma}_{w,i} \\
\text{subject to}\quad & \dot{r}(t) = v(t), \qquad \dot{v}(t) = u(t) - g_0 e_3, \\
& \dot{\eta}_{\mathrm{p}}(t) = \chi_\theta(u(t)) + \chi_T(u(t)) + \chi_v(v(t)), \\
& \dot{y}_i(t) = \frac{y_i(t)}{t_{\mathrm{f}}} \log\Big(\varepsilon_{\mathrm{evt}} + [\rho_i(r)]_-^2\Big), \quad \forall i, \\
& \dot{z}_i(t) = \frac{1}{t_{\mathrm{f}}}[\rho_i(r)]_+^2, \quad \forall i, \\
& x(0) \in \mathcal{X}_{\mathrm{i}}(r_{\mathrm{i}}), \qquad x(t_{\mathrm{f}}) \in \mathcal{X}_{\mathrm{f}}(r_{\mathrm{f}}), \\
& u(0) = u(t_{\mathrm{f}}) = u_{\mathrm{h}}, \qquad \eta_{\mathrm{p}}(0) = 0, \\
& y_i(0) = 1, \qquad z_i(0) = 0, \qquad \forall i.
\end{aligned}}$$

Figure 5.3 shows the optimized trajectory for the three-waypoint `eventually` example. Rather than proceeding directly to the goal, the trajectory bends toward each waypoint region in turn and then continues to the terminal state. The result highlights the role of the continuous-time event formulation. Satisfaction is achieved through the actual continuous-time trajectory rather than being tied to a small set of coarse discretization nodes. Figure 5.4 verifies that all three distance traces enter the satisfaction zone $d \leq r_w$ at some time in the interval. This behavior is exactly consistent with the three `eventually` requirements.

Example 3: ***`until`*** *charging-station task*

The third example uses

$$r_{\mathrm{i}} = (-6, 0, 0), \qquad r_{\mathrm{f}} = (6, 0, 0), \qquad t_{\mathrm{f}} = 5.5 \text{ s}, \qquad K = 6,$$

and requires the vehicle to keep its speed below a threshold until it reaches a charging-station region centered at

$$p_{\mathrm{c}} = (-4, -2, 0), \qquad d_{\mathrm{c}} = 0.2, \qquad v_{safe} = 2.0 \text{ m/s}.$$

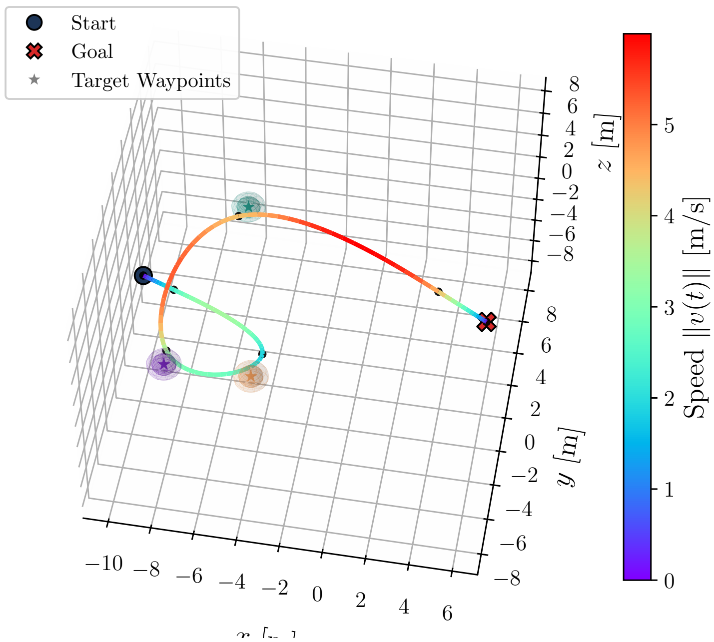


Figure 5.3: Optimized trajectory for the three-waypoint `eventually` example. The vehicle successively bends toward and visits all three waypoint regions before proceeding to the terminal state. The trajectory is color-coded by speed.

Define

$$\mu_{\mathrm{s}}(v) := \left(v_{safe}^2 - \|v\|_2^2 \geq 0\right),$$
$$\mu_{\mathrm{c}}(r) := \left(d_{\mathrm{c}}^2 - \|r - p_{\mathrm{c}}\|_2^2 \geq 0\right),$$

so that the desired temporal task is

$$\varphi_{\mathrm{unt}} := \varphi_{\mathrm{path}} \wedge \left(\mu_{\mathrm{s}}\, \boldsymbol{U}_{[0,t_{\mathrm{f}}]}\, \mu_{\mathrm{c}}\right).$$

To keep this example simple, the implementation uses a reduced augmentation tailored to the satisfaction mechanism of the `until` condition. We augment the state as

$$x^{\varphi_{\mathrm{unt}}} := (x, \eta_{\mathrm{p}}, y, z),$$

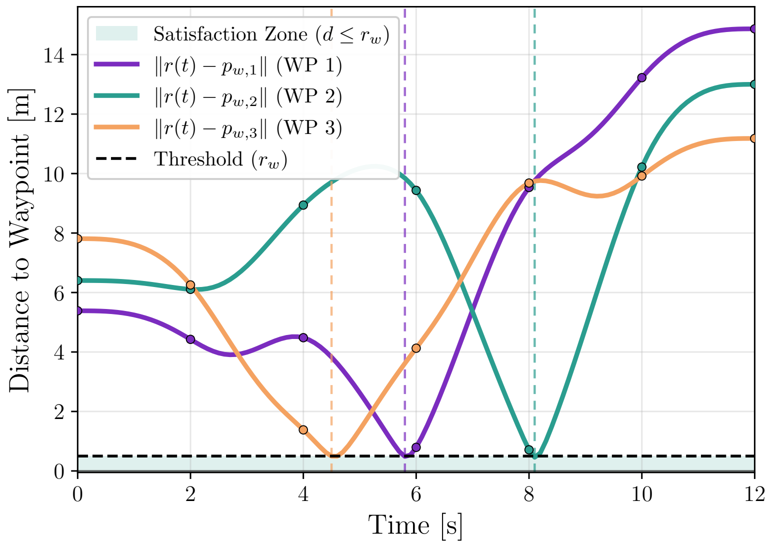


Figure 5.4: Distances from the vehicle to the three waypoint centers in the `eventually` example. Each trace enters the satisfaction zone $d \leq r_w$ at some time in the interval, confirming continuous-time satisfaction of all three `eventually` requirements.

where η_{p} handles the common path constraints, y accumulates the pre-arrival speed-threshold violation, and z acts as a multiplicative witness-state for the coupled `until` condition.

First, define

$$\dot{y}(t) = \frac{1}{t_{\mathrm{f}}} \left[v_{safe}^2 - \|v\|_2^2 \right]_-^2, \qquad y(0) = 0.$$

Thus, $y(t)$ remains small only if the speed bound is respected up to time t. We then introduce the normalized pre-arrival violation measure

$$q(t) := \frac{y(t)}{t + \varepsilon_t},$$

where $\varepsilon_t > 0$.

Next, define the charging-station miss measure

$$d(r) := \|r - p_{\mathrm{c}}\|_2^2 - d_{\mathrm{c}}^2,$$

which is negative inside the charging region and positive outside. The speed and charging conditions are coupled through

$$\chi(r, y, t) := \sqrt{C_1^2 + \frac{1}{2}\left([d(r)]_+^2 + [q(t)]_+^2\right)} - C_1,$$

with $C_1 > 0$. This quantity can be small only when the vehicle is near or inside the charging region and the pre-arrival speed-violation measure is also small.

Finally, the multiplicative witness-state z evolves according to

$$\dot{z}(t) = \frac{z(t)}{t_\mathrm{f}} \log\Big(\chi(r(t), y(t), t)^2 + \varepsilon_\mathrm{u}\Big), \qquad z(0) = 1,$$

with $\varepsilon_\mathrm{u} > 0$. Since $z(0) = 1$ and the dynamics are multiplicative, the terminal value $z(t_\mathrm{f})$ depends on the entire time history of the coupled miss measure χ. We therefore define the terminal surrogate

$$\widehat{\Gamma}_\mathrm{unt} := c_\mathrm{u} - \sqrt{c_\mathrm{u}^2 + z(t_\mathrm{f})},$$

with $c_\mathrm{u} > 0$. Maximizing $\widehat{\Gamma}_\mathrm{unt}$, or equivalently minimizing $z(t_\mathrm{f})$, favors trajectories for which the coupled quantity χ remains small in the sense required by the `until` semantics.

The resulting problem is

$$\boxed{\begin{array}{ll}
\underset{x^{\varphi_\mathrm{unt}}(\cdot), u(\cdot)}{\text{maximize}} & -w_\eta \eta_\mathrm{p}(t_\mathrm{f}) + \widehat{\Gamma}_\mathrm{unt} \\
\text{subject to} & \dot{r}(t) = v(t), \qquad \dot{v}(t) = u(t) - g_0 e_3, \\
& \dot{\eta}_\mathrm{p}(t) = \chi_\theta(u(t)) + \chi_T(u(t)) + \chi_v(v(t)), \\
& \dot{y}(t) = \dfrac{1}{t_\mathrm{f}} \left[v_{safe}^2 - \|v\|_2^2\right]_-^2, \\
& \dot{z}(t) = \dfrac{z(t)}{t_\mathrm{f}} \log\Big(\chi(r(t), y(t), t)^2 + \varepsilon_\mathrm{u}\Big), \\
& x(0) \in \mathcal{X}_\mathrm{i}(r_\mathrm{i}), \qquad x(t_\mathrm{f}) \in \mathcal{X}_\mathrm{f}(r_\mathrm{f}), \\
& u(0) = u(t_\mathrm{f}) = u_\mathrm{h}, \qquad \eta_\mathrm{p}(0) = 0, \\
& y(0) = 0, \qquad z(0) = 1.
\end{array}}$$

Figures 5.5–5.7 show the optimized trajectory, speed history, and charging-station margin for the `until` example. The vehicle first diverts toward the charging station and only

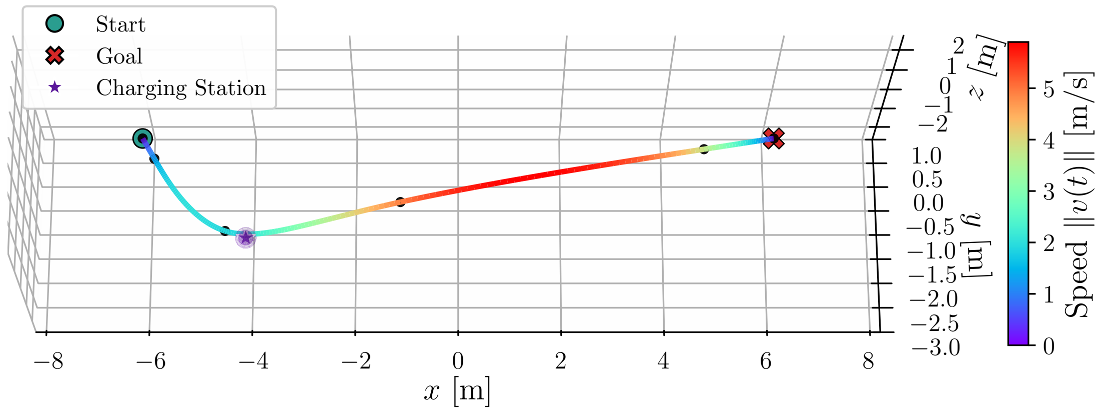


Figure 5.5: Optimized trajectory for the charging-station `until` example. The vehicle first reaches the charging-station region and then proceeds to the terminal state, consistent with the continuous-time `until` specification. The trajectory is color-coded by speed.

afterwards proceeds to the terminal state. The speed trace remains below the prescribed threshold v_{safe} before the visiting event, while also remaining below the global speed bound for the entire maneuver. The signed charging-station margin is negative before arrival, crosses zero at the station boundary, and becomes positive inside the charging region. The first zero crossing therefore identifies the witness time at which the charging event occurs, exactly as required by the `until` semantics.

5.3 Receding-horizon extension: NMPC with `always` specifications

This section specializes the continuous-time path-constraint machinery of Chapters 2 and 3 to a receding-horizon setting. In the language of Chapter 3, the constraints considered here are `always`-type mission specifications, while the numerical transcription and solution strategy build directly on the time-dilation, multiple-shooting, exact-penalty, and convexification ingredients introduced earlier. The resulting nonlinear model predictive control (NMPC) framework ensures continuous-time constraint satisfaction and accurate evaluation of the running cost without sacrificing the efficiency required for real-time implementation.

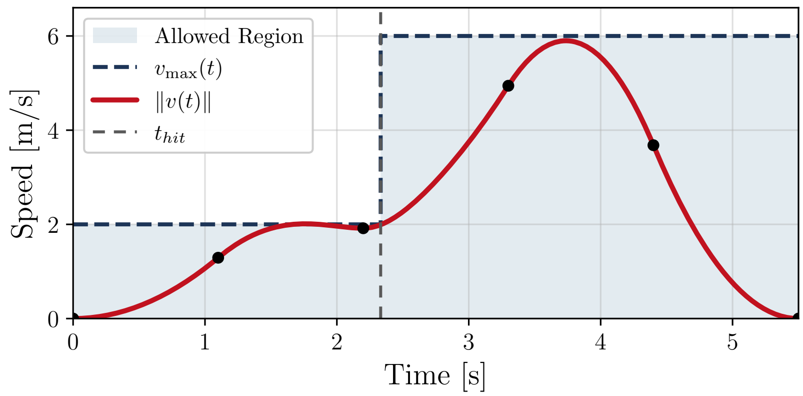


Figure 5.6: Speed profile for the charging-station `until` example. The speed remains below the pre-charging threshold v_{safe} until the charging event occurs, while also remaining below the global speed bound throughout the maneuver.

5.3.1 Introduction

Nonlinear model predictive control (NMPC) is a promising method for autonomous systems owing to its ability to achieve key mission objectives while satisfying constraints and ensuring robustness to uncertainties [119]. This capability stems from the optimal-control viewpoint, in which control objectives and constraints are encoded in a finite-horizon optimization problem that is repeatedly solved online in a receding-horizon fashion. Due to these benefits, NMPC has been widely applied to various real-world applications, including the control of wind turbines [257], humanoid robots [276], ground vehicles [79], and autonomous aerospace systems [108].

Different types of optimal-control solvers have been used for NMPC. Indirect methods based on the maximum principle [169] solve a two-point boundary value problem induced by the necessary conditions for optimality and have been used in real-world NMPC applications [153, 303]. Another important family is differential dynamic programming (DDP) [191], which iteratively performs a forward rollout of the system dynamics and a backward dynamic-programming update. DDP has been used for MPC problems with dynamics and initial-

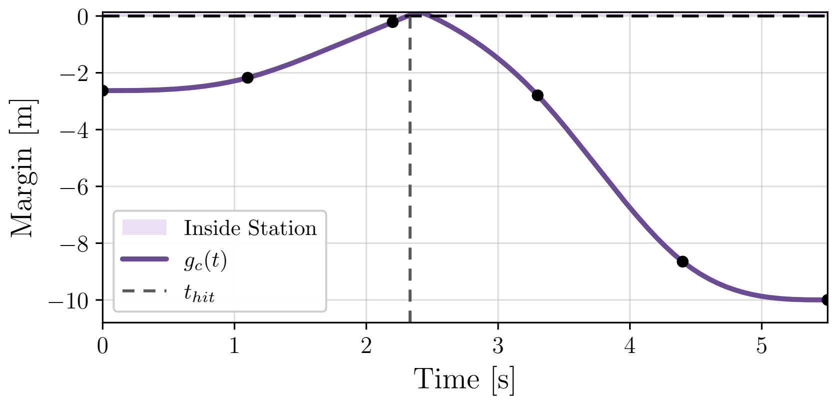


Figure 5.7: Signed charging-station margin for the `until` example. The margin is negative outside the charging region, zero on its boundary, and positive inside. The first zero crossing marks the witness time for the charging event in the `until` specification.

condition constraints [276], extended to input constraints through constrained dynamic programming [277], and combined with augmented Lagrangian techniques to handle more general state and input constraints [139].

The direct optimal-control approach to NMPC has also received significant attention [235]. In this approach, the infinite-dimensional optimal control problem is first discretized to yield a finite-dimensional problem using methods such as single shooting [253] or multiple shooting [36, 232]. The resulting nonlinear program is then typically solved using nonlinear interior-point methods [120], sequential quadratic programming (SQP) [137], or sequential convex programming (SCP) [184]. SQP-type approaches have seen widespread adoption owing to their real-time capability [231], especially in conjunction with the real-time iteration (RTI) scheme [82, 126]. SCP-based approaches to NMPC have also been gaining popularity recently [83, 186, 207].

The SCP approach adopted here has several advantages over SQP [183, 202]. First, SCP requires only gradients, that is, first-order information, whereas SQP typically requires Hessian information as well. Second, SCP subproblems depend only on the primal variables from the previous iteration, whereas SQP subproblems also depend on dual variables. Third, the SCP formulation used in this dissertation can preserve the existing convex structure. Convex objectives and convex constraints can be kept intact, while only the nonconvex smooth components are locally approximated. Because the framework used in this section is built entirely from $\mathcal{C}^1$-smooth functions after reformulation, gradient-based SCP methods are particularly attractive.

Typically, NMPC implementations are developed for nonlinear dynamics with relatively simple input constraints, often box-type constraints [177]. Moreover, because piecewise-constant, i.e., zero-order hold, or piecewise-affine, i.e., first-order hold, control parameterizations are commonly adopted, and because the imposed control constraints are usually convex, continuous-time satisfaction of such control constraints can often be guaranteed directly from nodal enforcement. Modern applications, however, require more general mission constraints involving both state and control variables [133]. In such settings, the standard practice is to enforce constraints, and to approximate the running cost, only at finitely many instants, typically at the transcription nodes. While this is often computationally convenient, it does not in general guarantee continuous-time constraint satisfaction, nor does it preserve accurate accumulation of the running cost between nodes. A common mitigation strategy is to use dense grids or adaptive mesh-refinement techniques [117]. However, these approaches increase the problem dimension and computational burden, in direct conflict with real-time requirements [312]. Other approaches to continuous-time safety rely on maximal robust control invariant sets, especially for linear systems [248, 298] and sampled-data systems [127, 203]. To the best of our knowledge, however, a systematic real-time-capable model-based NMPC framework for continuous-time constraint satisfaction had remained unavailable prior to the line of work built on the CTCS framework [104].

To address these issues, this section develops an NMPC framework based on the continuous-time constraint-satisfaction ideas introduced earlier in the dissertation. More specifically, it uses the path-constraint reformulation of Chapter 3, the multiple-shooting transcription of Chapter 2, and a SCP solution strategy. The resulting framework has three key features:

1. a constraint and cost reformulation that enables longer planning horizons with fewer nodes without degradation in inter-sample constraint satisfaction or running-cost accuracy;
2. multiple-shooting exact discretization using the inverse-free variational approach [36, 144, 231];
3. a convergence-guaranteed SCP method, namely the prox-linear method [91], which supports warm-starting and premature termination and is therefore well suited to real-time receding-horizon implementation. This is precisely the composite structure discussed in Chapter 4, specialized here to the prox-linear setting.

The efficacy of the resulting framework is demonstrated on an NMPC reference-tracking problem with obstacle avoidance. The implementation corresponding to the original paper is available at `https://github.com/UW-ACL/nmpc-ctcs`.

5.3.2 Receding-horizon optimal control problem with `always` *specifications*

To distinguish the physical plant state from the auxiliary states introduced by the reformulation, let

$$\xi(t) \in \mathbb{R}^{n_\xi}$$

denote the physical state and

$$u(t) \in \mathbb{R}^{n_u}$$

the control input. Consider the continuous-time system

$$\dot{\xi}(t) = F\Big(t, \xi(t), u(t)\Big), \tag{5.1}$$

subject to path constraints

$$g\Big(t, \xi(t), u(t)\Big) \leq 0, \tag{5.2a}$$

$$h\Big(t, \xi(t), u(t)\Big) = 0, \tag{5.2b}$$

where g and h are vector-valued functions.

At each current time $t_c \in \mathbb{R}_{\geq 0}$ and horizon length $t_h \in \mathbb{R}_{++}$, the receding-horizon optimal control problem is

$$\begin{aligned}
\underset{u(\cdot)}{\text{minimize}} \quad & \Phi\Big(\xi(t_c + t_h)\Big) + \int_{t_c}^{t_c+t_h} L\Big(t, \xi(t), u(t)\Big)\, dt && \text{(5.3a)} \\
\text{subject to} \quad & \dot{\xi}(t) = F\Big(t, \xi(t), u(t)\Big), \qquad t \in [t_c, t_c + t_h], && \text{(5.3b)} \\
& g\Big(t, \xi(t), u(t)\Big) \leq 0, \qquad t \in [t_c, t_c + t_h], && \text{(5.3c)} \\
& h\Big(t, \xi(t), u(t)\Big) = 0, \qquad t \in [t_c, t_c + t_h], && \text{(5.3d)} \\
& \xi(t_c) = \xi_c, && \text{(5.3e)} \\
& \Psi\Big(\xi(t_c + t_h)\Big) \leq 0, && \text{(5.3f)}
\end{aligned}$$

where ξ_c is the current state estimate, L is the running cost, Φ is the terminal cost, and Ψ is a terminal constraint function. We assume that F, g, h, L, Φ, and Ψ are $\mathcal{C}^1$ and potentially nonconvex.

Following the running-cost augmentation introduced in Chapter 2 and the continuous-time path-constraint construction introduced in Chapter 3, the path constraints and running cost are reformulated by augmenting the dynamics. Define the augmented state

$$x(t) := \Big(\xi(t), \ell(t), y(t)\Big),$$

where ℓ accumulates the running cost and y accumulates the continuous-time path-constraint violation. The augmented dynamics are

$$\dot{x}(t) = \begin{bmatrix} F\Big(t, \xi(t), u(t)\Big) \\ L\Big(t, \xi(t), u(t)\Big) \\ 1^\top |g\Big(t, \xi(t), u(t)\Big)|_+^2 + 1^\top h\Big(t, \xi(t), u(t)\Big)^2 \end{bmatrix} =: f\Big(t, x(t), u(t)\Big). \tag{5.4}$$

The integral term in (5.3a) is therefore equal to $\ell(t_c + t_h)$ when the initial condition

$$\ell(t_c) = 0$$

is imposed. Continuous-time satisfaction of the path constraints is enforced through the periodic boundary condition

$$y(t_c) = y(t_c + t_h), \tag{5.5}$$

which is the receding-horizon counterpart of the `always`-type path-constraint construction used earlier.

Defining the augmented terminal functions

$$\widetilde{\Phi}\Big(x(t_c + t_h)\Big) := \Phi\Big(\xi(t_c + t_h)\Big) + \ell(t_c + t_h), \tag{5.6}$$

$$\widetilde{\Psi}\Big(x(t_c + t_h)\Big) := \Psi\Big(\xi(t_c + t_h)\Big), \tag{5.7}$$

we obtain a Mayer-form receding-horizon problem in the augmented variables. The multiple-shooting transcription, FOH control parameterization, and exact-penalty finite-dimensional formulation follow exactly the machinery developed in Chapter 2, so they are not repeated here in detail. Likewise, the resulting penalized finite-dimensional problem belongs to the $g + h(C)$ subclass of the composite optimization framework discussed in Chapter 4, and is solved here by the corresponding prox-linear method [91].

For numerical implementation, the continuous-time boundary condition (5.5) is relaxed interval-wise after discretization as in the CTCS framework, using a small parameter $\epsilon > 0$ to preserve constraint qualifications. The resulting SCP subproblems are convex and retain the same structure regardless of the number or type of path constraints, which is particularly useful for real-time and code-generated implementations.

More explicitly, after discretization and exact penalization, the finite-dimensional receding-horizon problem has the compact form

$$\underset{Z \in \mathcal{Z}}{\text{minimize}} \quad G(Z) + w_{\text{ep}} H(C(Z)), \tag{5.8}$$

where $Z = (x_1, \dots, x_N, u_1, \dots, u_N)$, the set $\mathcal{Z}$ collects the convex constraints, and $H \circ C$ is a convex-composite term built from the discretized defects and terminal conditions. At SCP iteration $j + 1$, the prox-linear method solves the convex subproblem

$$\underset{Z \in \mathcal{Z}}{\text{minimize}} \quad G(Z) + w_{\text{ep}} H\Big(C(Z^j) + \nabla C(Z^j)(Z - Z^j)\Big) + \frac{w_{\text{prox}}}{2} \|Z - Z^j\|_2^2, \tag{5.9}$$

where Z^j is the solution from the previous SCP iteration. This is precisely the composite structure discussed in Chapter 4, specialized here to the prox-linear setting.

In the implementation corresponding to the original NMPC paper, CVXPY is used for modeling, together with ECOS or MOSEK for solving the convex subproblems [81, 88, 208]. Customizable conic solvers such as the proportional-integral projected gradient (PIPG) algorithm [307, 308], especially when combined with conditioning techniques [146], could also be used for efficient implementations. A useful feature of the reformulation is that, once a fixed path-constraint template is chosen, the resulting convex subproblem structure is largely unaffected by the specific number and type of active path constraints. This is especially beneficial for customized solver generation and code generation, because enabling or disabling constraints does not change the conic program template and therefore does not require re-parsing or re-customization [103, 143].

To assess the convergence of SCP within each NMPC call, we monitor three quantities:

$$J_\ell := E_\ell \sum_{k=1}^{N-1} (\mu_k^+ + \mu_k^-), \tag{5.10a}$$

$$J_y := E_y \sum_{k=1}^{N-1} (\mu_k^+ + \mu_k^-), \tag{5.10b}$$

$$J_{\text{prox}} := \sum_{k=1}^{N} \Big(\|x_k - x_k^j\|_2^2 + \|u_k - u_k^j\|_2^2 \Big), \tag{5.10c}$$

where E_ℓ extracts the running-cost accumulator and E_y extracts the path-constraint accumulator. Here, μ_k^+ and μ_k^- denote the interval-wise relaxation variables introduced in the multiple-shooting transcription, while E_ℓ and E_y are selector row vectors that extract, respectively, the running-cost and path-constraint accumulator components from the corresponding relaxed defect channels. Taken together, these quantities measure the linearization error in the running-cost accumulation, violation of the continuous-time path constraints, and overall convergence of the SCP iterates.

5.3.3 The NMPC framework

The NMPC algorithm recursively solves the receding-horizon problem (5.3). In the present setting, the path-constraint and running-cost reformulation makes it possible to operate with relatively sparse discretization grids without sacrificing inter-sample feasibility or running-cost accuracy. This is particularly important in the receding-horizon setting, where the horizon must be long enough to anticipate future events but the optimization problem must still be solved in real time.

Let the horizon $[t_c, t_c + t_h]$ be discretized by N nodes, so that

$$\Delta t = \frac{t_h}{N-1}.$$

Let $K \leq N-1$ denote the number of subintervals between consecutive NMPC calls. The time between successive NMPC runs is then

$$\Delta t_{\mathrm{MPC}} = K\,\Delta t.$$

The choice of N and K should balance two competing considerations. On the one hand, N must be large enough to provide sufficient control degrees of freedom and to handle stiffness [296] in the augmented dynamics. On the other hand, keeping N modest and K reasonably large is essential for computational efficiency. The CTCS reformulation is particularly valuable in the non-stiff setting, where it allows the discretization to be selected primarily according to control expressiveness rather than inter-sample feasibility requirements. In situations where the state changes rapidly or the objectives and constraints vary quickly, the augmented dynamics can become stiff, and then choosing N large enough to make Δt sufficiently small is desirable to alleviate ill-conditioning in the sensitivity matrices and to allow the controller to react promptly. In non-stiff settings, by contrast, the reformulation permits the use of relatively sparse node grids without risking inter-sample constraint violations or degrading the running-cost accuracy.

A second key ingredient is warm-starting. Since consecutive NMPC problems differ only gradually, the solution from the previous horizon provides a highly effective initialization for the next SCP call. The left superscript of the optimization variables denotes the NMPC run, while the right superscript denotes the SCP iteration within that run. When K is a positive integer, SCP at the current time instant is warm-started by shifting the SCP solution from the previous time instant. Figure 5.8 illustrates the procedure.

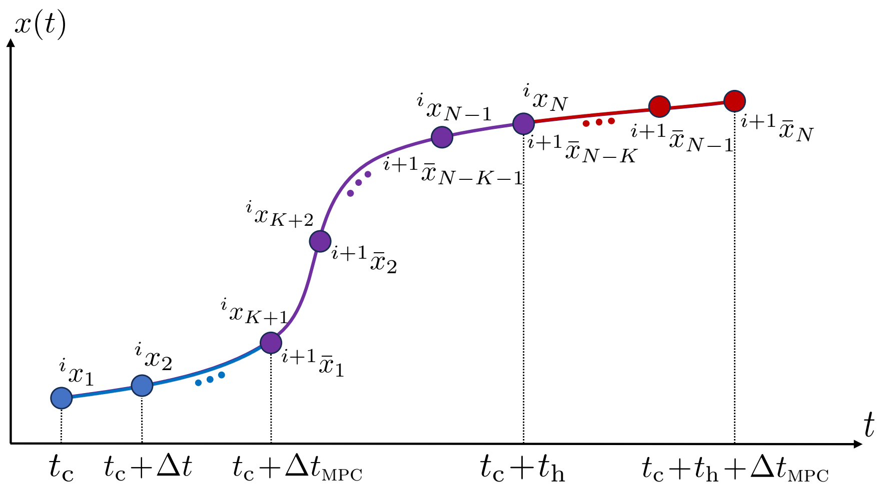


Figure 5.8: Illustration of the warm-starting procedure. The blue and purple trajectories form the SCP solution at the ith NMPC run. The purple and red trajectories form the initialization to SCP at the $(i+1)$th NMPC run.

The shifted warm start is defined by

$$^{i+1}\bar{x}_j = {}^{i}x_{j+K}, \tag{5.11}$$

$$^{i+1}\bar{u}_j = {}^{i}u_{j+K}, \qquad j = 1, \dots, N-K. \tag{5.12}$$

For the remaining control nodes, we set

$$^{i+1}\bar{u}_{N-K+j} = {}^{i}u_N, \qquad j = 1, \dots, K, \tag{5.13}$$

and the remaining states are generated by propagating the augmented dynamics forward from the terminal state of the previous horizon under the held terminal control:

$$^{i+1}\bar{x}_{N-K+j} = {}^{i}x_N + \int_{t_c+t_h}^{t_c+t_h+j\Delta t} f\Big(t, x(t), {}^{i}u_N\Big)\, dt, \qquad j = 1, \dots, K. \tag{5.14}$$

Therefore, if the preceding SCP solution is dynamically feasible, then the initialization for the subsequent SCP call is dynamically feasible as well. This aspect of warm-starting is crucial for achieving rapid convergence.

Because warm-starting typically places the new iterate near the current receding-horizon solution, the SCP algorithm can be terminated well before full convergence, in a manner analogous to RTI schemes [82]. In the present framework, premature termination does not compromise solution quality significantly in practice, while it substantially improves computational efficiency.

The overall NMPC framework is summarized in Algorithm 2.

Algorithm 2 NMPC Framework

Inputs: T, t_h, K, N, $j_{\max}$, ${}^1\bar{Z}$

$i \leftarrow 1$

$t_c \leftarrow 0$

$\Delta t \leftarrow t_h/(N-1)$

$\Delta t_{\text{MPC}} \leftarrow K\Delta t$

while $t_c \leq T$ **do**

  $\hat{x}(t_c), {}^i\!L, {}^i\!g, {}^i\!h \leftarrow \textsc{sensor}(x(t_c))$

  ${}^iZ \leftarrow \textsc{scp}^{j_{\max}}_{N,t_c,t_h}(\hat{x}(t_c), {}^i\!L, {}^i\!g, {}^i\!h, {}^i\bar{Z})$

  ${}^{i+1}\bar{Z} \leftarrow \textsc{warm\_start}({}^iZ)$

  $x(t_c + \Delta t_{\text{MPC}}) \leftarrow \textsc{evolve}(x(t_c), {}^iZ)$

  $i \leftarrow i + 1$

  $t_c \leftarrow t_c + \Delta t_{\text{MPC}}$

end while

At each NMPC run, sensor information provides the current state estimate together with the current running-cost and path-constraint data, both of which may vary over time. A fixed, small number of prox-linear iterations is then performed to solve the current horizon problem. The resulting suboptimal solution is used both to generate the applied control input and to warm-start the next NMPC run. In this way, the framework combines continuous-time path-constraint satisfaction, warm-started receding-horizon optimization, and early termination of SCP iterations in a unified and computationally efficient NMPC loop.

5.3.4 Numerical results

We demonstrate the proposed NMPC framework on a reference-tracking problem with obstacle avoidance for a double-integrator system with drag and bounded control inputs. The physical state is

$$\xi(t) = (r(t), v(t)),$$

where $r(t)$ and $v(t)$ denote position and velocity, and the governing dynamics are

$$F\Big(t, \xi(t), u(t)\Big) = \begin{bmatrix} v(t) \\ u(t) - c_d\|v(t)\|v(t) \end{bmatrix}, \tag{5.15}$$

with drag coefficient c_d.

The running cost is chosen as the reference-tracking penalty

$$L\Big(t, \xi(t), u(t)\Big) = \|r(t) - r_{\text{ref}}(t)\|^2, \tag{5.16}$$

where r_{ref} is the desired output trajectory. We assume that the vehicle has a local sensor, such as LiDAR, with sensing radius r_{sens}, and that only the currently detected obstacles are incorporated into each receding-horizon problem. If the total number of obstacles is n_{obs}, with centers r^i_{obs} and shape matrices S^i, then the detected index set at time t_c is

$$\mathcal{O}_c(t_c) := \Big\{ i \in \{1, \ldots, n_{\text{obs}}\} : \|S^i(r(t_c) - r^i_{\text{obs}})\| \leq r_{\text{sens}} \Big\}. \tag{5.17}$$

The path-constraint function then includes the corresponding obstacle-avoidance terms,

$$g\Big(t, \xi(t), u(t)\Big) = \begin{bmatrix} \vdots \\ 1 - \|S^i(r(t) - r^i_{\text{obs}})\| \\ \vdots \end{bmatrix}, \qquad i \in \mathcal{O}_c(t_c). \tag{5.18}$$

Strictly speaking, the obstacle-margin function is nondifferentiable at the obstacle center; in practice this causes no issue unless the iterate is initialized exactly at such a point. The control input is constrained by

$$\mathcal{U} := \{u \in \mathbb{R}^2 : \|u\| \leq u_{\max}\}.$$

As in the original CTCS development, the discretized relaxation parameter $\epsilon > 0$ may admit a pointwise path-constraint violation unless a small tightening parameter $\delta_{\text{obs}} > 0$ is introduced. For each fixed $\epsilon > 0$, there exists a corresponding $\delta_{\text{obs}} > 0$ such that the tightened integral condition is equivalent to exact obstacle satisfaction in continuous time. In the experiments below, the value of δ_{obs} is chosen empirically.

The system and simulation parameters are summarized in Table 5.1.

Table 5.1: Parameters for the NMPC reference-tracking and obstacle-avoidance example.

Parameter	Value
$T,\ t_h,\ K,\ N,\ j_{\max}$	$30\ \text{s},\ 8\ \text{s},\ 4,\ 17,\ 3$
$u_{\max}$	$5\ \text{m}\,\text{s}^{-2}$
c_d	$0.05\ \text{m}^{-1}$
$r_{\text{ref}}(t)$	$\begin{bmatrix} -10 + 2t/3 \\ 5e^{-0.05t}\sin\left(\pi/2 + t/5 + t^2/36\right) \end{bmatrix}$
$S^i,\ i = 1, \ldots, 5$	$\begin{bmatrix} 2.4 & 0 \\ 0 & 0.4 \end{bmatrix}$
r_{obs}	$\begin{bmatrix} -6.5 & -1.8 & 0.5 & 2.7 & 6.2 \\ -3 & 3.5 & -3.5 & 1.5 & -3 \end{bmatrix}$
δ_{obs}	0.05
r_{sens}	$6\ \text{m}$

We first study the impact of premature termination on solution quality. Within each NMPC run, SCP is executed for ten iterations, and the quantities J_ℓ, J_y, and J_{prox} from (5.10) are recorded at each iteration. When these quantities converge to small values, the reformulated running cost is being accumulated accurately, the obstacle-avoidance constraint is satisfied, and the overall SCP iteration is close to convergence. Figure 5.9 shows the mean and standard deviation of these quantities along the trajectory as a function of the number of SCP iterations. The plots indicate that terminating SCP after only three iterations yields an acceptable solution.

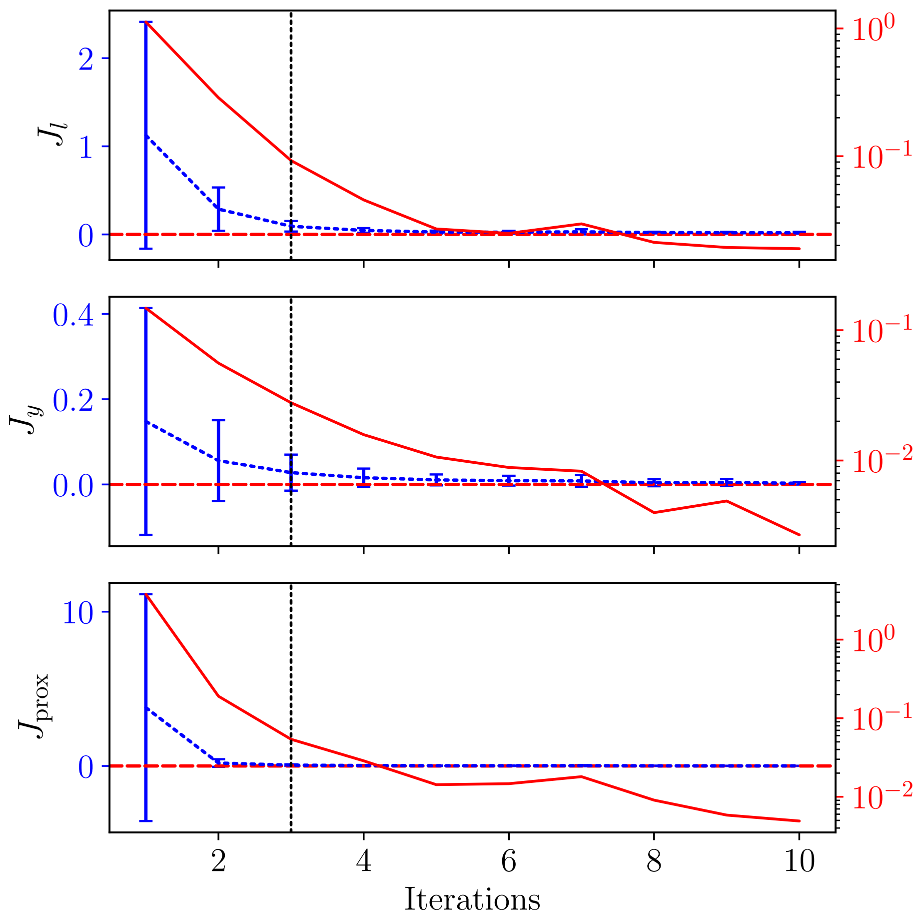


Figure 5.9: Mean and standard deviation of J_ℓ, J_y, and J_{prox} along the trajectory as functions of the number of SCP iterations. The left axis corresponds to the mean and standard deviation shown by the blue curve, while the right axis corresponds to the mean shown by the red curve on a logarithmic scale. The black dashed line marks the premature-termination choice used in the NMPC simulation.

We next compare the proposed continuous-time formulation against the standard node-only approach, in which the running cost and obstacle constraints are evaluated only at transcription nodes. Figures 5.10 and 5.11 summarize the results.

Figure 5.10 shows that in the node-only case, the obstacle-avoidance constraint is violated by the continuous-time trajectory obtained by propagating the system dynamics under the NMPC control law, even though the constraint is satisfied at all transcription nodes. By contrast, the CTCS formulation guarantees that the obstacle-avoidance constraint is satisfied at all times along the trajectory.

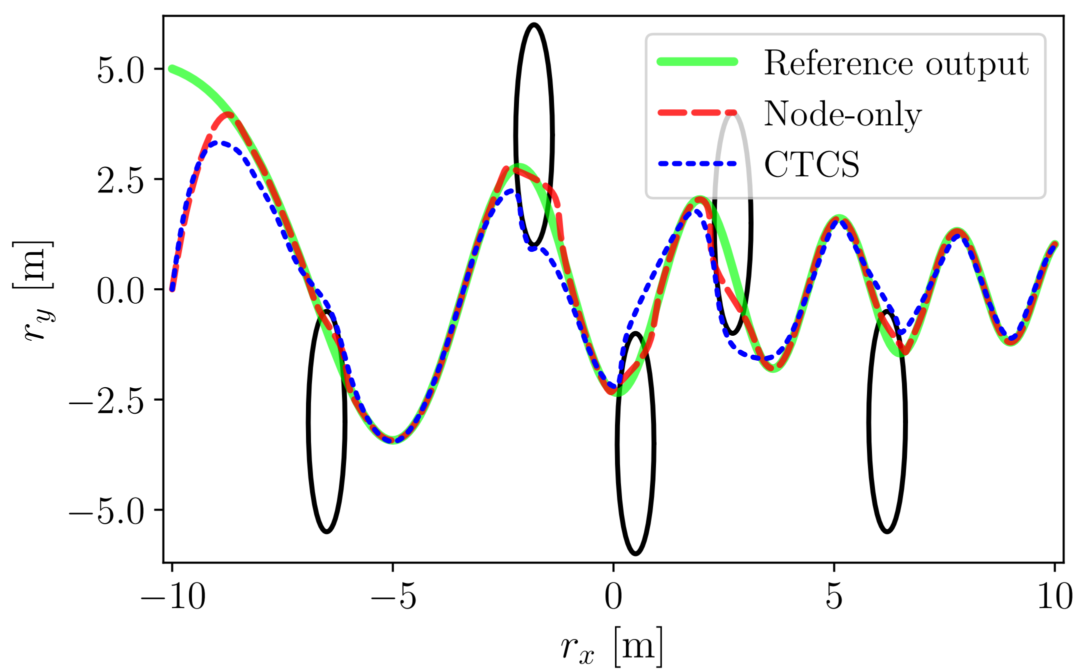


Figure 5.10: Reference-tracking and obstacle-avoidance trajectories obtained by NMPC. The green solid curve denotes the reference output, and the black ellipses denote the obstacles. The red dashed curve is the trajectory obtained with the node-only formulation, while the blue dotted curve is obtained with the CTCS formulation. The node-only formulation violates obstacle avoidance between nodes, whereas the CTCS trajectory remains continuously feasible.

Figure 5.11 shows that both formulations achieve comparable output-tracking performance. The CTCS formulation incurs only a slight degradation in tracking accuracy, and this difference is precisely what enables the continuous-time satisfaction of the obstacle-avoidance requirement. Thus, the reformulated NMPC framework preserves the overall control objective while eliminating inter-sample safety violations.

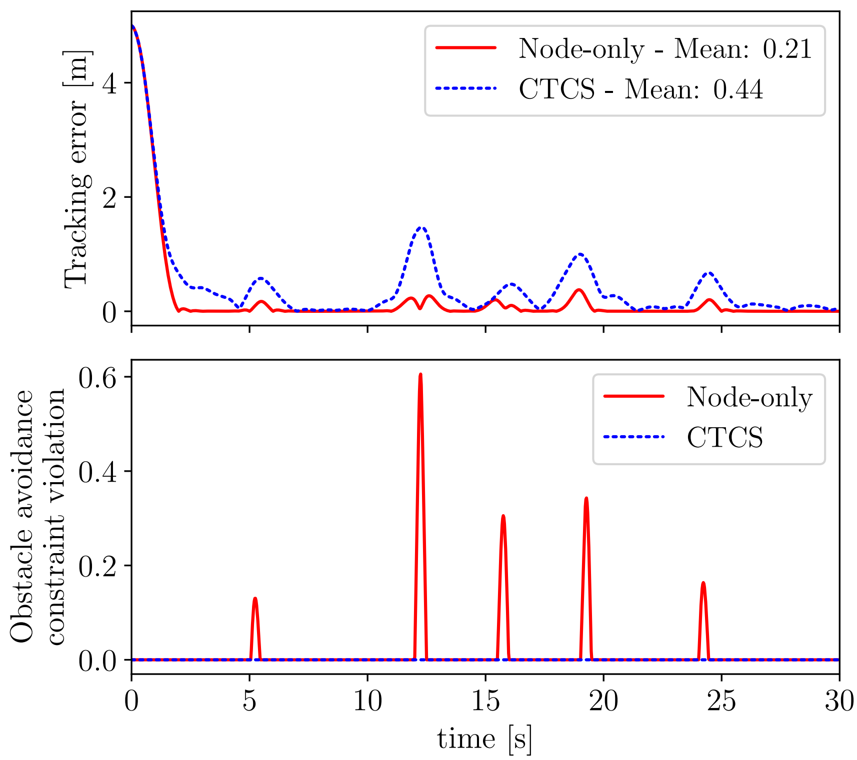


Figure 5.11: Comparison of tracking and obstacle-avoidance performance. The first plot shows the discrepancy between the reference output and the trajectories obtained from NMPC using the node-only and CTCS formulations throughout the simulation; the legend reports the average discrepancy over the full simulation. The second plot shows the obstacle-avoidance violation, quantified by $1^\top|g(t, \xi(t), u(t))|_+$, throughout the simulation. The CTCS formulation maintains zero continuous-time violation, whereas the node-only formulation does not.

These results illustrate the main point of this section. By embedding both the running cost and the path constraints into the continuous-time dynamics, one can operate NMPC on relatively sparse grids without sacrificing inter-sample path-constraint satisfaction or running-cost accuracy. When combined with warm-starting and early termination of the SCP iterations, this yields a practically efficient receding-horizon control framework for continuous-time constrained nonlinear systems.

5.4 Quadrotor motion planning for information acquisition under sensing limitations

This section applies the modeling and optimization machinery developed in the previous chapters to a perception-constrained motion-planning problem. In contrast to the illustrative CT-STL examples of Section 5.2 and the receding-horizon setting of Section 5.3, the objective here is to generate a trajectory that both satisfies the vehicle dynamics and constraints and acquires a prescribed amount of information about multiple monitoring targets under sensing limitations. A key modeling distinction is that the continuous-time path constraints and the accumulated-information requirement are embedded directly into the augmented dynamics, while the sensing windows that guarantee nonzero information-acquisition rate and nonzero gradient are enforced through discrete-time STL specifications parameterized by generalized mean-based smooth robustness. The resulting finite-dimensional problem is then solved using the continuous-time successive convexification machinery of Chapters 2 and 3, together with the convexification framework developed in Chapter 4.

5.4.1 Introduction

Monitoring distributed infrastructures, such as power lines, roadways, and water networks, for preventive maintenance, or tracking large-scale natural phenomena like erosion, forest overgrowth, and farmland flooding for risk mitigation is a labor-intensive, physically demanding, and potentially hazardous task. These challenges make automation technologies an ideal solution to alleviate the burden on human operators by mitigating the associated risks and difficulties [97, 114, 313]. Rather than dispatching people to survey impervious and faraway areas, teams of autonomous robots, for example drones potentially supported by ground robots acting as mobile base stations, can be deployed to gather information about monitoring targets, which can then be reviewed and analyzed for corrective actions or further investigation.

As a result, in recent years there has been growing interest in using robots for monitoring purposes. In particular, it may be convenient to use small and inexpensive mobile robots in larger numbers as opposed to fewer, larger robots for cost, safety, and increased parallelization. Lower-cost robots, such as micro-drones, are often equipped with less capable sensors, making it crucial to consider sensor limitations and information-acquisition performance when planning drone operations. For instance, considering sensor limitations such as range and field of view, decisions must be made on whether to fly at low altitude for detailed target acquisition or at a higher altitude to scan a larger area.

In prior works on automated monitoring, task-assignment problems for dispatching drones based on current knowledge of the targets have been investigated in [171, 278], and motion planning for ground robots serving as mobile stations for drone deployment and recovery has been studied in [154]. More recently, there has also been growing interest in perception-aware planning and control [41, 110, 125, 251, 267, 282]. Most of these works focus on limitations imposed by the information perceived about the environment rather than on planning trajectories specifically to acquire information while accounting for sensor limitations.

The present section addresses motion planning for information-acquisition tasks, such as monitoring, while explicitly accounting for acquisition performance in relation to sensor limitations including range, field of view, and target acquisition direction. We introduce a smooth, nonnegative information-acquisition metric that quantifies the rate of information acquisition and takes a positive value only when all relevant sensing limitations are satisfied. The smoothness of the proposed metric makes it compatible with optimization-based trajectory generation and, in particular, with the optimal-control framework reviewed earlier in Chapter 2, where information acquisition can be imposed by continuous-time path constraints [104].

A fundamental difficulty, however, is that the information-acquisition metric is uniformly zero beyond the sensor limitations, which also yields a uniformly zero gradient and therefore hinders numerical optimization. To address this issue, we impose signal temporal logic specifications that ensure the information-acquisition metric remains positive and retains a nonzero gradient over designated time intervals. These specifications are parameterized using the generalized mean-based smooth robustness framework of Chapter 3, yielding an exact smooth parameterization of the underlying discrete-time STL condition that remains compatible with the convergence guarantees of the optimal-control framework [104, 286].

5.4.2 Information acquisition metric

We consider an agent with dynamics

$$\dot{x}(t) = F\Big(t, x(t), u(t)\Big), \tag{5.19}$$

where $u \in \mathbb{R}^m$ is the input vector, $x \in \mathbb{R}^n$ is the state vector that includes position $r \in \mathbb{R}^3$, attitude $\Omega \in \mathbb{R}^3$, and possibly additional states, and where the agent is subject to path constraints

$$g\Big(t, x(t), u(t)\Big) \leq 0, \qquad h\Big(t, x(t), u(t)\Big) = 0. \tag{5.20}$$

Given a set of targets indexed by $i \in [1\,{:}\,n_{\mathrm{m}}]$, we seek a trajectory from an initial state x_0 to a final state x_f that collects a desired amount of information c_{m_i} for each target i located at position p_{m_i}.

The information acquired at each time instant is modeled as dependent on the target distance, the angle from which the target is observed, and the sensor field of view. We therefore introduce a nonnegative information-acquisition metric, as a function of r and Ω, to quantify the information gathered per unit time. The metric is positive only when all of the following are satisfied:

1. the distance between the sensor and the target is below a prescribed threshold;
2. the sensor lies within the acquisition cone defined by the target position and its desired acquisition direction;

3. the target lies within the field-of-view cone determined by the position and direction of the sensor.

Accordingly, the metric increases as the sensor gets closer to the target and aligns with the centers of the acquisition and field-of-view cones.

To represent these effects, we define three component functions associated with target i, namely m_{d_i}, m_{a_i}, and m_{v_i}; see Figure 5.12. The first is a distance-dependent term,

$$m_{\mathrm{d}_i}\big(r(t)\big) := 1 + c_{\mathrm{m}_i}\left(\left(1 + e^{\tau_{\mathrm{m}_i}\rho_i(t)}\right)^{-1} - 0.5\right), \tag{5.21a}$$

where

$$\rho_i(t) := \|r(t) - p_{\mathrm{m}_i}\|, \tag{5.21b}$$

c_{m_i} is a scaling coefficient, and τ_{m_i} is the acquisition-decay rate for distance.

The second component models acquisition direction via the scaled cosine similarity between the desired acquisition direction of the ith target, d_{m_i}, and the actual acquisition direction $\overline{(r(t) - p_{\mathrm{m}_i})}$:

$$\tilde{m}_{\mathrm{a}_i}\big(r(t)\big) := d_{\mathrm{m}_i}^{\top}\overline{\big(r(t) - p_{\mathrm{m}_i}\big)}, \tag{5.21c}$$

$$m_{\mathrm{a}_i}\big(r(t)\big) := \frac{\tilde{m}_{\mathrm{a}_i}\big(r(t)\big) - \cos(\alpha_{\mathrm{a}_i}^{\mathrm{d}})}{1 - \cos(\alpha_{\mathrm{a}_i}^{\mathrm{d}})}, \tag{5.21d}$$

where $\alpha_{\mathrm{a}_i}^{\mathrm{d}}$ is the acquisition-cone maximum angle.

The third component models sensor field of view via the scaled cosine similarity between the sensor field-of-view direction,

$$d_{\mathrm{v}}(\Omega) := \mathcal{R}(\Omega) d_{\mathrm{v}}^{\mathcal{B}}, \tag{5.21e}$$

and the acquisition direction $\overline{(r(t) - p_{\mathrm{m}_i})}$:

$$\tilde{m}_{\mathrm{v}_i}\big(r(t), \Omega(t)\big) := -d_{\mathrm{v}}\big(\Omega(t)\big)^{\top}\overline{\big(r(t) - p_{\mathrm{m}_i}\big)}, \tag{5.21f}$$

$$m_{\mathrm{v}_i}\big(r(t), \Omega(t)\big) := \frac{\tilde{m}_{\mathrm{v}_i}\big(r(t), \Omega(t)\big) - \cos(\alpha_{\mathrm{v}}^{\mathrm{d}})}{1 - \cos(\alpha_{\mathrm{v}}^{\mathrm{d}})}, \tag{5.21g}$$

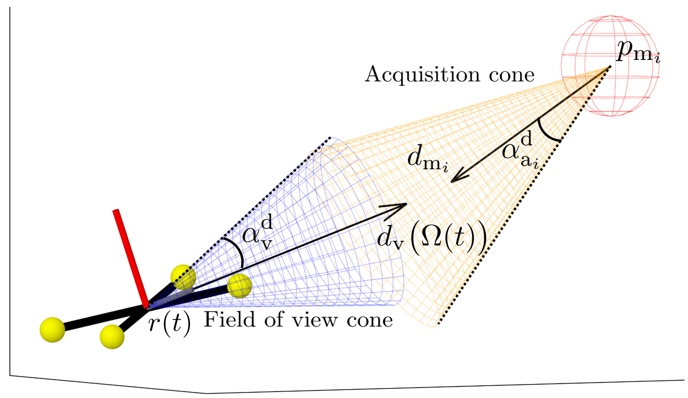


Figure 5.12: Illustration of the key components of the information-acquisition metric, including the range between the agent position r and target position p_{m_i}, the field-of-view direction $d_{\mathrm{v}}(\Omega)$, and the desired acquisition direction d_{m_i}.

where $d_{\mathrm{v}}^{\mathcal{B}}$ is the sensor direction in the body frame and $\alpha_{\mathrm{v}}^{\mathrm{d}}$ is the field-of-view maximum angle.

With proper scaling, the quantities m_{d_i}, m_{a_i}, and m_{v_i} attain a maximum value of 1 and become zero at the boundary of the corresponding sensing limitation. Beyond these boundaries they become negative, which is not meaningful from an information-acquisition standpoint, since it would indicate increasingly negative information acquisition. We therefore compose them with the $\mathcal{C}^1$-smooth function

$$q^c(x) := \Big(\max(0, x)^2 + c^2\Big)^{1/2} - c, \qquad c > 0, \tag{5.22}$$

and define, for simplicity,

$$q^c(m_{\mathrm{j}}) =: m_{\mathrm{j}}^q. \tag{5.23}$$

When applied to vector-valued arguments, the function q^c is understood elementwise. The resulting composed functions under-approximate the original functions in (5.21) by an arbitrarily small amount and are exactly zero beyond the sensing limitations; see Figure 5.13.

The instantaneous information acquisition for target i is then modeled as the product of the modified components,

$$m_i\Big(x(t)\Big) = \bar{m}_i\, m^q_{\mathrm{d}_i}\Big(r(t)\Big)\, m^q_{\mathrm{a}_i}\Big(r(t)\Big)\, m^q_{\mathrm{v}_i}\Big(r(t), \Omega(t)\Big), \tag{5.24}$$

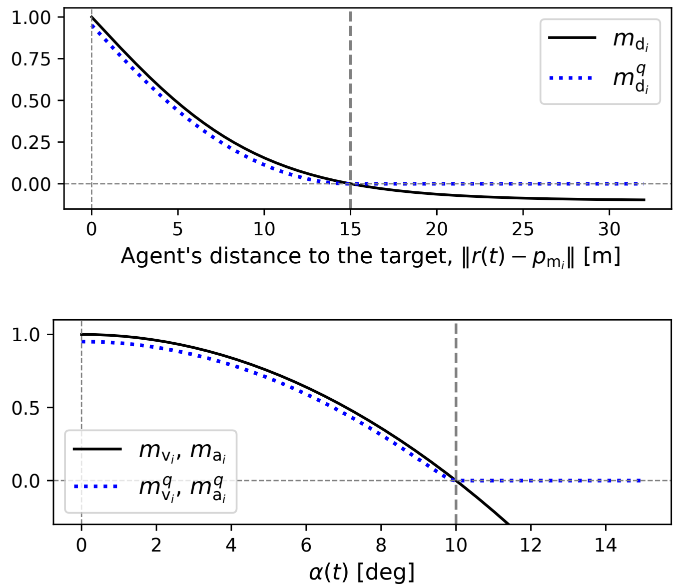


Figure 5.13: Top: m_{d_i} and $m^q_{\mathrm{d}_i}$ for $c = 0.05$, as functions of distance to the target, $\|r - p_{\mathrm{m}_i}\|$, for $c_{\mathrm{m}_i} = 2.2$ and $\tau_{\mathrm{m}_i} = 0.203$. Bottom: m_{a_i}, m_{v_i} and the corresponding $m^q_{\mathrm{a}_i}$, $m^q_{\mathrm{v}_i}$ for $c = 0.05$, as functions of the angle α for $\alpha^{\mathrm{d}}_{\mathrm{a}_i} = 10°$ and $\alpha^{\mathrm{d}}_{\mathrm{v}} = 10°$, respectively.

where $\bar{m}_i$ denotes the maximum achievable acquisition rate for target i. This imposes that information is acquired only if all sensor limitations are satisfied and increases as the trajectory moves farther away from the sensing boundaries. The cumulative acquired information then evolves as

$$\dot{z}(t) = m\Big(x(t)\Big) = \Big(m_1\Big(x(t)\Big), \ldots, m_{n_{\mathrm{m}}}\Big(x(t)\Big)\Big). \tag{5.25}$$

Based on (5.25), we seek a trajectory $x(t)$, $t \in [0, t_{\mathrm{f}}]$, that satisfies the agent dynamics and constraints while collecting a sufficient amount of information for each target:

$$z(0) = 0, \qquad z(t_{\mathrm{f}}) \geq c_{\mathrm{m}} = \Big(c_{\mathrm{m}_1}, c_{\mathrm{m}_2}, \ldots, c_{\mathrm{m}_{n_{\mathrm{m}}}}\Big). \tag{5.26}$$

5.4.3 Driving information acquisition by STL specifications

In the regions where the sensing limitations (5.21) are not satisfied, the information-acquisition metric $m_i(x)$ is uniformly zero, resulting in a uniformly zero gradient that causes convergence issues for gradient-based trajectory generation. To mitigate this, we enforce the existence of time intervals over which the information-acquisition rate and its gradient remain nonzero by imposing signal temporal logic specifications.

We first define the predicates

$$
\begin{aligned}
{}^{\mathrm{d}}\varphi^i &:= \Big(f_{\mathrm{d}_i}(r(t)) \geq 0\Big) := \Big(r^{\mathrm{de}}_{\mathrm{d}_i} - \|r(t) - p_{\mathrm{m}_i}\| \geq 0\Big), && (5.27\text{a})\\
{}^{\mathrm{a}}\varphi^i &:= \Big(f_{\mathrm{a}_i}(r(t)) \geq 0\Big) := \Big(\tilde{m}_{\mathrm{a}_i}(r(t)) - \cos(\alpha^{\mathrm{de}}_{\mathrm{a}_i}) \geq 0\Big), && (5.27\text{b})\\
{}^{\mathrm{v}}\varphi^i &:= \Big(f_{\mathrm{v}_i}(r(t), \Omega(t)) \geq 0\Big) := \Big(\tilde{m}_{\mathrm{v}_i}(r(t), \Omega(t)) - \cos(\alpha^{\mathrm{de}}_{\mathrm{v}}) \geq 0\Big), && (5.27\text{c})
\end{aligned}
$$

where

$$
\alpha^{\mathrm{de}}_{\mathrm{a}_i} < \alpha^{\mathrm{d}}_{\mathrm{a}_i}, \qquad \alpha^{\mathrm{de}}_{\mathrm{v}} < \alpha^{\mathrm{d}}_{\mathrm{v}}.
$$

If (5.27) holds at time t, then the corresponding modified sensing components are nonnegative, and hence the sensing limitations are satisfied at time t.

To ensure that all three sensing conditions hold over a time window of duration $t_{\mathrm{E}_i} > 0$, we impose the STL formula

$$
{}^{\mathrm{g}}\varphi^i := \boldsymbol{G}_{[0, t_{\mathrm{E}_i}]}\Big({}^{\mathrm{d}}\varphi^i \wedge {}^{\mathrm{a}}\varphi^i \wedge {}^{\mathrm{v}}\varphi^i\Big), \tag{5.28}
$$

where $\boldsymbol{G}$ is the *always* operator. We then require this condition to occur sometime within a larger window by enforcing

$$
{}^{\mathrm{f}}\varphi^i := \boldsymbol{F}_{[t_{\mathrm{S}_i}, t_{\mathrm{S}_{i+1}} - t_{\mathrm{E}_i}]}\,{}^{\mathrm{g}}\varphi^i, \tag{5.29}
$$

where $\boldsymbol{F}$ is the *eventually* operator and the interval satisfies

$$
t_{\mathrm{E}_i} \leq t_{\mathrm{S}_{i+1}} - t_{\mathrm{S}_i}.
$$

The interpretation is that, during the horizon assigned to target i, the trajectory must eventually enter and remain within an information-acquisition window for a sufficient duration.

Composing `eventually` and `always` in (5.29) ensures that, for sufficiently small $r_{\mathrm{d}_i}^{\mathrm{de}}$ and with

$$\alpha_{\mathrm{a}_i}^{\mathrm{de}} < \alpha_{\mathrm{a}_i}^{\mathrm{d}}, \qquad \alpha_{\mathrm{v}}^{\mathrm{de}} < \alpha_{\mathrm{v}}^{\mathrm{d}},$$

if (5.29) is satisfied, then there exists

$$t' \in [t_{\mathrm{S}_i}, t_{\mathrm{S}_{i+1}} - t_{\mathrm{E}_i}]$$

such that $m_i(x)$ and its gradient are nonzero for almost every

$$t \in [t', t' + t_{\mathrm{E}_i}].$$

This is precisely the mechanism needed to avoid the flat, zero-gradient regions that otherwise hinder numerical optimization.

5.4.4 Trajectory generation formulation and solution

The trajectory generation problem for information acquisition is formulated as a free-final-time Mayer optimal control problem with additional constraints on cumulative information acquisition and on the satisfaction of the STL specifications:

$$\begin{aligned}
\underset{x(\cdot),u(\cdot),t_{\mathrm{f}}}{\text{minimize}} \quad & L\Big(t_{\mathrm{f}}, x(t_{\mathrm{f}})\Big) \\
\text{subject to} \quad & \dot{x}(t) = F\Big(t, x(t), u(t)\Big), \qquad t \in [0, t_{\mathrm{f}}], \\
& \dot{z}(t) = m\Big(x(t)\Big), \qquad t \in [0, t_{\mathrm{f}}], \\
& g\Big(t, x(t), u(t)\Big) \le 0, \qquad t \in [0, t_{\mathrm{f}}], \\
& h\Big(t, x(t), u(t)\Big) = 0, \qquad t \in [0, t_{\mathrm{f}}], \\
& P\Big(x(0), u(0), x(t_{\mathrm{f}}), u(t_{\mathrm{f}})\Big) \le 0, \\
& Q\Big(x(0), u(0), x(t_{\mathrm{f}}), u(t_{\mathrm{f}})\Big) = 0, \\
& c_{\mathrm{m}} - z(t_{\mathrm{f}}) \le 0, \\
& z(0) = 0, \\
& \Big(x, 0\Big) \models {}^{\mathrm{f}}\varphi^i, \qquad \forall i \in [1 : n_{\mathrm{m}}].
\end{aligned}$$

Here L is the terminal cost, g and h are the path-constraint functions, and P and Q collect the boundary conditions. The additional terminal inequality

$$c_{\mathrm{m}} - z(t_{\mathrm{f}}) \leq 0$$

ensures that the total information acquired for each target exceeds the prescribed threshold.

A key modeling point is that the continuous-time path constraints and the information accumulator are embedded directly into the continuous-time dynamics, while the STL conditions are imposed through the discrete-time GMSR parameterization after discretization. This is the hybrid structure emphasized throughout the present application. Continuous-time path-constraint satisfaction is handled through the augmentation, whereas the mission-level requirements that guarantee useful sensing intervals are enforced via smooth exact discrete-time STL specifications.

To ensure continuous-time satisfaction of the path constraints, we introduce the auxiliary state y and define the augmented state

$$x_{\mathrm{a}}(t) := \Big(x(t), y(t), z(t)\Big),$$

with augmented dynamics

$$\dot{x}_{\mathrm{a}}(t) = F_{\mathrm{a}}\Big(t, x_{\mathrm{a}}(t), u(t)\Big) := \begin{bmatrix} F\Big(t, x(t), u(t)\Big) \\ \mathbf{1}^\top q^c\Big(g\Big(t, x(t), u(t)\Big)\Big) + \mathbf{1}^\top q^d\Big(h\Big(t, x(t), u(t)\Big)\Big) \\ m\Big(x(t)\Big) \end{bmatrix}, \tag{5.31}$$

where

$$q^d(x) := (x^2 + d^2)^{1/2} - d, \qquad d > 0, \tag{5.32}$$

and the periodic boundary condition

$$y(0) = 0, \qquad y(t_{\mathrm{f}}) = 0 \tag{5.33}$$

ensures continuous-time path-constraint satisfaction.

Time-dilation, FOH parameterization and multiple-shooting discretization then follow the machinery of Chapter 2. The free-final-time problem is transformed to a fixed-final-time problem through the time-dilated dynamics

$$\tilde{x}(\tau) := \Big(x_{\mathrm{a}}(\tilde{t}(\tau)), \tilde{t}(\tau)\Big), \qquad \tilde{u}(\tau) := \Big(u(\tilde{t}(\tau)), s(\tau)\Big),$$

and then discretized on a finite node set using multiple shooting.

After time-dilation and discretization, the continuous-time windows

$$[t_{\mathrm{S}_i},\, t_{\mathrm{S}_{i+1}} - t_{\mathrm{E}_i}]$$

and durations t_{E_i} are represented by the corresponding node indices k_{S_i}, $k_{\mathrm{S}_{i+1}}$, and integer window lengths n_{E_i}.

The STL specifications are parameterized using the D-GMSR constructions of Chapter 3. In particular, the inner `always` condition in (5.28) is parameterized over a finite node window as

$$\Gamma^{\mathrm{g}\varphi^i}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(\tilde{x}, k) = {}^{\wedge}h^{c}_{p,w}\Big({}^{\wedge}h^{c}_{p,w}\Big(f_{\mathrm{s}_i}(r_k, \Omega_k)\Big), \ldots, {}^{\wedge}h^{c}_{p,w}\Big(f_{\mathrm{s}_i}(r_{k+n_{\mathrm{E}_i}}, \Omega_{k+n_{\mathrm{E}_i}})\Big)\Big),$$

where

$$f_{\mathrm{s}_i}(r_k, \Omega_k) := \Big(f_{\mathrm{d}_i}(r_k), f_{\mathrm{a}_i}(r_k), f_{\mathrm{v}_i}(r_k, \Omega_k)\Big).$$

Here, r_k and Ω_k denote the values of $r(\tilde{t}(\tau_k))$ and $\Omega(\tilde{t}(\tau_k))$, respectively, and $n_{\mathrm{E}_i} \in \mathbb{Z}_{++}$ is chosen such that

$$\tilde{t}(\tau_{k+n_{\mathrm{E}_i}}) - \tilde{t}(\tau_k) \geq t_{\mathrm{E}_i}.$$

The corresponding outer `eventually` specification in (5.29) is then parameterized exactly as the smooth disjunction

$$\Gamma^{\mathrm{f}\varphi^i}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(\tilde{x}, 0) = {}^{\vee}h^{c}_{p,w}\Big(\Gamma^{\mathrm{g}\varphi^i}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(\tilde{x}, k_{\mathrm{S}_i}), \ldots, \Gamma^{\mathrm{g}\varphi^i}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(\tilde{x}, k_{\mathrm{S}_{i+1}} - n_{\mathrm{E}_i} - 1)\Big),$$

where k_{S_i} and $k_{\mathrm{S}_{i+1}}$ are the node indices defining the interval

$$[k_{\mathrm{S}_i} : k_{\mathrm{S}_{i+1}} - 1],$$

over which the inner `always` specification may start.

This discrete-time STL condition is used to select sensing windows after transcription. In continuous time, the information-acquisition metric is still accumulated through the augmented dynamics, so useful information can be gathered throughout the intervals, not only at the nodes. The role of the STL constraint is to ensure that at least one admissible window contains node points for which the sensing predicates are satisfied, thereby placing the trajectory inside a region where the information-acquisition metric is positive and has nonzero gradient. This avoids the zero-gradient regions that would otherwise prevent the optimizer from discovering informative sensing trajectories.

In the implementation, only the nonnegative side of this outer disjunction is needed to certify satisfaction. For that reason, we use the simpler satisfaction-oriented surrogate

$$\tilde{\Gamma}^{\mathrm{f}\varphi^i}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(\tilde{x},0) = -\prod_{k=k_{\mathrm{S}_i}}^{k_{\mathrm{S}_{i+1}}-n_{\mathrm{E}_i}-1} \sum_{j=0}^{n_{\mathrm{E}_i}} \Big(\mathbf{1}^\top q^c\Big(-f_{\mathrm{s}_i}(r_{k+j},\Omega_{k+j})\Big)\Big), \tag{5.34}$$

where q^c is applied elementwise. This surrogate preserves the relevant sign logic:

$$\tilde{\Gamma}^{\mathrm{f}\varphi^i}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(\tilde{x},0) \geq 0$$

if and only if there exists an admissible start index

$$k \in [k_{\mathrm{S}_i} : k_{\mathrm{S}_{i+1}} - n_{\mathrm{E}_i} - 1]$$

such that

$$f_{\mathrm{s}_i}(r_{k+j},\Omega_{k+j}) \geq 0$$

componentwise for all $j = 0, \ldots, n_{\mathrm{E}_i}$. Hence,

$$\tilde{\Gamma}^{\mathrm{f}\varphi^i}_{\boldsymbol{c},\boldsymbol{p},\boldsymbol{w}}(\tilde{x},0) \geq 0$$

certifies satisfaction of the corresponding `eventually-always` requirement, even though $\tilde{\Gamma}$ need not coincide with the exact GMSR robustness value.

After exact penalization of the nonlinear constraints, the resulting finite-dimensional problem takes the composite form

$$\underset{Z \in \mathcal{Z}}{\text{minimize}} \quad J_{\text{nl}}(Z) := G(Z) + w_{\text{ep}} H(C(Z)), \tag{5.35}$$

where $Z = (\tilde{x}_1, \ldots, \tilde{x}_N, \tilde{u}_1, \ldots, \tilde{u}_N)$, the convex set $\mathcal{Z}$ collects the convex constraints, and G and $H \circ C$ encode the objective, penalized defect constraints, boundary conditions, and the parameterized STL specifications. As in Sections 5.3 and 5.5, this problem is solved by the prox-convex method discussed in Chapter 4, with adaptive proximal weighting.

5.4.5 Case study: monitoring drone

We demonstrate the proposed framework in a case study where a quadrotor must acquire information about six targets. We consider a small quadrotor drone with mass $m = 1$ kg and principal moments of inertia about the body-frame axes

$$(I_x, I_y, I_z) = (0.07, 0.07, 0.127) \text{ kg m}^2.$$

The vehicle is modeled by 6-DoF translational and rotational Euler–Lagrange dynamics, where (r_x, r_y, r_z) is the position vector in the inertial frame, (v_x, v_y, v_z) is the velocity vector in the body frame, (ϕ, θ, ψ) are the roll, pitch, and yaw angles, and (p, q, r) are the corresponding angular rates. The control inputs are thrust T and torques $(\tau_\phi, \tau_\theta, \tau_\psi)$ in the body-frame axes. Angular rates, pitch and roll angles, altitude, velocity, thrust, and torques are all subject to bounds, and the drone must keep a distance of at least 1.5 m from the targets to avoid collisions.

The acquisition-cone directions for the six targets are given in Table 5.2. Some targets use directions in the (x, y) plane, while others use tilted directions.

In the simulation, the sensing model uses

$$c_{\text{m}_i} = 2.2, \qquad \bar{m}_i = 0.38, \qquad i \in [1\!:\!6],$$

with acquisition-decay parameters

$$\tau_{\text{m}} = (0.223,\ 0.063,\ 0.223,\ 0.085,\ 0.223,\ 0.223),$$

Vector	Value
d_{m}	$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} & -\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} & -1 & \frac{1}{\sqrt{3}} & -\frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} & 0 & 0 \end{bmatrix}$

Table 5.2: Acquisition directions for the six targets, arranged as columns of d_{m}.

field-of-view angle

$$\alpha_{\mathrm{v}}^{\mathrm{d}} = 20^\circ,$$

and acquisition-cone angles

$$\alpha_{\mathrm{a}_i}^{\mathrm{d}} = 20^\circ, \qquad i \in [1\!:\!6].$$

The target sequence is assumed to be preassigned, for example by the task-allocation method of [278].

The STL specification (5.29) is implemented with

$$r_{\mathrm{d}}^{\mathrm{de}} = (6,\ 20,\ 6,\ 15,\ 6,\ 6)\ \mathrm{m},$$

$$\alpha_{\mathrm{v}}^{\mathrm{de}} = 15^\circ, \qquad \alpha_{\mathrm{a}_i}^{\mathrm{de}} = 15^\circ, \qquad i \in [1\!:\!6],$$

and

$$k_{\mathrm{S}} = (1,\ 16,\ 25,\ 36,\ 45,\ 50,\ 61), \qquad n_{\mathrm{E}_i} = 3, \qquad i \in [1\!:\!6].$$

The smoothing parameters in (5.22) and (5.32) are set to

$$c = d = 10^{-4},$$

to ensure smoothness with negligible effect on the numerical algorithm.

For the given boundary conditions, we use

$$N = 61$$

node points and treat the flight time as a decision variable to be minimized. The optimization is initialized with a trajectory composed of straight-line segments connecting the initial position to the target sequence and then to the final position, with node points distributed uniformly along this path. Affine scaling is also applied to improve numerical conditioning. The implementation uses CVXPY [81] together with the Clarabel solver [124].

Figure 5.14 shows the optimized monitoring trajectory. Figures 5.15 and 5.16 show, respectively, the information-acquisition history for each target and the associated state and control trajectories. The quadrotor acquires the required information for each target in less than 51 s while satisfying the state and input constraints continuously during the flight.

For comparison, a conventional method that decouples motion planning and information acquisition, where the agent repeatedly flies to a point, stops to acquire information, and then proceeds to the next, results in a total mission time of 66.17 s, comprising 42.31 s for motion and 23.86 s for acquisition. This is approximately 29% longer than the proposed approach. In addition, if the sensor field-of-view and acquisition-direction cones are too narrow, such a decoupled strategy may fail altogether for targets with steeply tilted acquisition directions, since acquisition is then impossible without drone pitching, which prevents the flight from being stationary.

The computation of the trajectory in a simple, nonoptimized Python implementation takes approximately 14.5 s on a 2023 MacBook Pro. Speedups of 10×–40× are expected through code optimization, compilation, and translation to C/C++. Thus, although the proposed method is primarily intended for offline trajectory generation, a receding-horizon implementation with a shorter time window and fewer nodes may be able to achieve real-time operation.

5.5 Rocket landing with compound state-triggered constraints

This section applies the modeling and optimization machinery developed in the previous chapters to a free-final-time powered-descent guidance problem with six-degree-of-freedom rocket dynamics and compound state-triggered constraints, a subset of STL specifications. In

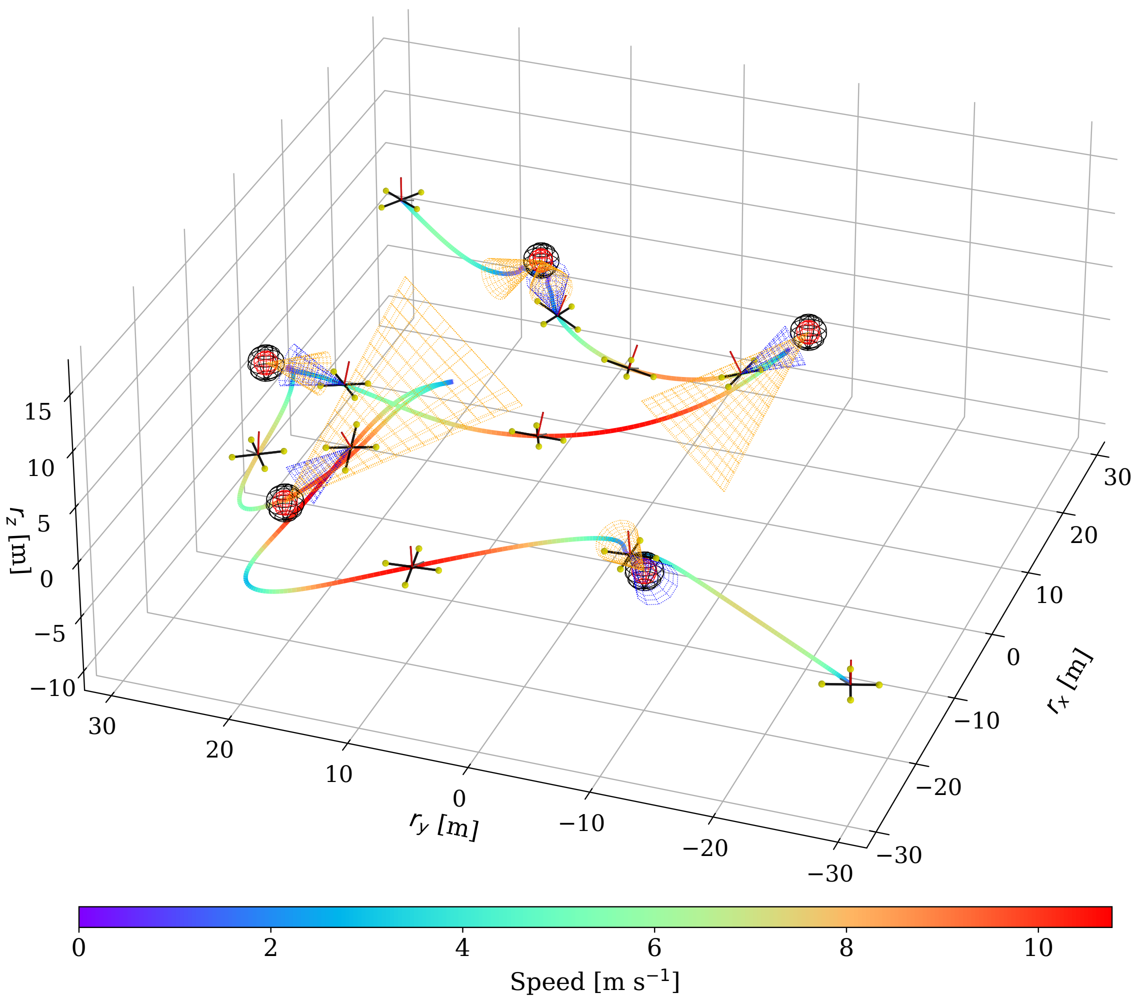


Figure 5.14: Trajectory generation for a quadrotor acquiring information from six targets. The trajectory color represents speed, as indicated by the colorbar. Snapshots show the quadrotor at different time instants together with targets (red spheres), collision areas (black spheres), acquisition cones (yellow cones), and sensor fields of view during information acquisition (blue cones).

contrast to the receding-horizon setting of Section 5.3 and the sensing-driven information-acquisition problem of Section 5.4, the emphasis here is on embedding mission specifications directly into a continuous-time trajectory optimization problem for a highly nonlinear

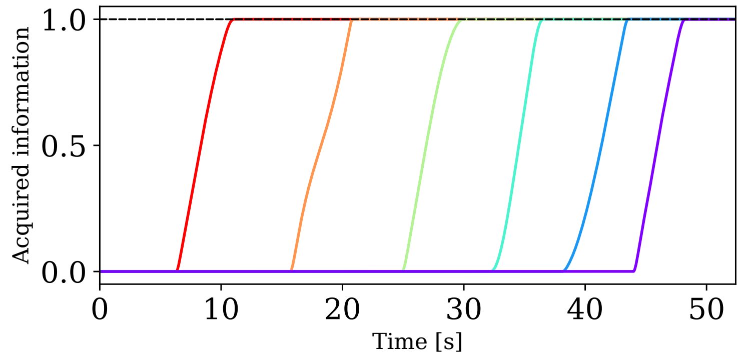


Figure 5.15: History of information acquired during the quadrotor flight for each target, from red (first target) to purple (last target).

aerospace vehicle. The continuous-time state and control bounds are handled through the path-constraint machinery of Chapters 2 and 3, while the compound state-triggered constraints are parameterized through the smooth and exact logical constructions of Chapter 3 and solved using the convexification framework developed in Chapter 4. This example therefore illustrates how the general framework can accommodate free-final-time 6-DoF dynamics together with implication, conjunction, and disjunction structures that arise naturally in powered-descent missions.

5.5.1 Introduction

Autonomous guidance algorithms are an enabling technology for future manned and unmanned planetary missions. Powered-descent guidance (PDG) involves generating reference trajectories and feedforward control commands to facilitate the soft landing of a rocket-powered vehicle on a planetary surface, playing an essential role in enabling human exploration missions to destinations within the solar system [60]. Polynomial guidance methods developed for PDG [200, 261, 263] have played a crucial role in the Apollo program for the landing of the lunar module on the Moon [158] as well as in the Mars Science Laboratory (MSL) and Mars 2020 missions for the landing of the Curiosity and the Perseverance rovers on Mars

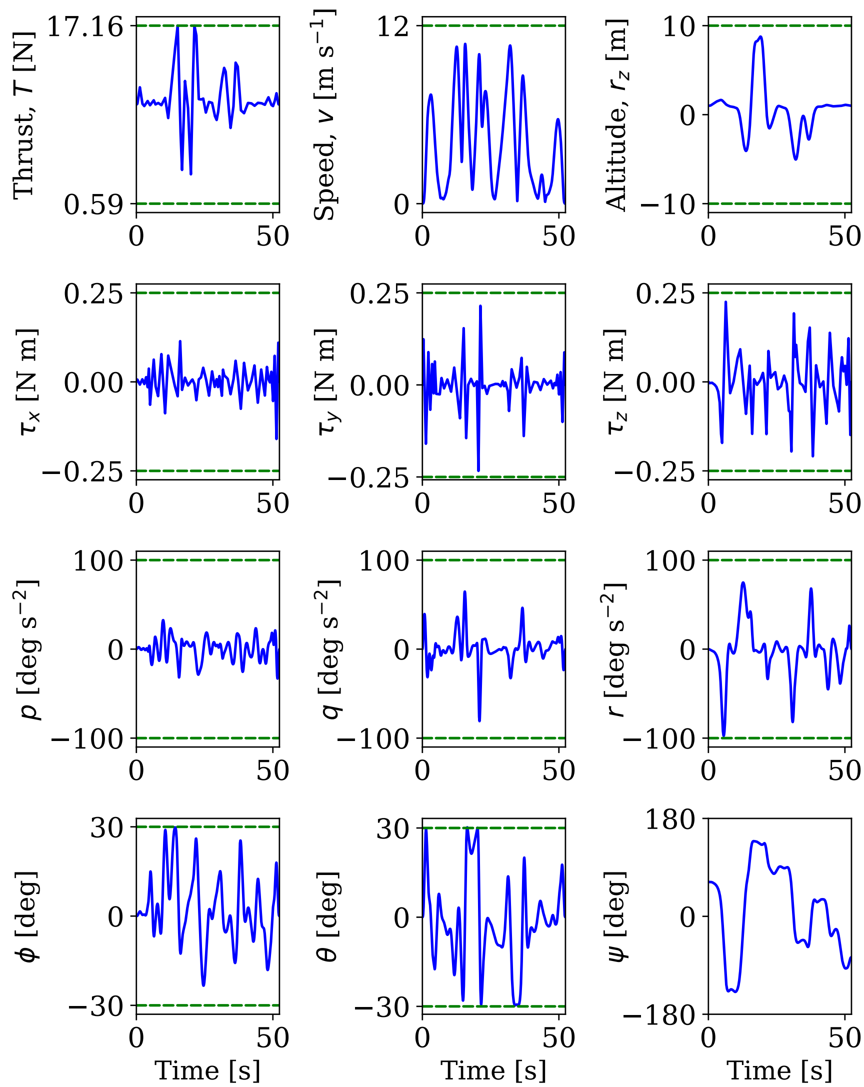


Figure 5.16: History of state and control variables during the quadrotor flight (blue solid) together with the corresponding constraints (green dashed).

[252, 297]. While polynomial guidance methods have achieved remarkable success for PDG in space missions, the requirements for precise landing in challenging environments and the imposition of complex mission constraints have driven the development of more advanced PDG algorithms.

Advancements in numerical optimization algorithms have enabled the formulation and solution of the PDG problem as an optimal control problem. The effectiveness of optimization algorithms on the PDG problem has been demonstrated in the landings of vertical-takeoff/vertical-landing reusable space rockets [33]. The development of interior-point methods [118, 214] has enabled the solving of convex optimization problems [42] to global optimality, ensuring convergence within polynomial time. Lossless convexification (LCvx) is a transformative method that applies convex optimization algorithms to solve the PDG problem [6]. Through the introduction of a slack variable, the nonconvex thrust-magnitude lower-bound constraint in the three-degree-of-freedom (3-DoF) PDG problem is transformed into a convex one. By leveraging Pontryagin's maximum principle, the convex problem's optimal solution is also optimal for the original nonconvex problem. LCvx is subsequently extended to address a variety of constraints, including general nonconvex input sets [5], minimum-error landing [34], thrust-pointing constraints [2], affine and quadratic state constraints [130, 131], nonlinear dynamics [35], and binary constraints [180]. Customized second-order cone programming solvers [95, 96] are developed to assess the real-time performance of LCvx, while the Guidance for Fuel Optimal Large Divert (G-FOLD) algorithm [7, 255, 256] was specifically developed to test LCvx on Masten Space Systems' vertical-takeoff/vertical-landing suborbital rocket, Xombie [209, 210]. Although considering only translation dynamics and constraints related either to translation or to the thrust vector via a 3-DoF model achieved sufficient success to solve the PDG problem, six-degree-of-freedom (6-DoF) modeling of vehicle dynamics and increasing complexity of mission constraints necessitate the solution of more general nonconvex problems.

Sequential convex programming (SCP) has emerged as an effective method for transitioning from fully convex 3-DoF PDG problems to more complex, nonconvex 6-DoF PDG problems, which often include nonconvex constraints [183, 184]. SCP is an iterative method that handles nonconvex problems by successively solving a sequence of locally convex approximations, achieved through linearizing the nonconvexities [40, 61, 92, 185, 237]. Therefore, SCP algorithms enable the handling of more general formulations of vehicle dynamics and mission constraints, which often introduce nonconvexities, such as aerodynamic forces on the vehicle [269], free-final-time formulation [268], multi-phase landing [144], state-triggered constraints [238, 273, 274], and integer constraints [179, 182]. The SCP algorithm has been further employed to solve quadrotor flight [270–272] and hypersonic entry problems [192–194], and its real-time capability has been demonstrated in quadrotor flight [270–272] as well as PDG problems [181, 237]. An SCP-based solution method utilizing the dual quaternion formulation for the PDG problem [238] has been selected as the candidate PDG algorithm for NASA's Safe and Precise Landing – Integrated Capabilities Evolution (SPLICE) project. This algorithm has been flight-tested in an open-loop configuration on the Blue Origin New Shepard suborbital rocket [87, 266]. The traditionally utilized interior-point method to solve each convex subproblem in these algorithms is replaced with the customized first-order proportional-integral projected gradient (PIPG) algorithm [307, 308] for addressing the 3-DoF PDG problem via LCvx, leading to a significant improvement in the solution speed of the problem [103]. The PIPG algorithm is then employed to solve the 6-DoF PDG problem formulated via dual quaternion, resulting in a significant enhancement in real-time performance [143].

In all of these algorithms, path constraints are imposed only at the node points of the discretized continuous-time problem, so inter-sample constraint satisfaction is not guaranteed. The continuous-time successive convexification framework (CT-SCVx), first developed for continuous-time path constraints and reviewed in Chapters 2 and 3, combines continuous-time path-constraint reformulation, time-dilation, multiple shooting, exact penalties, and a convergence-guaranteed SCP scheme to solve the resulting finite-dimensional nonconvex optimal control problem [104]. Implementations of this framework have been demonstrated for passively safe [105] and GPU-accelerated [66] trajectory optimization for 6-DoF PDG, nonlinear model predictive control for obstacle avoidance [287], and trajectory optimization for 6-DoF aircraft approach and landing [155].

While formulations exist to impose state-triggered constraints [238, 273, 274] and integer constraints [179, 182], a general formulation has not yet been addressed for imposing temporal logic specifications by ensuring their continuous-time satisfaction within an SCP framework. Signal temporal logic (STL) provides a general framework for imposing temporal and logical specifications on optimization problems [27, 89, 178]; however, the standard robustness semantics involve nested min and max operators, leading either to mixed-integer formulations [233, 247] or to smooth approximations that compromise soundness, completeness, or numerical robustness [123, 128, 187, 223, 292]. Moreover, the direct use of min and max can give rise to locality and masking effects [196], which can prevent local optimization algorithms from finding feasible solutions even when the problem is feasible. Chapter 3 addressed these issues through the generalized mean-based smooth robustness (GMSR) formulation [286], which is smooth, sound, complete, and equipped with favorable gradient properties for numerical optimization.

The present section applies that modeling machinery to a 6-DoF rocket landing problem with compound state-triggered requirements. In particular, we employ the exact smooth logical constructions of Chapter 3 to parameterize the compound state-triggered constraints, and we use the numerical optimal-control machinery of Chapter 2 and Chapter 4 to solve the resulting powered descent guidance problem while ensuring continuous-time constraint satisfaction and convergence.

5.5.2 6-DoF powered descent guidance problem with CT-cSTCs

This subsection presents the 6-DoF powered descent guidance problem with continuous-time compound state-triggered constraints (CT-cSTCs). We first outline the rocket dynamics, general state and control constraints, and boundary conditions, adapted from [143, 274]. We then describe the altitude-triggered gimbal-deflection, speed, angular-velocity, tilt-angle, glideslope, and line-of-sight constraints, together with speed- and tilt-triggered thrust constraints.

Rocket dynamics

The state vector and the control input of the vehicle are defined as

$$x := (m, r_{\mathcal{I}}, v_{\mathcal{I}}, q_{\mathcal{B}\leftarrow\mathcal{I}}, \omega_{\mathcal{B}}),$$
$$u := (T, \delta^{\mathrm{e}}, \phi^{\mathrm{e}}, \delta^{\mathrm{b}}, \phi^{\mathrm{b}}),$$

where subscripts $\mathcal{I}$ and $\mathcal{B}$ denote quantities expressed in the inertial and body-fixed reference frames, respectively; $m \in \mathbb{R}_+$ is the mass of the vehicle; $r_{\mathcal{I}} \in \mathbb{R}^3$ is the position vector; $v_{\mathcal{I}} \in \mathbb{R}^3$ is the velocity vector; $q_{\mathcal{B}\leftarrow\mathcal{I}} = (q_1, q_2, q_3, q_4) \in \mathbb{R}^4$ is the quaternion parameterizing the transformation from the inertial frame to the body frame; $\omega_{\mathcal{B}} \in \mathbb{R}^3$ is the angular-velocity vector; and T, δ^{e}, ϕ^{e}, δ^{b}, and ϕ^{b} denote the thrust magnitude, engine gimbal deflection and azimuth angles, and boresight gimbal deflection and azimuth angles, respectively.

The dynamical system of the rocket is given by

$$\begin{aligned}\dot{x}(t) &= F\Big(x(t), u(t)\Big) \\ &= \begin{bmatrix} -\alpha_{\dot{m}} \|T_{\mathcal{B}}(t)\|_2 \\ v_{\mathcal{I}}(t) \\ \dfrac{1}{m(t)} C_{\mathcal{I}\leftarrow\mathcal{B}}(t)\Big(T_{\mathcal{B}}(t) + A_{\mathcal{B}}(t)\Big) + g_{\mathcal{I}} \\ \dfrac{1}{2}\Omega\Big(\omega_{\mathcal{B}}(t)\Big) q_{\mathcal{B}\leftarrow\mathcal{I}}(t) \\ J_{\mathcal{B}}(t)^{-1}\Big([r_{\mathrm{cm},\mathcal{B}}\times] T_{\mathcal{B}}(t) + [r_{\mathrm{cp},\mathcal{B}}\times] A_{\mathcal{B}}(t) - \omega_{\mathcal{B}}(t) \times \Big(J_{\mathcal{B}}(t)\omega_{\mathcal{B}}(t)\Big)\Big) \end{bmatrix}. \end{aligned} \tag{5.36}$$

In this model,

$$\alpha_{\dot{m}} = \frac{1}{I_{\mathrm{sp}} g_0}, \qquad g_{\mathcal{I}} := (0, 0, -g_0),$$

where I_{sp} is the vacuum specific impulse of the engine and g_0 is standard Earth gravity. The moment-of-inertia matrix of the vehicle about its center of mass at time t is denoted by $J_{\mathcal{B}}(t) \in \mathbb{R}_+^{3\times 3}$. The thrust moment arm, which is the vector from the vehicle's center of mass to the engine gimbal hinge point, is denoted by $r_{\mathrm{cm},\mathcal{B}} \in \mathbb{R}^3$. The aerodynamic moment arm, which is the vector from the vehicle's center of mass to the center of pressure, is denoted by $r_{\mathrm{cp},\mathcal{B}} \in \mathbb{R}^3$.

The thrust vector

$$T_{\mathcal{B}} \in \mathbb{R}^3$$

is defined by

$$T_{\mathcal{B}} := \begin{bmatrix} T\sin(\delta^{\mathrm{e}})\cos(\phi^{\mathrm{e}}) \\ T\sin(\delta^{\mathrm{e}})\sin(\phi^{\mathrm{e}}) \\ T\cos(\delta^{\mathrm{e}}) \end{bmatrix},$$

where T is the thrust magnitude and δ^{e}, ϕ^{e} are the gimbal deflection and azimuth angles of the engine.

The aerodynamic force $A_{\mathcal{B}}$ is given by

$$A_{\mathcal{B}}(t) = -\frac{1}{2}\rho\|v_{\mathcal{I}}(t)\|_2 S_A C_A C_{\mathcal{B}\leftarrow\mathcal{I}}(t) v_{\mathcal{I}}(t),$$

where ρ is the ambient atmospheric density, $S_A \in \mathbb{R}_{++}$ is the reference area, $C_A \in \mathbb{R}^{3\times 3}_{++}$ is the aerodynamic coefficient matrix, and $C_{\mathcal{B}\leftarrow\mathcal{I}} = C^{\top}_{\mathcal{I}\leftarrow\mathcal{B}} \in SO(3)$ is the direction-cosine matrix that encodes the attitude transformation from the inertial frame to the body frame. The matrix $C_{\mathcal{B}\leftarrow\mathcal{I}}$ is defined by

$$C_{\mathcal{B}\leftarrow\mathcal{I}} = \begin{bmatrix} 1-2(q_3^2+q_4^2) & 2(q_2q_3+q_1q_4) & 2(q_2q_4-q_1q_3) \\ 2(q_2q_3-q_1q_4) & 1-2(q_2^2+q_4^2) & 2(q_3q_4+q_1q_2) \\ 2(q_2q_4+q_1q_3) & 2(q_3q_4-q_1q_2) & 1-2(q_2^2+q_3^2) \end{bmatrix}.$$

The skew-symmetric matrix operator $\Omega(\cdot)$ and the vector cross-product matrix $[\cdot\times]$ are defined by

$$\Omega(\xi) := \begin{bmatrix} 0 & -\xi_1 & -\xi_2 & -\xi_3 \\ \xi_1 & 0 & \xi_3 & -\xi_2 \\ \xi_2 & -\xi_3 & 0 & \xi_1 \\ \xi_3 & \xi_2 & -\xi_1 & 0 \end{bmatrix}, \qquad [\xi\times] := \begin{bmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{bmatrix},$$

for $\xi = (\xi_1, \xi_2, \xi_3) \in \mathbb{R}^3$.

General state and control constraints

The vehicle's fuel consumption, tilt angle θ, angular rate $\omega_{\mathcal{B}}$, and glideslope angle γ are subject to the constraints

$$\begin{aligned} m_{\text{dry}} &\leq m(t), \\ \cos(\theta_{\max}) &\leq 1 - 2\Big(q_2^2(t) + q_3^2(t)\Big), \\ \|\omega_{\mathcal{B}}(t)\|_2 &\leq \omega_{\max}, \\ \tan(\gamma_{\max})\Big\|[e_1\ e_2]^\top r_{\mathcal{I}}(t)\Big\|_2 &\leq e_3^\top r_{\mathcal{I}}(t), \end{aligned}$$

where e_i denotes the ith basis vector of $\mathbb{R}^3$.

Additionally, the gimbal deflection and azimuth angles of the engine and the boresight vector are constrained by

$$\begin{aligned} \|\delta^{\text{e}}(t)\|_1 &\leq \delta^{\text{e}}_{\max}, & \|\delta^{\text{b}}(t)\|_1 &\leq \delta^{\text{b}}_{\max}, \\ \|\phi^{\text{e}}(t)\|_1 &\leq \phi^{\text{e}}_{\max}, & \|\phi^{\text{b}}(t)\|_1 &\leq \phi^{\text{b}}_{\max}. \end{aligned}$$

Since the thrust force is itself regulated by state-triggered constraints, it is not explicitly constrained in this subsection. For clarity, we represent the state and control constraints as

$$g_x(x) \leq 0_{4\times 1}, \qquad g_u(u) \leq 0_{4\times 1},$$

where

$$g_x : \mathbb{R}^{n_x} \to \mathbb{R}^4, \qquad g_u : \mathbb{R}^{n_u} \to \mathbb{R}^4,$$

and both g_x and g_u are convex componentwise.

Boundary conditions

The boundary conditions are

$$\begin{aligned} m(0) = m_{\text{i}}, \qquad m(t_{\text{f}}) \geq m_{\text{dry}}, \qquad r_{\mathcal{I}}(0) = r_{\text{i}}, \qquad r_{\mathcal{I}}(t_{\text{f}}) = r_{\text{f}}, \\ v_{\mathcal{I}}(0) = v_{\text{i}}, \qquad v_{\mathcal{I}}(t_{\text{f}}) = v_{\text{f}}, \qquad q_{\mathcal{B}\leftarrow\mathcal{I}}(0) = q_{\mathcal{B}\leftarrow\mathcal{I}_{\text{i}}}, \qquad q_{\mathcal{B}\leftarrow\mathcal{I}}(t_{\text{f}}) = q_{\mathcal{B}\leftarrow\mathcal{I}_{\text{f}}}, \\ \omega_{\mathcal{B}}(0) = \omega_{\mathcal{B}_{\text{i}}}, \qquad \omega_{\mathcal{B}}(t_{\text{f}}) = \omega_{\mathcal{B}_{\text{f}}}. \end{aligned}$$

For clarity, these boundary conditions are represented compactly as

$$\big(x(0), x(t_\mathrm{f})\big) \in \mathcal{X},$$

where $\mathcal{X}$ is a closed convex set.

Compound state-triggered constraints

The altitude-triggered gimbal-deflection angle, speed, angular-velocity, tilt-angle, and glideslope constraints are defined as

$$\begin{aligned}\left(e_3^\top r_\mathcal{I}(t) < h_1^\mathrm{trig}\right) \implies \left(\delta(t) \le \delta^\mathrm{stc}\right) &\wedge \left(\|v_\mathcal{I}(t)\|_2 \le v_\mathcal{I}^\mathrm{stc}\right)\\ &\wedge \left(\|\omega_\mathcal{B}(t)\|_2 \le \omega^\mathrm{stc}\right)\\ &\wedge \left(\cos(\theta^\mathrm{stc}) \le 1 - 2\big(q_2^2(t) + q_3^2(t)\big)\right)\\ &\wedge \left(\tan(\gamma^\mathrm{stc})\Big\|[e_1\ e_2]^\top r_\mathcal{I}(t)\Big\|_2 \le e_3^\top r_\mathcal{I}(t)\right).\end{aligned}$$

The altitude-triggered line-of-sight angle constraint is defined as

$$\left(e_3^\top r_\mathcal{I}(t) < h_2^\mathrm{trig}\right) \implies \left(\cos\left(\psi^\mathrm{stc}\right)\|r_\mathcal{I}(t) - r_\mathrm{f}\|_2 \le \big(r_\mathcal{I}(t) - r_\mathrm{f}\big)^\top C_{\mathcal{I}\leftarrow\mathcal{B}}(t)\ell_\mathcal{B}(t)\right),$$

where ψ^stc is the view-cone angle and the boresight vector in the body frame is

$$\ell_\mathcal{B}(t) := \begin{bmatrix}\sin\left(\delta^\mathrm{b}(t)\right)\cos\left(\phi^\mathrm{b}(t)\right)\\ \sin\left(\delta^\mathrm{b}(t)\right)\sin\left(\phi^\mathrm{b}(t)\right)\\ \cos\left(\delta^\mathrm{b}(t)\right)\end{bmatrix}.$$

The speed- and tilt-triggered thrust constraints are

$$\left(\|v_\mathcal{I}(t)\|_2 < v_\mathcal{I}^\mathrm{trig}\right) \wedge \left(\cos(\theta^\mathrm{trig}) < 1 - 2\big(q_2^2(t) + q_3^2(t)\big)\right) \implies \left(T_\mathrm{min}^{\mathrm{stc}_1} \le T(t)\right) \wedge \left(T(t) \le T_\mathrm{max}^{\mathrm{stc}_1}\right),$$

and, if either the speed or the tilt angle exceeds the specified threshold,

$$\left(v_\mathcal{I}^\mathrm{trig} < \|v_\mathcal{I}(t)\|_2\right) \vee \left(1 - 2\big(q_2^2(t) + q_3^2(t)\big) < \cos(\theta^\mathrm{trig})\right) \implies \left(T_\mathrm{min}^{\mathrm{stc}_2} \le T(t)\right) \wedge \left(T(t) \le T_\mathrm{max}^{\mathrm{stc}_2}\right).$$

For clarity, the state-triggered constraints are represented compactly as

$$\Big(g^{\text{trig}_1}(x) < 0\Big) \implies \bigwedge_{i=1}^{5} \Big(g^{\text{stc}_i}(x) \le 0\Big), \tag{5.37a}$$

$$\Big(g^{\text{trig}_2}(x) < 0\Big) \implies \Big(g^{\text{stc}_6}(u) \le 0\Big), \tag{5.37b}$$

$$\Big(g^{\text{trig}_3}(x) < 0\Big) \wedge \Big(g^{\text{trig}_4}(x) < 0\Big) \implies \Big(g^{\text{stc}_7}(u) \le 0\Big) \wedge \Big(g^{\text{stc}_8}(u) \le 0\Big), \tag{5.37c}$$

$$\Big(-g^{\text{trig}_3}(x) < 0\Big) \vee \Big(-g^{\text{trig}_4}(x) < 0\Big) \implies \Big(g^{\text{stc}_9}(u) \le 0\Big) \wedge \Big(g^{\text{stc}_{10}}(u) \le 0\Big). \tag{5.37d}$$

5.5.3 Smooth parameterization and trajectory optimization formulation

Following Chapter 3, the compound state-triggered constraints (5.37) can be parameterized exactly and smoothly using the GMSR logical constructions. Since only logical operators appear here, the sign-equivalent algebraic constraints can be written directly in terms of nonnegative products and sums of positive-part functions. Applying the exact conjunction and disjunction reductions from Chapter 3 yields

$$\max\Big(0, -g^{\text{trig}_1}(x)\Big)^2 \left(\sum_{i=1}^{5} \max\Big(0, g^{\text{stc}_i}(x)\Big)^2\right) = 0,$$

$$\max\Big(0, -g^{\text{trig}_2}(x)\Big)^2 \max\Big(0, g^{\text{stc}_6}(u)\Big)^2 = 0,$$

$$\max\Big(0, -g^{\text{trig}_3}(x)\Big)^2 \max\Big(0, -g^{\text{trig}_4}(x)\Big)^2 \left(\max\Big(0, g^{\text{stc}_7}(u)\Big)^2 + \max\Big(0, g^{\text{stc}_8}(u)\Big)^2\right) = 0,$$

$$\left(\max\Big(0, g^{\text{trig}_3}(x)\Big)^2 + \max\Big(0, g^{\text{trig}_4}(x)\Big)^2\right)\left(\max\Big(0, g^{\text{stc}_9}(u)\Big)^2 + \max\Big(0, g^{\text{stc}_{10}}(u)\Big)^2\right) = 0.$$

Remark 5.2. Because all functions in (5.38) are nonnegative, these constraints do not require an exterior penalty function when they are embedded into the continuous-time path-constraint reformulation. This is a particularly convenient feature of the present logical subset.

For clarity, we denote the vector of smooth exact state-triggered constraints by

$$h_{\text{stc}}\Big(x(t), u(t)\Big) = 0_{4\times 1}, \qquad h_{\text{stc}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^4_+.$$

The resulting free-final-time powered descent guidance problem can then be written as

$$
\begin{aligned}
\underset{t_{\mathrm{f}},\,x,\,u}{\text{minimize}} \quad & t_{\mathrm{f}} \\
\text{subject to} \quad & \dot{x}\big(t\big) = F\big(x(t), u(t)\big), \quad t \in [0, t_{\mathrm{f}}], \\
& g_x\big(x(t)\big) \leq 0_{4\times 1}, \qquad t \in [0, t_{\mathrm{f}}], \\
& g_u\big(u(t)\big) \leq 0_{4\times 1}, \qquad t \in [0, t_{\mathrm{f}}], \\
& h_{\mathrm{stc}}\big(x(t), u(t)\big) = 0_{4\times 1}, \quad t \in [0, t_{\mathrm{f}}], \\
& \big(x(0), x(t_{\mathrm{f}})\big) \in \mathcal{X}.
\end{aligned}
$$

To ensure continuous-time satisfaction of the path constraints, we follow the construction reviewed in Chapter 3. Introduce the augmented state

$$x_{\mathrm{a}} := (x, y)$$

with auxiliary dynamics

$$
\begin{aligned}
\dot{x}_{\mathrm{a}}(t) =& F_{\mathrm{a}}\big(x_{\mathrm{a}}(t), u(t)\big), \\
:=& \begin{bmatrix} F\big(x(t), u(t)\big) \\ 1_{4\times 1}^{\top} q_c\Big(g_x\big(x(t)\big)\Big) + 1_{4\times 1}^{\top} q_c\Big(g_u\big(u(t)\big)\Big) + 1_{4\times 1}^{\top} h_{\mathrm{stc}}\big(x(t), u(t)\big) \end{bmatrix}
\end{aligned}
$$

where

$$q_c(s) = \max(0, s)^2$$

is applied elementwise. The periodic boundary condition

$$y(0) = y(t_{\mathrm{f}}) \tag{5.40}$$

then certifies continuous-time satisfaction of the original state, control, and compound state-triggered path constraints.

Accordingly, the reformulated continuous-time optimal control problem becomes

$$\begin{aligned} \underset{t_{\mathrm{f}},\,x_{\mathrm{a}},\,u}{\text{minimize}} \quad & t_{\mathrm{f}} \\ \text{subject to} \quad & \dot{x}_{\mathrm{a}}(t) = F_{\mathrm{a}}\Big(x_{\mathrm{a}}(t), u(t)\Big), \quad t \in [0, t_{\mathrm{f}}], \\ & {}^{y}E\Big(x_{\mathrm{a}}(t_{\mathrm{f}}) - x_{\mathrm{a}}(0)\Big) = 0, \\ & \Big({}^{x}Ex_{\mathrm{a}}(0), {}^{x}Ex_{\mathrm{a}}(t_{\mathrm{f}})\Big) \in \mathcal{X}, \end{aligned}$$

where ${}^{x}E$ and ${}^{y}E$ extract the components corresponding to x and y, respectively.

From this point onward, the free-final-time transcription follows the common numerical machinery already established in Chapter 2 and Chapter 4. In particular, the problem is converted to a fixed normalized horizon via time-dilation, the physical controls are parameterized by first-order hold, the dynamics are transcribed via multiple shooting, and the remaining nonlinear equalities are transferred into the objective through exact penalties. In the present specialization, the objective and all constraints other than the dynamics are linear after reformulation, so only the dynamic constraints require exact penalization. The resulting finite-dimensional nonconvex problem is then solved by the prox-convex algorithm.

A convenient compact representation of the penalized finite-dimensional problem is

$$\underset{Z \in \mathcal{Z}}{\text{minimize}} \quad G(Z) + w_{\mathrm{eq}}^{\mathrm{dyn}} H\Big(C(Z)\Big),$$

where Z stacks the nodal states and controls, $\mathcal{Z}$ collects the convex constraints, G is a proper convex function containing the discretized final-time objective and convex side constraints, and $H \circ C$ is a convex-composite exact-penalty term associated with the multiple-shooting defects.

At SCP iteration $j+1$, the corresponding prox-convex convex subproblem takes the form

$$\underset{Z \in \mathcal{Z}}{\text{minimize}} \quad G(Z) + w_{\mathrm{eq}}^{\mathrm{dyn}} H\Big(C(Z^j) + \nabla C(Z^j)(Z - Z^j)\Big) + \frac{w_{\mathrm{prox}}^{j}}{2}\|Z - Z^j\|_2^2,$$

where Z^j is the previous SCP iterate and w_{prox}^{j} is the adaptive proximal weight. The exact interval sensitivities entering the linearized defect map are computed through the standard variational equations of the multiple-shooting flow map; see Chapter 2 and Chapter 4, as well as [66, 104], for the details of this construction. In the original implementation, CVXPY together with ECOS or MOSEK is used for modeling and solving the convex subproblems [81, 88, 208]. Customizable conic solvers such as the proportional-integral projected gradient (PIPG) algorithm [307, 308], especially when combined with conditioning and preconditioning techniques [146], could also be used for efficient implementations.

5.5.4 Numerical results

The solution to the powered descent guidance problem with continuous-time compound state-triggered constraints is illustrated in Figures 5.17 and 5.18. The system and simulation parameters are presented in Table 5.3. To improve the performance of the optimizer, affine scaling is applied to the state and control inputs. Initial guesses are set as follows: the state is interpolated linearly between initial and final values, the thrust force is set to

$$\frac{1}{2}\Big(T_{\max}^{\mathrm{stc}_1} + T_{\min}^{\mathrm{stc}_1}\Big)$$

throughout the flight, the gimbal deflection and azimuth angles are set to zero, and the free-final-time is initialized at 21 s.

At approximately 5.72 s, the vehicle's speed and tilt angle drop below the specified thresholds,

$$v_{\mathcal{I}}^{\mathrm{trig}} \qquad \text{and} \qquad \theta^{\mathrm{trig}},$$

with the thrust constraints satisfied immediately afterward. Around 11.07 s, the altitude decreases below

$$h_2^{\mathrm{trig}},$$

triggering satisfaction of the line-of-sight constraint. Finally, the altitude decreases below

$$h_1^{\mathrm{trig}}$$

around 15 s, followed by satisfaction of the gimbal-deflection angle, speed, angular-velocity, tilt-angle, and glideslope constraints. All conditions are met between consecutive node points, and the corresponding constraints are satisfied in continuous time immediately afterward. The implementation is available at `https://github.com/UW-ACL/CT-cSTC`.

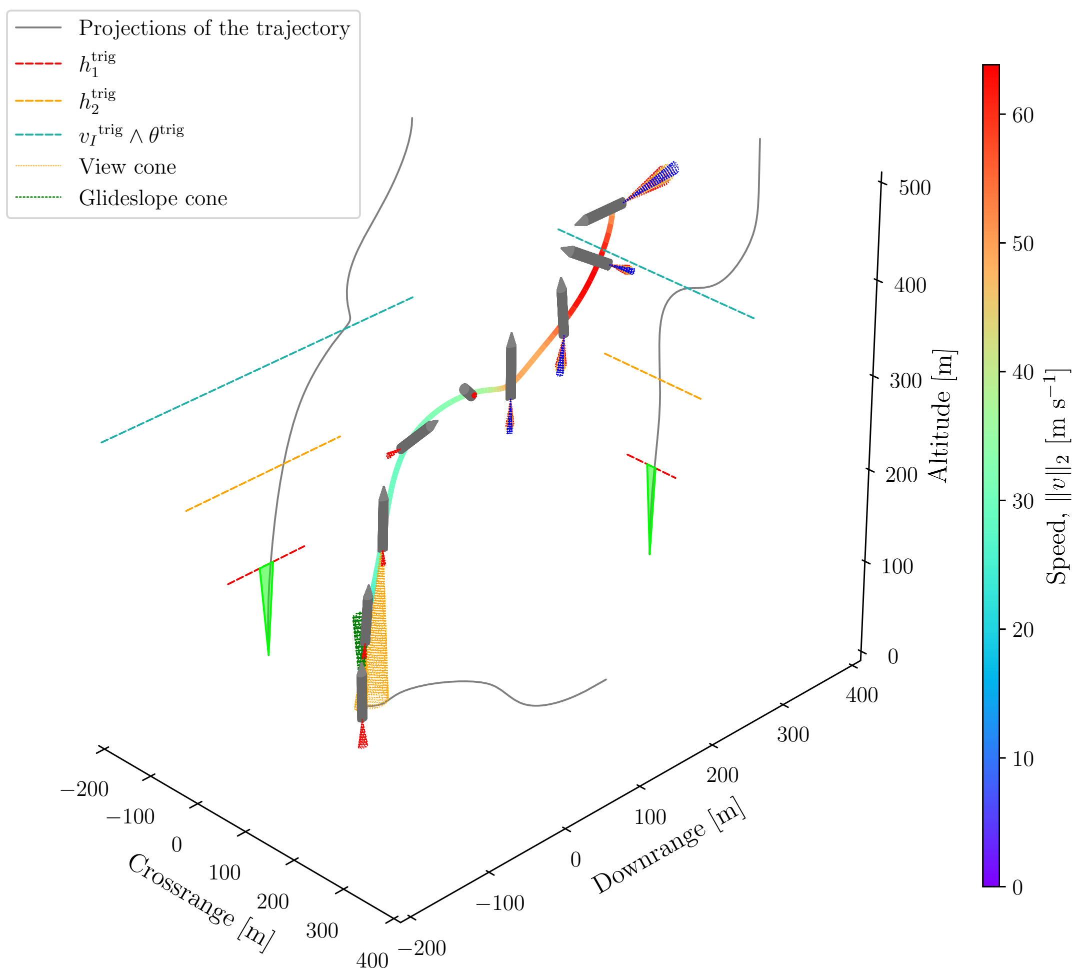


Figure 5.17: The landing trajectory of the rocket.

5.6 Quadrotor flight with CT-STL specifications

The double-integrator examples in Section 5.2 isolated the `always`, `eventually`, and `until` operators in simple settings. The present section returns to the main nonlinear example from the CT-STL study and shows how the modeling framework of Chapter 3, together with

Table 5.3: Vehicle, boundary-condition, and state-triggered-constraint parameters used in the 6-DoF rocket-landing example.

Parameter	Value
K	15
g_0	9.806 m s^{-2}
ρ	1.225 kg m^{-3}
I_{sp}	330 s
$J_{\mathcal{B}}(t)$	$m(t)\mathrm{diag}([60, 60, 1.5])$ kg m^2
C_A	$\mathrm{diag}([0.4068, 0.4068, 0.0522])$
S_A	545 m^2
$r_{\mathrm{cm},\mathcal{B}}$	$(0, 0, -14)$ m
$r_{\mathrm{cp},\mathcal{B}}$	$(0, 0, 3)$ m
m_{i}	100000 kg
m_{dry}	85000 kg
r_{i}	$(200, 200, 500)$ m
r_{f}	$(0, 0, 0)$ m
v_{i}	$(0, 0, -50)$ m s^{-1}
v_{f}	$(0, 0, -5)$ m s^{-1}
$q_{\mathcal{B}\leftarrow\mathcal{I}_{\mathrm{i}}}$	$(\sqrt{2}, \sqrt{2}, 0, 0)$
$q_{\mathcal{B}\leftarrow\mathcal{I}_{\mathrm{f}}}$	$(1, 0, 0, 0)$
$\omega_{\mathcal{B}_{\mathrm{i}}}$	$(0, 0, 0)$ deg s^{-1}
$\omega_{\mathcal{B}_{\mathrm{f}}}$	$(0, 0, 0)$ deg s^{-1}

Parameter	Value
$\omega_{\max}$	90 deg s^{-1}
$\theta_{\max}$	90 deg
$\gamma_{\max}$	35 deg
$\delta^{\mathrm{e}}_{\max}$	10 deg
$\phi^{\mathrm{e}}_{\max}$	180 deg
$\delta^{\mathrm{b}}_{\max}$	20 deg
$\phi^{\mathrm{b}}_{\max}$	180 deg
h_1^{trig}	100 m
h_2^{trig}	200 m
$v_{\mathcal{I}}^{\mathrm{trig}}$	35 m s^{-1}
θ^{trig}	60 deg
$v_{\mathcal{I}}^{\mathrm{stc}}$	20 m s^{-1}
ω^{stc}	2.5 deg s^{-1}
θ^{stc}	5 deg
γ^{stc}	5 deg
ψ^{stc}	5 deg
δ^{stc}	1 deg s^{-1}
$T_{\max}^{\mathrm{stc}_1}$	2 200 000 kg m s^{-2}
$T_{\min}^{\mathrm{stc}_1}$	880 000 kg m s^{-2}
$T_{\max}^{\mathrm{stc}_2}$	6 600 000 kg m s^{-2}
$T_{\min}^{\mathrm{stc}_2}$	2 640 000 kg m s^{-2}

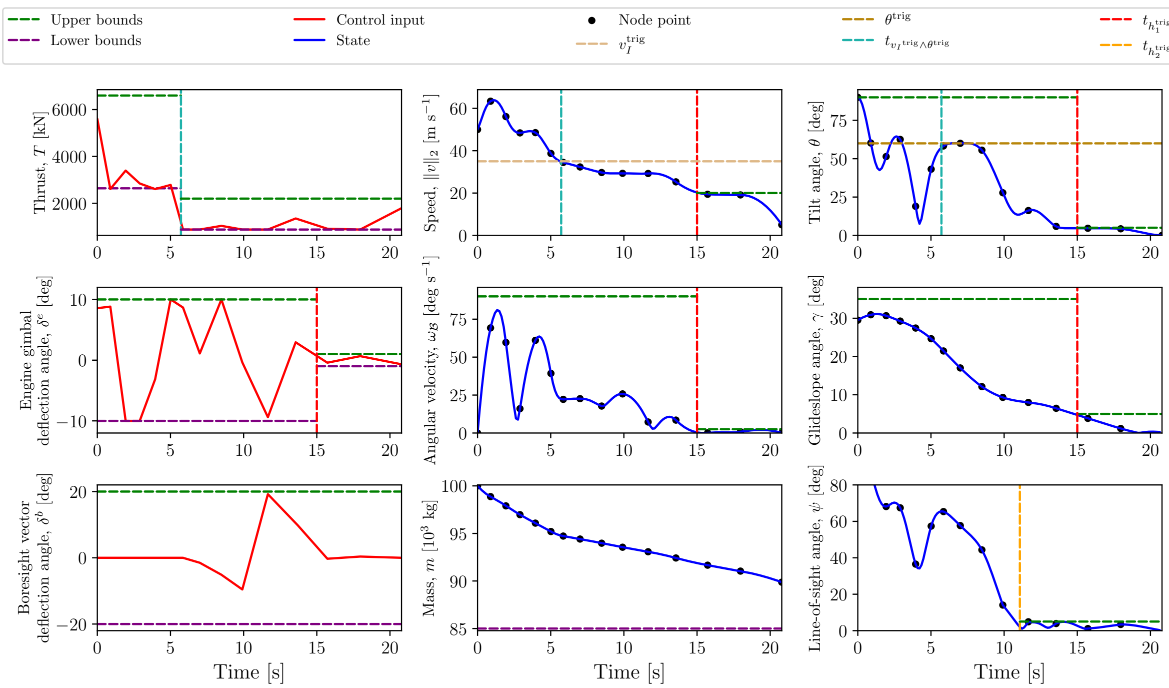


Figure 5.18: State and control inputs of the rocket during landing.

the transcription machinery of Chapter 2 and the SCP framework of Chapter 4, can handle a free-final-time 6-DoF quadrotor problem with moving waypoints, moving implication-type obstacle constraints, and terminal-set requirements. Unlike the earlier illustrative problems, this example combines nonlinear rigid-body dynamics, continuously moving predicates, temporal waypoint visitation requirements, and continuous-time safety constraints within a single trajectory optimization problem.

As in Section 5.2, the objective is specification satisfaction rather than faithful numerical reproduction of the full GMSR value. Accordingly, we use the notation $\widehat{\Gamma}$ for the terminal surrogate quantities appearing below.

5.6.1 Free-final-time 6-DoF quadrotor example

For the main example, the physical quadrotor state and control are

$$x_q := (r, v, \phi, \theta, \psi, p_b, q_b, r_b) \in \mathbb{R}^{12}, \qquad u_q := (\tau_\phi, \tau_\theta, \tau_\psi, T) \in \mathbb{R}^4,$$

where $r = (r_x, r_y, r_z)$ and $v = (v_x, v_y, v_z)$ denote position and velocity, $\Omega := (\phi, \theta, \psi)$ are the Euler angles, $\omega_b := (p_b, q_b, r_b)$ are the body angular rates, T is the thrust magnitude, and $(\tau_\phi, \tau_\theta, \tau_\psi)$ are the body torques. The dynamics are modeled by the standard Newton–Euler equations for a quadrotor, as in [175], and are written compactly as

$$\dot{x}_q(t) = f_q(x_q(t), u_q(t)).$$

The quadrotor is subject to the continuous-time bounds

$$0.06\, g_0 \le T(t) \le 1.75\, g_0, \qquad |\tau_\phi(t)|,\ |\tau_\theta(t)|,\ |\tau_\psi(t)| \le 0.25,$$

together with

$$\|v(t)\|_2 \le 20, \quad -5 \le r_z(t) \le 10, \quad |\phi(t)|,\ |\theta(t)| \le 45^\circ, \quad |p_b(t)|,\ |q_b(t)|,\ |r_b(t)| \le 100^\circ/\mathrm{s}.$$

In the implementation, these continuous-time state and control bounds are handled through an additional accumulated path-penalty state η_{q}, analogous to the path-constraint treatment used in Section 5.2. Concretely, if

$$g_{q,\ell}(x_q, u_q) \le 0, \qquad \ell = 1, \dots, n_{q,\text{path}},$$

collect the scalar inequalities associated with the thrust, torque, speed, altitude, attitude, and body-rate bounds, then the accumulated path-penalty state evolves according to

$$\dot{\eta}_{\text{q}}(t) = \chi_{\text{path},q}(x_q(t), u_q(t)) := \sum_{\ell=1}^{n_{q,\text{path}}} [g_{q,\ell}(x_q(t), u_q(t))]_+^2, \qquad \eta_{\text{q}}(0) = 0.$$

By construction, $\eta_{\text{q}}(t_{\text{f}}) = 0$ is equivalent to continuous-time satisfaction of the encoded bound constraints.

Define the compact boundary-condition sets

$$\mathcal{X}_{q,\text{i}} := \{x_q : r = (0, -10, 1),\ v = 0,\ \Omega = 0,\ \omega_b = 0\},$$
$$\mathcal{X}_{q,\text{f}} := \{x_q : 8 \le r_x \le 10,\ v = 0,\ \phi = \theta = 0,\ \omega_b = 0\}.$$

The problem uses $K = 32$ shooting nodes and an initial final-time guess of 10 s. The final time is free and is optimized through the time-dilation variables in the multiple-shooting transcription.

To represent the moving predicates, the implementation augments the state with an internal clock state ϑ, the path-penalty state η_{q}, six obstacle-violation states, and two eventually-type waypoint pairs:

$$x^{\varphi_{\text{q}}} := \Big(x_q, \vartheta, \eta_{\text{q}}, \xi_1^\ell, \xi_1^u, \xi_2^\ell, \xi_2^u, \xi_3^\ell, \xi_3^u, y_1, z_1, y_2, z_2\Big).$$

The internal clock is governed by

$$\dot{\vartheta}(t) = 1, \qquad \vartheta(0) = 0,$$

and is used to parameterize the moving waypoints and moving obstacle gaps.

The two moving waypoint centers are

$$c_1(\vartheta) := \Big(-5,\ -16 + 11.5\Big(\sin(\pi\vartheta/5.655 - \pi/2) + 1\Big),\ 1\Big),$$
$$c_2(\vartheta) := \Big(-10,\ 7 - 11.5\Big(\sin(\pi\vartheta/4.38 - \pi/2) + 1\Big),\ 1\Big),$$

both with radius $r_w = 0.5$. For each waypoint $i = 1, 2$, define

$$\rho_i(r, \vartheta) := r_w^2 - \|r - c_i(\vartheta)\|_2^2,$$

and the predicates

$$\mu_{w,i}(r, \vartheta) := (\rho_i(r, \vartheta) \geq 0).$$

The moving-waypoint task is therefore

$$\varphi_{\mathrm{wp}} := \boldsymbol{F}_{[0,t_{\mathrm{f}}]}\mu_{w,1} \wedge \boldsymbol{F}_{[0,t_{\mathrm{f}}]}\mu_{w,2}.$$

As in the double-integrator `eventually` example of Section 5.2, each waypoint is encoded by a geometric-integral state y_i and an additive state z_i:

$$\begin{aligned}
\dot{y}_i(t) &= \frac{y_i(t)}{t_{\mathrm{f}}} \log\Big(\varepsilon_{\mathrm{e}} + [\rho_i(r(t), \vartheta(t))]_-^2\Big), \\
\dot{z}_i(t) &= \frac{1}{t_{\mathrm{f}}}[\rho_i(r(t), \vartheta(t))]_+^2,
\end{aligned}$$

with

$$y_i(0) = 1, \qquad z_i(0) = 0, \qquad i = 1, 2.$$

The associated waypoint surrogates are

$$\widehat{\Gamma}_{w,i}^q := \sqrt{c_q^2 + z_i(t_{\mathrm{f}})} - \sqrt{c_q^2 + y_i(t_{\mathrm{f}})}, \qquad i = 1, 2.$$

The moving obstacle field is represented by three x-bands,

$$[a_1, b_1] = [0.5, 1.5], \qquad [a_2, b_2] = [3.5, 4.5], \qquad [a_3, b_3] = [6.5, 7.5],$$

with time-varying gap boundaries

$$\begin{aligned}
&\Delta(\vartheta) := 3(\sin(\vartheta - \pi/2) + 1), \\
&\ell_1(\vartheta) = -1 - \Delta(\vartheta), \qquad u_1(\vartheta) = 1 - \Delta(\vartheta), \\
&\ell_2(\vartheta) = -10 - \Delta(\vartheta), \qquad u_2(\vartheta) = -8 - \Delta(\vartheta), \\
&\ell_3(\vartheta) = -1 + \Delta(\vartheta), \qquad u_3(\vartheta) = 1 + \Delta(\vartheta).
\end{aligned}$$

The corresponding implication-type obstacle predicates are

$$\varphi^{\ell}_{\mathrm{obs},j}(r,\vartheta) := \Big(a_j \le r_x \le b_j \Rightarrow r_y \ge \ell_j(\vartheta)\Big),$$
$$\varphi^{u}_{\mathrm{obs},j}(r,\vartheta) := \Big(a_j \le r_x \le b_j \Rightarrow r_y \le u_j(\vartheta)\Big), \qquad j = 1,2,3.$$

Hence,

$$\varphi_{\mathrm{obs}} := \boldsymbol{G}_{[0,t_{\mathrm{f}}]}\left(\bigwedge_{j=1}^{3} \varphi^{\ell}_{\mathrm{obs},j} \wedge \varphi^{u}_{\mathrm{obs},j}\right).$$

In the implementation, these six moving implication constraints are encoded by the auxiliary states

$$\dot{\xi}^{\ell}_j(t) = \chi^{\ell}_j(r(t),\vartheta(t)), \qquad \xi^{\ell}_j(0) = 0,$$
$$\dot{\xi}^{u}_j(t) = \chi^{u}_j(r(t),\vartheta(t)), \qquad \xi^{u}_j(0) = 0,$$

for $j = 1,2,3$, where

$$\chi^{\ell}_j(r,\vartheta) := [a_j - r_x]^2_-\,[r_x - b_j]^2_-\,[r_y - \ell_j(\vartheta)]^2_-,$$
$$\chi^{u}_j(r,\vartheta) := [a_j - r_x]^2_-\,[r_x - b_j]^2_-\,[u_j(\vartheta) - r_y]^2_-.$$

Each function χ^{ℓ}_j or χ^{u}_j vanishes when the corresponding moving implication constraint is satisfied, and becomes positive only under violation. For compactness, define the total accumulated moving-obstacle violation as

$$\Xi_{\mathrm{obs}}(t) := \sum_{j=1}^{3}\Big(\xi^{\ell}_j(t) + \xi^{u}_j(t)\Big).$$

We then define the corresponding obstacle-avoidance surrogate

$$\widehat{\Gamma}_{\mathrm{obs}} := -\Xi_{\mathrm{obs}}(t_{\mathrm{f}}).$$

Combining the temporal requirements, the main logical task for the quadrotor is

$$\varphi_{\mathrm{q}} := \varphi_{\mathrm{obs}} \wedge \varphi_{\mathrm{wp}},$$

together with the terminal end-zone constraint encoded in $\mathcal{X}_{q,\mathrm{f}}$.

Finally, define the overall terminal surrogate

$$\widehat{\Gamma}_{\mathrm{q}} := \widehat{\Gamma}^{q}_{w,1} + \widehat{\Gamma}^{q}_{w,2} + w_{\Xi}\widehat{\Gamma}_{\mathrm{obs}} - w_{\eta}\eta_{\mathrm{q}}(t_{\mathrm{f}}) - w_t t_{\mathrm{f}}.$$

The resulting free-final-time problem is written as

$$\boxed{\begin{aligned}
\underset{x^{\varphi_{\mathrm{q}}}, u_q, t_{\mathrm{f}}}{\text{maximize}}\quad & \widehat{\Gamma}_{\mathrm{q}} \\
\text{subject to}\quad & \dot{x}_q(t) = f_q(x_q(t), u_q(t)), \\
& \dot{\vartheta}(t) = 1, \\
& \dot{\eta}_{\mathrm{q}}(t) = \chi_{\mathrm{path},q}(x_q(t), u_q(t)), \\
& \dot{y}_i(t) = \frac{y_i(t)}{t_{\mathrm{f}}}\,\log\!\Big(\varepsilon_{\mathrm{e}} + [\rho_i(r(t), \vartheta(t))]^2_-\Big), \quad \forall i, \\
& \dot{z}_i(t) = \frac{1}{t_{\mathrm{f}}}[\rho_i(r(t), \vartheta(t))]^2_+, \quad \forall i, \\
& \dot{\xi}^{\ell}_j(t) = \chi^{\ell}_j(r(t), \vartheta(t)), \quad \forall j, \\
& \dot{\xi}^{u}_j(t) = \chi^{u}_j(r(t), \vartheta(t)), \quad \forall j, \\
& x_q(0) \in \mathcal{X}_{q,\mathrm{i}}, \quad x_q(t_{\mathrm{f}}) \in \mathcal{X}_{q,\mathrm{f}}, \\
& \vartheta(0) = 0, \quad \eta_{\mathrm{q}}(0) = 0, \\
& y_i(0) = 1, \quad z_i(0) = 0, \quad \forall i, \\
& \xi^{\ell}_j(0) = 0, \quad \xi^{u}_j(0) = 0, \quad \forall j.
\end{aligned}}$$

This free-final-time 6-DoF quadrotor problem is the main nonlinear demonstration of the CT-STL framework. Here, the vehicle must coordinate its motion with both the moving waypoint regions and the moving obstacle gaps while simultaneously satisfying nonlinear rigid-body dynamics, terminal-set constraints, and state/control bounds. Since both the waypoint centers and the obstacle boundaries vary continuously in time, satisfaction only at the shooting nodes would be insufficient, and continuous-time evaluation is therefore essential in this example.

Figure 5.19 shows that the optimized 3D trajectory first intercepts the second moving waypoint region at approximately $t = 2.26$ s, then the first moving waypoint region at approximately $t = 3.11$ s, and subsequently passes through the three moving doorway gaps at approximately $t = 3.79$ s, $t = 4.98$ s, and $t = 6.38$ s before reaching the terminal end zone. This ordering is not imposed a priori; rather, it emerges from the optimizer's need to synchronize the trajectory with the moving waypoint and obstacle geometry. In the reported solution, the flight-time variable is initialized at 10 s and converges to 7.325 s, showing that the optimizer adapts the overall traversal time in order to coordinate the maneuver with the oscillatory target and obstacle dynamics. The final continuous-time trajectory satisfies both moving-waypoint visitation requirements and avoids all six moving obstacle constraints throughout the horizon.

Figure 5.20 confirms that the optimized trajectory remains within the prescribed thrust, torque, speed, altitude, attitude, and body-rate bounds over the continuous-time trajectory representation.

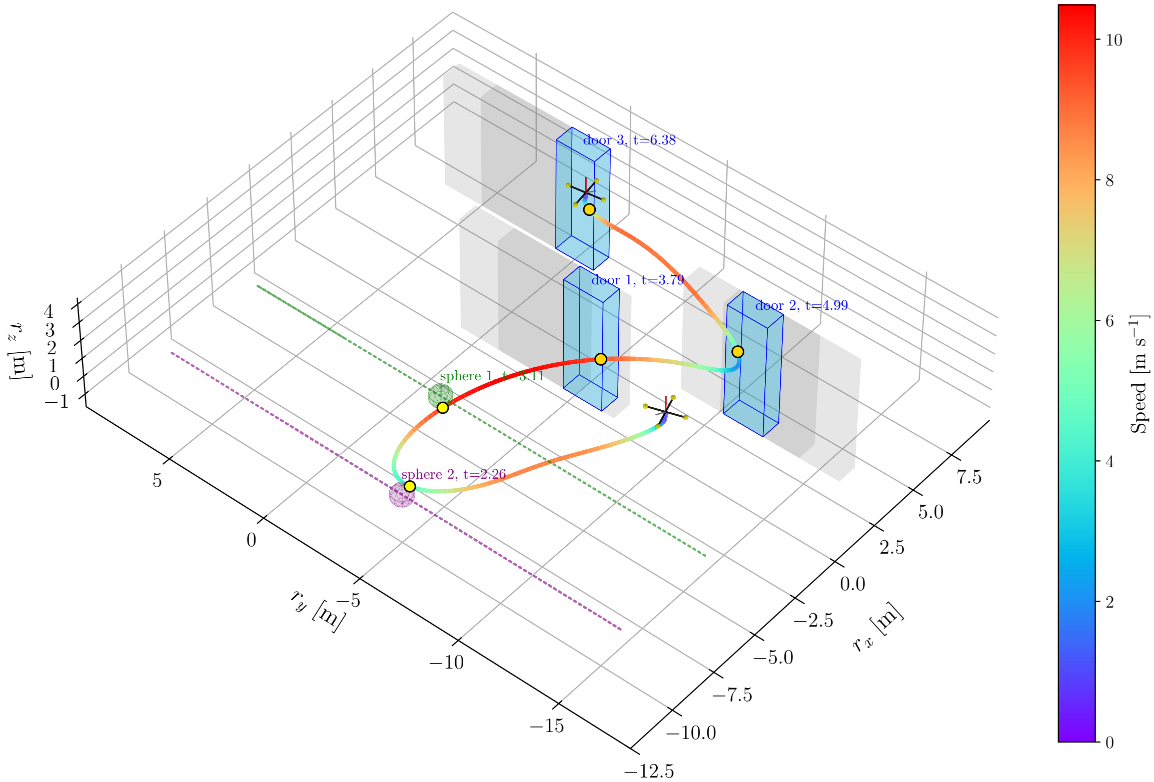


Figure 5.19: Optimized 3D trajectory for the free-final-time 6-DoF quadrotor example. The quadrotor first intercepts the two moving waypoint regions and then passes through the three moving doorway gaps while avoiding the shaded obstacle regions before reaching the terminal end zone. The annotated snapshots indicate the corresponding event times, and the trajectory is color-coded by speed.

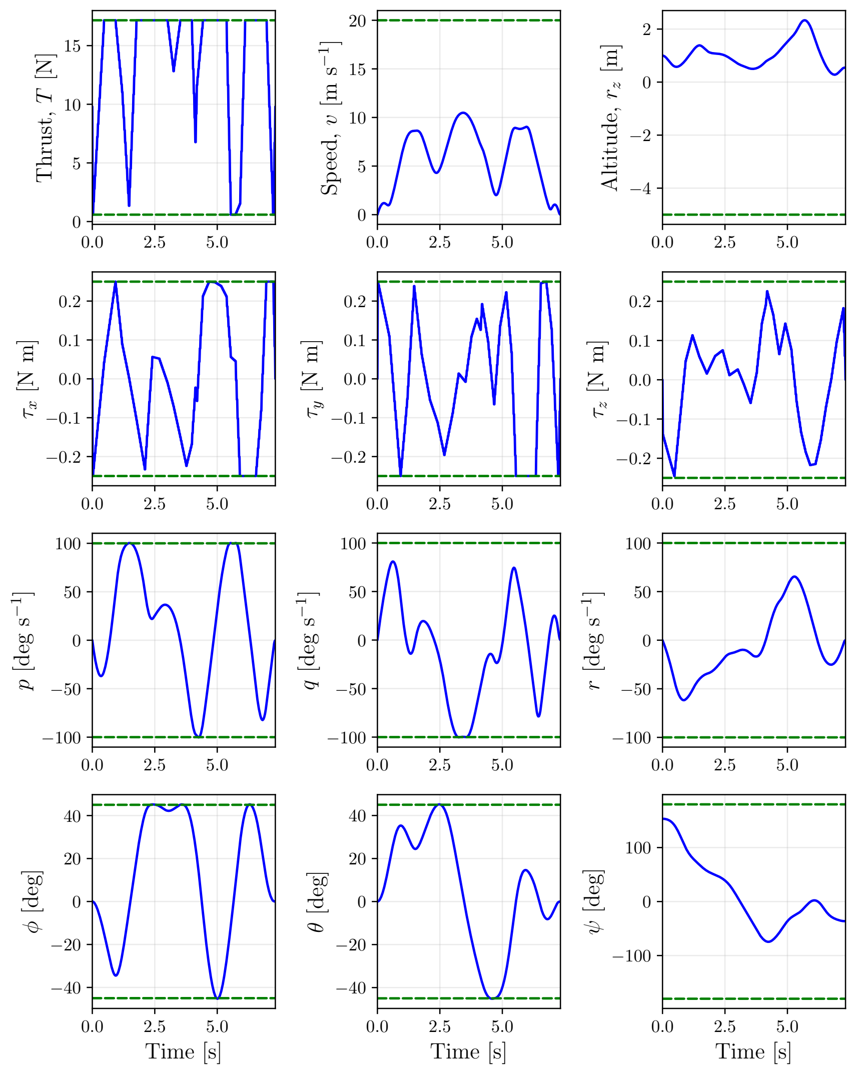


Figure 5.20: Representative state and input histories for the free-final-time 6-DoF quadrotor example, showing that the optimized continuous-time trajectory remains within the prescribed thrust, torque, speed, altitude, attitude, and body-rate bounds.

Chapter 6

DECENTRALIZED STATE-DEPENDENT MARKOV-CHAIN SYNTHESIS WITH AN APPLICATION TO SWARM GUIDANCE

6.1 Introduction

Whereas the previous chapters focused on trajectory optimization and mission-level specifications for individual systems, this chapter turns to large-scale distributed guidance through Markov-chain synthesis [288]. The common theme remains the design of structured algorithms with rigorous convergence guarantees, but the setting now shifts from continuous-state trajectory generation to finite-state population dynamics and swarm distributions.

Markov chain synthesis has garnered attention from various disciplines, including physics, systems theory, computer science, and numerous other fields of science and engineering. This attention is particularly notable within the context of Monte Carlo Markov Chain (MCMC) algorithms [113, 121, 166]. The fundamental idea underlying MCMC algorithms is to synthesize a Markov chain that converges to a specified steady-state distribution. Random sampling of a large state space while adhering to a predefined probability distribution is the predominant use of MCMC algorithms. The current literature covers a broad spectrum of methodologies for Markov chain synthesis, incorporating both heuristic approaches and optimization-based techniques [44, 46, 132]. Each method provides specialized algorithms tailored to the synthesis of Markov chains in alignment with specific objectives or constraints. Markov chain synthesis plays a central role in probabilistic swarm guidance, which has led to the development of various algorithms incorporating additional transition and safety constraints [3, 4, 8, 21, 76–78, 84, 85, 100, 141]. The probabilistic swarm guidance algorithm models the spatial distribution of the swarm agents as a probability distribution and employs the synthesized Markov chain to guide the spatial distribution of the swarm.

Consensus protocols form an important field of research that has a strong connection with Markov chains [201]. Consensus protocols are a set of rules used in distributed systems to achieve agreement among a group of agents on the value of a variable [57, 152, 221, 230]. Markov chains are often employed to model and analyze the dynamics and convergence properties of consensus protocols, providing insights into their behavior and performance [45, 222, 275, 301].

This chapter first presents a consensus protocol specifically tailored to operate on a dynamic graph topology. We establish that the protocol provides exponential convergence guarantees under mild technical conditions, thereby eliminating the reliance on the conventional connectivity assumptions commonly found in the existing literature [112, 140, 206, 220, 236, 262]. Building on this consensus protocol, the chapter then develops a decentralized state-dependent Markov chain (DSMC) synthesis algorithm. We further show that the synthesized Markov chain satisfies the same mild conditions, which in turn ensures exponential convergence to the desired steady-state distribution. Finally, the DSMC algorithm is demonstrated on the probabilistic swarm guidance problem, where the objective is to control the spatial distribution of swarm agents. Simulation results show that the DSMC algorithm achieves considerably faster convergence than existing Markov chain synthesis algorithms.

The remainder of this introduction places these developments in context. We first review the relevant literature on Markov chain synthesis, swarm guidance, and consensus protocols, then summarize the main contributions of the chapter and outline the remaining sections.

Related work. The Metropolis-Hastings algorithm is widely recognized as one of the most prominent techniques for MCMC algorithms [132]. For the fastest mixing Markov chain synthesis, the problem is formulated as a convex optimization problem in [44], assuming that the Markov chain is symmetric. This paper also presents an extension to the method that involves synthesizing the fastest mixing reversible Markov chain with a given desired distribution. Furthermore, the number of variables in the optimization problem is reduced in

[46] by exploiting the symmetries in the graph.

The probabilistic swarm guidance problem has been a crucial domain for the application and enhancement of Markov chain synthesis algorithms. A comprehensive review of the broader category of multi-agent algorithms is presented in [246], while a survey specifically focusing on aerial swarm robotics is provided in [69]. Additionally, [258] offers an overview of existing swarm robotic applications. For swarm guidance purposes, certain deterministic algorithms have been developed in [157, 174, 234, 239, 254, 281]. However, these algorithms may become computationally infeasible when dealing with swarms that comprise hundreds to thousands of agents. In the context of addressing the guidance problem for a large number of agents, considering the spatial distribution of swarm agents and directing it towards a desired steady-state distribution offers a computationally efficient approach. In this regard, both probabilistic and deterministic swarm guidance algorithms are presented in [31, 101, 102, 107, 160, 161, 314] for continuous state spaces. For discrete state spaces, a probabilistic guidance algorithm is introduced in [3]. This algorithm treats the spatial distribution of swarm agents, called the density distribution, as a probability distribution and employs the Metropolis-Hastings (M-H) algorithm to synthesize a Markov chain that guides the density distribution toward a desired state. The probabilistic guidance algorithm led to the development of numerous Markov chain synthesis algorithms involving specific objectives and constraints [4, 8, 21, 76–78, 84, 85, 100, 141]. In [100], the Metropolis-Hastings algorithm is extended to incorporate safety upper bound constraints on the probability vector. This paper includes numerical simulations that demonstrate the application of the extension in a probabilistic swarm guidance problem. In order to enhance convergence rates, [4] introduces a convex optimization-based technique for Markov chain synthesis. This technique formulates the objective function and constraints using linear matrix inequalities (LMIs) to synthesize a Markov chain capable of achieving a desired distribution while adhering to specified transition constraints. Notably, this study does not impose any assumptions on the Markov chain, rendering the problem inherently non-convex. The problem is convexified for practical purposes but the optimal convergence rate of the original non-convex problem cannot be attained. In [76], the approach presented in [4] is enhanced by incorporating state-feedback to further improve the convergence rate. These works are also extended to impose density upper bounds and density rate constraints in [8] and density flow and density diffusivity constraints in [77]. In [78], a more general approach is proposed for cases where the environment is stochastic, and the resulting Markov chain is shown to be a linear function of the stochastic environment and decision policy. These convex optimization problems can be solved using the interior-point algorithm but the computation time of the algorithm increases rapidly with increasing dimensionality of the state space [43]. Furthermore, neither the M-H algorithm nor convex optimization-based approaches attempt to minimize the number of state transitions. The swarm guidance problem is formulated as an optimal transport problem in [20] to minimize the number of agent transitions. However, besides the similar computational complexity issues, the performance of the algorithm drops significantly if the current density distribution of the swarm cannot be estimated accurately. The time-inhomogeneous Markov chain approach to the probabilistic swarm guidance problem (PSG-IMC algorithm) is developed in [21] to minimize the number of state transitions. This algorithm is computationally efficient and yields reasonable results with low estimation errors. However, all feedback-based algorithms mentioned above require global feedback on the state of the density distribution. Communication between all agents has to be established to estimate the density distribution in the probabilistic swarm guidance problem. Generating perfect feedback of the density distribution is often impractical, leading to the routine occurrence of estimation errors. An alternative approach that requires only local information is developed in [141]. In this work, the time-inhomogeneous Markov chain approach presented in [21] is modified to work with local information, and the algorithm is used for minimizing the number of state transitions. Nevertheless, in both global and local information based time-inhomogeneous Markov chain approaches, it is observed that the convergence rate diminishes significantly when the state transition capabilities become more restricted. As it can be seen in Corollary 1 in [21] or Corollary 3 in [141], the transition probability from a state to any directly connected state cannot be higher than the desired density value of the corresponding directly connected state. In situations where there are sparsely connected regions in the state space, it is common to observe a relatively low sum of desired density values among directly connected states. Consequently, there is a higher probability of remaining in the same state rather than transitioning to other states. In terms of the convergence rate, these algorithms are only effective in cases with high transition capabilities. Additionally, the performance of these algorithms is highly sensitive to hyperparameters and requires careful selection for optimum results in each experiment.

Graph temporal logic (GTL) is introduced in [84] to impose high-level task specifications as a constraint to the Markov chain synthesis. Markov chain synthesis is formulated as mixed-integer nonlinear programming (MINLP) feasibility problem and the problem is solved using a coordinate descent algorithm. In addition, an equivalence is proven between the feasibility of the MINLP and the feasibility of a mixed-integer linear program (MILP) for a particular case where the agents move along the nodes of a complete graph. While this study assumes homogeneous swarms for Markov chain synthesis subject to finite-horizon GTL formulas, an improved version of the formulation is presented in [85] to enable probabilistic control of heterogeneous swarms subject to infinite-horizon GTL formulas. Instead of solving the resulting MINLP using a coordinate descent algorithm, a sequential scheme, which is faster, more accurate, and robust to the choice of the starting point, is developed in the aforementioned paper.

Markov chains and consensus protocols share a close relationship. The rich theory of Markov chains has proven to be valuable in analyzing specific consensus protocols. Notable works such as [45, 222, 275, 301] have leveraged Markov chain theory to provide insights and analysis for consensus protocols. Consensus protocols, in contrast to Markov chains, operate without the limitations of non-negative nodes and edges or the requirement for the sum of nodes to equal one [201]. This broader scope enables consensus protocols to address a significantly wider range of problem spaces. Therefore, there is a significant interest in consensus protocols in a broad range of multi-agent networked systems research, including distributed coordination of mobile autonomous agents [112, 140, 206, 218, 220, 236], distributed optimization [47, 151, 211, 259, 305], distributed state estimation [142, 219], or dynamic load-balancing for parallel processors [74, 302]. There are comprehensive survey papers that review the research on consensus protocols [57, 152, 221, 230]. In many scenarios, the network topology of the consensus protocol is a switching topology due to failures, formation reconfiguration, or state-dependence. There is a large number of papers that propose consensus protocols with switching network topologies and convergence proofs of these algorithms are provided under various assumptions [112, 140, 206, 220, 236, 262]. In [140], a consensus protocol is proposed to solve the alignment problem of mobile agents, where the switching topology is assumed to be periodically connected. This assumption means that the union of networks over a finite time interval is strongly connected. Another algorithm is proposed in [206] that assumes the underlying switching network topology is ultimately connected. This assumption means that the union of graphs over an infinite interval is strongly connected. In [236], previous works are extended to solve the consensus problem on networks under limited and unreliable information exchange with dynamically changing interaction topologies. The convergence of the algorithm is provided under the ultimately connected assumption. Another consensus protocol is introduced in [112] for the cooperation of vehicles performing a shared task using inter-vehicle communication. Based on this work, a theoretical framework is presented in [220] to solve consensus problems under a variety of assumptions on the network topology such as strongly connected switching topology. In [262], a consensus protocol with state-dependent weights is proposed and it is assumed that corresponding graphs are weakly connected, which means that the network is assumed to be connected over the iterations. Additionally, optimization-based algorithms are proposed to obtain a high convergence rate for the consensus protocol in [301] and [156].

Main contributions. In this chapter, we first propose a consensus protocol with state-dependent weights. The proposed consensus protocol does not require any connectivity assumption on the dynamic network topology, unlike the existing methods in the literature [112, 140, 206, 220, 236, 262]. We provide an exponential convergence guarantee for the consensus protocol under some mild technical conditions, which can be verified straightforwardly. We then present a decentralized Markov-chain synthesis (DSMC) algorithm based on the proposed consensus protocol and we prove that the resulting DSMC algorithm satisfies these mild conditions. This result is employed to prove that the resulting Markov chain has a desired steady-state distribution and that all initial distributions exponentially converge to this steady-state. Unlike the homogeneous Markov chain synthesis algorithms in [3, 4, 44, 46, 100, 132], the Markov matrix, synthesized by our algorithm, approaches the identity matrix as the probability distribution converges to the desired steady-state distribution. Hence the proposed algorithm attempts to minimize the number of state transitions, which eventually converge to zero as the probability distribution converges to the desired steady-state distribution. Whereas previous time-inhomogeneous Markov chain synthesis algorithms in [21, 141] only provide asymptotic convergence, the DSMC algorithm provides an exponential convergence rate guarantee. Furthermore, unlike previous algorithms in [21, 141], the convergence rate of the DSMC algorithm does not rapidly decrease in scenarios where the state space contains sparsely connected regions. Due to the decentralized nature of the consensus protocol, the Markov chain synthesis relies on local information, similar to the approach presented in [141], and a complex communication architecture is not required for the estimation of the distribution. By presenting numerical evidence within the context of the probabilistic swarm guidance problem, we demonstrate that the convergence rate of the swarm distribution to the desired steady-state distribution is substantially faster when compared to previous methodologies. In summary:

- We propose a consensus protocol with state-dependent weights and prove its exponential convergence under mild technical conditions, without relying on the typical connectivity assumptions associated with the underlying graph topology.
- Based on the proposed consensus protocol, we introduce the DSMC algorithm, which is shown to meet the convergence conditions outlined by the consensus protocol.
- Simulation results demonstrate that the DSMC algorithm achieves faster convergence,

characterized by an exponential convergence guarantee, compared to existing homogeneous and time-inhomogeneous Markov chain synthesis algorithms presented in [3] and [21].

Organization and Notation. The chapter is organized as follows. Section 6.2 presents the consensus protocol with state-dependent weights. Section 6.3 introduces the decentralized state-dependent Markov-chain synthesis (DSMC) algorithm and establishes its main properties and convergence guarantees. Section 6.4 introduces the probabilistic swarm guidance problem formulation and presents numerical simulations of swarms converging to desired distributions.

Notation: $\mathbb{R}$ and $\mathbb{C}$ represent the set of real numbers and complex numbers, respectively. $\mathbb{R}_+$ and $\mathbb{Z}_+$ represent the set of non-negative real numbers and non-negative integers, respectively. $\mathbb{R}^n$ is the n dimensional real vector space. $\mathbf{0}$ is a zero matrix and $\mathbf{1}$ is a matrix of ones in appropriate dimensions. $x[i](k)$ represents the i^{th} element of the vector $x \in \mathbb{R}^n$ at time k. $\|x\|_p$ denotes the ℓ_p vector norm. $(v_1, v_2, ..., v_n)$ represents a vector, such that $(v_1, v_2, ..., v_n) \equiv [v_1^T, v_2^T, ..., v_n^T]^T$ where v_i have arbitrary dimensions. $M > (\geq)\ \mathbf{0}$ implies that $M \in \mathbb{R}^{m\times n}$ is a positive (non-negative) matrix. $\mathcal{I}_1^m$ denotes the set of integers such that $\mathcal{I}_1^m = \{1, 2, \ldots, m\}$. $\lambda_i(A)$ is the i^{th} eigenvalue of a matrix $A \in \mathbb{R}^{m\times m}$ such that $\lambda_i(A) \leq \lambda_{i+1}(A)$ for $i \in \mathcal{I}_1^{m-1}$. $\sigma(A)$ is the set of eigenvalues of $A \in \mathbb{R}^{m\times m}$ such that $\sigma(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$. $\emptyset$ denotes the empty set. $V \setminus W$ represents the elements in set V that are not in set W. $\mathcal{P}$ denotes the probability of a random variable.

6.2 A Consensus Protocol with State-Dependent Weights

The consensus protocol is a process used for achieving an agreement among distributed agents. In this section, we introduce a consensus protocol with state-dependent weights to reach a consensus on time-varying weighted graphs. Unlike other proposed consensus protocols in the literature, the consensus protocol we introduce does not require any connectivity assumption on the dynamic network topology. We provide a theoretical analysis for the proof of exponential convergence under some mild technical conditions. We then use the proposed consensus protocol for the synthesis of the column stochastic Markov matrix in Section 6.3. It is proven that these mild assumptions required by the proposed consensus protocol are satisfied by the Markov matrix synthesis algorithm. We first define the graph for our consensus approach.

Definition 6.1. *(Time-varying weighted graph) A time-varying weighted graph is a tuple, $\mathcal{G}_w(k) = (\mathcal{V}, \mathcal{E}, w(k))$, where $\mathcal{V} = \{\mathrm{v}_1, ..., \mathrm{v}_m\}$ is the set of vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of undirected edges, and $w(k)$ is a time-varying function such that $w : \mathcal{E} \to \mathbb{R}_+$, where $w_{i,j}(k)$ represents the value of weight for edge $\{\mathrm{v}_i, \mathrm{v}_j\} \in \mathcal{E}$ at time k.*

- *(Values of vertices) A time-varying vector $e(k) \in \mathbb{R}^m$ such that $\mathbf{1}^T e(k) = 0$ for $k \in \mathbb{Z}_+$ can define a vector of values of vertices, i.e., $e[i](k)$ represents the value of the vertex v_i at time k.*
- *(Uniform graph induced by time-varying weighted graph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the uniform graph induced by $\mathcal{G}_w(k)$ and it is obtained by setting $w_{i,j}(k) = 1$, where $\{\mathrm{v}_i, \mathrm{v}_j\} \in \mathcal{E}$ for all $k \in \mathbb{Z}_+$.*
- *(Adjacency matrices) Two vertices v_i and v_j are called adjacent if $\{\mathrm{v}_i, \mathrm{v}_j\} \in \mathcal{E}$. The adjacency matrix of the uniform graph $\mathcal{G}$ is represented by $A_a(\mathcal{G})$, where $A_a(\mathcal{G})[i,j] = 1$, if v_i and v_j are adjacent, and is 0 otherwise. The adjacency matrix of the time-varying weighted graph $\mathcal{G}_w(k)$ is represented by $A_a(\mathcal{G}_w(k))$, where $A_a(\mathcal{G}_w(k))[i,j] = A_a(\mathcal{G})[i,j]w_{i,j}(k)$.*
- *(Degree matrices) $\mathcal{D}(\mathcal{G})$ and $\mathcal{D}(\mathcal{G}_w(k))$ are the diagonal degree matrices of the graphs $\mathcal{G}$ and $\mathcal{G}_w(k)$ such that $\mathcal{D}(\mathcal{G})[i,i] = \sum_j A_a(\mathcal{G})[i,j]$ and $\mathcal{D}(\mathcal{G}_w(k))[i,i] = \sum_{j \in \mathcal{I}_1^m} A_a(\mathcal{G}_w(k))[i,j]$.*
- *(Laplacian matrices) $\mathcal{L}(\mathcal{G})$ and $\mathcal{L}(\mathcal{G}_w(k))$ are the Laplacian matrices of the graphs $\mathcal{G}$ and $\mathcal{G}_w(k)$ such that $\mathcal{L}(\mathcal{G}) = \mathcal{D}(\mathcal{G}) - A_a(\mathcal{G})$ and $\mathcal{L}(\mathcal{G}_w(k)) = \mathcal{D}(\mathcal{G}_w(k)) - A_a(\mathcal{G}_w(k))$.*
- *(Activated and Deactivated Edges) If $w_{i,j}(k) = 0$, where $\{\mathrm{v}_i, \mathrm{v}_j\} \in \mathcal{E}$, the edge $\{\mathrm{v}_i, \mathrm{v}_j\}$ is deactivated. $\mathcal{E}_A(k)$ and $\mathcal{E}_D(k)$ represent the activated and deactivated edges for time k such that $\mathcal{E} = \mathcal{E}_A(k) \cup \mathcal{E}_D(k)$.*

For convenience, we use A_a, $A_{a_w}(k)$, $\mathcal{D}$, $\mathcal{D}_w(k)$, $\mathcal{L}$ and $\mathcal{L}_w(k)$ instead of using $A_a(\mathcal{G})$,

$A_a(\mathcal{G}_w(k))$, $\mathcal{D}(\mathcal{G})$, $\mathcal{D}(\mathcal{G}_w(k))$, $\mathcal{L}(\mathcal{G})$ and $\mathcal{L}(\mathcal{G}_w(k))$, respectively.

The following assumption on the topology of the uniform graph $\mathcal{G}$ is needed for convergence analysis.

Assumption 6.2. *The uniform graph $\mathcal{G}$ is a connected graph, which means there exists a path between all vertices, or equivalently, $(I + A_a)^{m-1} > \mathbf{0}$ for $A_a = A_a^T \in \mathbb{R}^{m\times m}$ [201, section 2.1].*

We will consider time-varying weighted graphs where the weights of the edges depend on the values of vertices. The following definition is needed to present the relation between the weights of the edges and the values of vertices.

Definition 6.3. *(Index sets with respect to values of vertices) The index sets $I_p(k)$ and $I_n(k)$ contain the indices of the non-negative and negative valued vertices for time k, i.e.,*

$$I_p(k) = \{i : e[i](k) \geq 0\},$$
$$I_n(k) = \{i : e[i](k) < 0\}.$$

The index set $I_{n_p}(k) \subseteq I_n(k)$ consists of the indices of the negative valued vertices that are adjacent to the non-negative valued vertices, i.e.,

$$I_{n_p}(k) = \{i : i \in I_n(k), j \in I_p(k) \text{ and } A_a[i,j] = 1\}.$$

The set that contains the edges between v_i, $i \in I_n(k)$ and v_j, $j \in I_{n_p}(k)$ is defined as

$$I_{n_p,n}(k) = \{\{i,j\} : i \in I_n(k), j \in I_{n_p}(k) \text{ and } A_a[i,j] = 1\}.$$

We provide a condition that represents a relation between the values of the vertices and the weights of the edges of the graph $\mathcal{G}_w(k)$. This condition is the key to providing the convergence proof of the consensus protocol with state-dependent weights.

Condition 6.4.

(a) If $i \in I_p(k)$, then there exists a c_1 such that $0 < c_1 \leq w_{i,j}(k)$ for all $\{\mathrm{v}_i, \mathrm{v}_j\} \in \mathcal{E}$.

(b) If $\{i, j\} \in I_{n_p,n}(k)$, then $w_{i,j}(k) = 0$.

In other words, Condition 6.4(a) implies that the weight of an edge that connects a non-negative valued vertex to any other vertex cannot be arbitrarily close to 0. Therefore, if $A_{a_w}[i,j](k) = 0$ where $e[i](k) \geq 0$, then $A_a[i,j](k) = 0$. Condition 6.4(b) implies that no transition is allowed between two negative valued vertices if at least one of them is adjacent to a non-negative valued vertex. Edges that have zero weights are called deactivated edges. Note that although $\mathcal{G}$ is assumed to be a connected graph in Assumption 6.2, $\mathcal{G}_w(k)$ may not be a connected graph for some k, due to these deactivated edges.

The following example is provided to help the reader interpret the implication of Condition 6.4.

Example 1 The values of the vertices and the weights of the edges are represented on a graph for a time k in Figure 6.1. According to Definition 6.3, $I_p(k) = \{1\}$, $I_n(k) = \{2, 3, 4, 5, 6\}$, $I_{n_p}(k) = \{2, 4\}$ and $I_{n_p,n}(k) = \{\{2, 3\}\{2, 5\}\{4, 5\}\}$. Condition 6.4(a) implies that $0 < c_1 \leq w_{1,2}(k)$ and $0 < c_1 \leq w_{1,4}(k)$ since $0 \leq e[1](k)$. Condition 6.4(b) implies that weights of all other edges of the nodes v_2 and v_4 are zero, which means that $w_{2,3}(k) = w_{2,5}(k) = w_{4,5}(k) = 0$.

6.2.1 Theoretical convergence analysis

We now turn to the convergence analysis of the proposed consensus protocol. The preceding example illustrates the structural implications of Condition 6.4. In this subsection, we formalize those implications and show how they lead to exponential convergence of the consensus error.

More specifically, we proceed in three steps. First, we show that whenever the error vector is nonzero, every non-negative valued vertex is connected, in a suitable graph-theoretic sense, to a negative valued vertex. This identifies the mixed-sign connected subgraphs along which the error can propagate. Second, for such connected weighted subgraphs, we derive a spectral bound on the associated Laplacian matrix. Third, we combine these structural and spectral

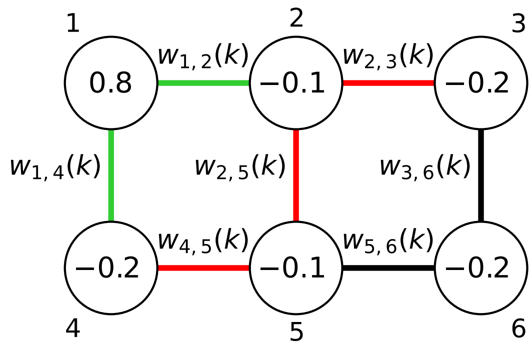


Figure 6.1: Vertices of the graph and corresponding values are represented by circles. The indices of the vertices are shown in their corners. Edges of the vertices are represented by lines, and weights of the edges are represented on the sides of the edges. Green and red lines represent the edges that are implied by Condition 6.4(a) and Condition 6.4(b), respectively. Black lines represent the remaining edges of the graph.

properties with the consensus update dynamics to prove that the error norm contracts at every iteration, and hence converges exponentially to zero.

The following lemma shows that each non-negative valued vertex is connected to a negative valued vertex in the graph $\mathcal{G}_w$ under Condition 6.4.

Lemma 6.5. *Assume that Condition 6.4 holds. If $e(k) \neq \mathbf{0}$, then for each i such that $e[i](k) \geq 0$, $\exists j$ such that $e[j](k) < 0$, where $A^n_{a_w}[i,j](k) \neq 0$ for some exponent $n \in \mathbb{Z}_+$.*

Proof. Since $\mathbf{1}^T e(k) = 0$ by Definition 6.1 and $e(k) \neq \mathbf{0}$, $\exists i, j$ such that $e[i](k) > 0$ and $e[j](k) < 0$.

The lemma is proven by contradiction. Suppose that there exists one non-negative valued vertex v_i such that $e[i](k) \geq 0$ and for all j such that $e[j](k) < 0$, $A^n_{a_w}[i,j](k) = 0$, $\forall n \in \mathbb{Z}_+$. In other words, there exists one non-negative valued vertex that is not connected to any negative valued vertices. Denote the set of such non-negative valued vertices as $\mathcal{V}^+(k)$. The set of remaining vertices is denoted as $\mathcal{V}^-(k)$ such that $\mathcal{V}^-(k) = \mathcal{V} \setminus \mathcal{V}^+(k)$. Note that all negative valued vertices and all non-negative valued vertices that are connected to negative valued vertices are in $\mathcal{V}^-(k)$. According to the assumption provided at the beginning of the proof, vertices in $\mathcal{V}^+(k)$ cannot be connected to the vertices in $\mathcal{V}^-(k)$, then the adjacency matrix of the time-varying weighted graph can be permuted as

$$\widehat{A}_{a_w}(k) = P(k)A_{a_w}(k)P(k)^T = \begin{bmatrix} A^+_{a_w}(k) & \mathbf{0} \\ \mathbf{0} & A^-_{a_w}(k) \end{bmatrix},$$

where $P(k)$ is a permutation matrix, $A^+_{a_w}(k)$ and $A^-_{a_w}(k)$ are the adjacency matrices of sub-graphs that consist of the vertices in $\mathcal{V}^+(k)$ and $\mathcal{V}^-(k)$, respectively. Note that due to Condition 6.4(a), if $A_{a_w}[i,j](k) = 0$, where $e[i](k) \geq 0$, then $A_a[i,j](k) = 0$. Therefore, A_a can also be permuted as

$$\widehat{A}_a(k) = P(k)A_aP(k)^T = \begin{bmatrix} A^+_a(k) & \mathbf{0} \\ \mathbf{0} & A^-_a(k) \end{bmatrix}. \tag{6.1}$$

Due to Assumption 6.2, $\mathcal{G}$ is a connected graph, the A_a matrix is an irreducible matrix, and then $\widehat{A}_a(k)$ matrix is also an irreducible matrix. However, the matrix in Eq. (6.1) is a reducible matrix, which means it contradicts the assumption provided at the beginning of the proof. Therefore, all non-negative valued vertices are connected to some negative valued vertices. ■

Therefore, even if $\mathcal{G}_w(k)$ may not be a connected graph for some k, there always exists at least one connected sub-graph that consists of both non-negative and negative valued vertices. It is important to note that in these connected sub-graphs, each edge is between a non-negative valued vertex and any other vertex and there is no edge between two negative valued vertices due to Condition 6.4(b). Therefore, weights of all edges are at least c_1 due to Condition 6.4(a).

The following lemma provides a bound for the eigenvalues of the Laplacian matrix $\mathcal{L}_w(k)$ of a connected time-varying weighted graph when the weights are within a certain range.

Lemma 6.6. *Let $\mathcal{G}_w(k)$ be a connected time-varying weighted graph with m vertices. Assume that the following inequality is satisfied for the weights of the edges of the graph $\mathcal{G}_w(k)$,*

$$c_1 \leq w_{i,j}(k) \leq \frac{1 - c_2}{\max(\mathcal{D}[i,i], \mathcal{D}[j,j])}, \tag{6.2}$$

where $0 < c_1$ *and* $0 < c_2 < 1$. *Then, the following inequality holds for the eigenvalues of the Laplacian matrix* $\mathcal{L}_w(k)$,

$$\sigma(\mathcal{L}_w(k)) \subseteq [c_3, (2 - c_3)] \cup \{0\},$$

where $0 < c_3 = \min\left(c_1 \frac{4}{m(m-1)}, 2c_2\right) \leq 1$.

Proof. According to Gershgorin's Circle Theorem [26], eigenvalues of a $\mathcal{L}_w(k)$ matrix satisfy, $\sigma(\mathcal{L}_w(k))$

$$\subseteq \bigcup_i \Big\{\lambda \in \mathbb{C} : \Big|\lambda - \mathcal{L}_w[i,i](k)\Big| \leq \sum_{j \in \mathcal{I}_1^m \setminus \{i\}} \Big|\mathcal{L}_w[i,j](k)\Big|\Big\}. \tag{6.3}$$

Since $\mathcal{L}_w[i,i](k) = \sum_{\{\mathrm{v}_i,\mathrm{v}_j\} \in \mathcal{E}} w_{i,j}(k)$ for weighted Laplacian matrix, where $0 < c_1 \leq w_{i,j}(k)$, Eq. (6.3) implies that, $\sigma(\mathcal{L}_w(k))$

$$\begin{aligned}
&\subseteq \bigcup_i \Big\{\lambda \in \mathbb{C} : \Big|\lambda - \sum_{\{\mathrm{v}_i,\mathrm{v}_j\} \in \mathcal{E}} w_{i,j}(k)\Big| \leq \sum_{\{\mathrm{v}_i,\mathrm{v}_j\} \in \mathcal{E}} w_{i,j}(k)\Big\} \\
&= \bigcup_i \Big\{\lambda \in \mathbb{C} : 0 \leq \lambda \leq 2 \sum_{\{\mathrm{v}_i,\mathrm{v}_j\} \in \mathcal{E}} w_{i,j}(k)\Big\}.
\end{aligned}$$

According to Eq. (6.2), $w_{i,j}(k) \leq (1 - c_2)/\mathcal{D}[i,i]$ and $w_{i,j}(k) \leq (1 - c_2)/\mathcal{D}[j,j]$. Then, eigenvalues of the Laplacian matrix $\mathcal{L}_w(k)$ satisfy that,

$$0 \leq \lambda_i(k) \leq (2 - 2c_2) \text{ for } \lambda_i(k) \in \sigma(\mathcal{L}_w(k)). \tag{6.4}$$

We now show that the second smallest eigenvalue of the Laplacian matrix $\mathcal{L}_w(k)$ is also lower bounded. In [204, Theorem 4.2], the following lower bound is provided for the second smallest eigenvalue of the Laplacian matrix of a connected unweighted graph $\mathcal{G}$,

$$\frac{4}{\mathrm{diam}(\mathcal{G})m} \leq \lambda_2(\mathcal{L}),$$

where $\mathrm{diam}(\mathcal{G})$ is the diameter of the graph, which is the length of the longest shortest path between any two vertices, and m is the number of vertices of the graph. Let us define the Laplacian matrix $\mathcal{L}_{c_1}$ for the case that weights of all edges of the graph $\mathcal{G}$ are c_1 instead of 1. Suppose that, $\lambda_i(\mathcal{L})$ and $\lambda_i(\mathcal{L}_{c_1})$ are the i^{th} eigenvalues of $\mathcal{L}$ and $\mathcal{L}_{c_1}$ matrices such that $\lambda_i(\mathcal{L}) \leq \lambda_{i+1}(\mathcal{L})$ and $\lambda_i(\mathcal{L}_{c_1}) \leq \lambda_{i+1}(\mathcal{L}_{c_1})$ for $i \in \mathcal{I}_1^{m-1}$. Then, $c_1\lambda_i(\mathcal{L}) = \lambda_i(\mathcal{L}_{c_1})$ for

$i \in \mathcal{I}_1^m$. Since $c_1 \leq w_{i,j}$ for the graph $\mathcal{G}_w(k)$, $\mathcal{L}_w(k) - \mathcal{L}_{c_1}$ is also a positive semi-definite Laplacian matrix, which means $u^T(\mathcal{L}_w(k) - \mathcal{L}_{c_1})u \geq 0$ for all $u \in \mathbb{R}^m$. Let us denote unit eigenvector $u_2 \in \mathbb{R}^m$ such that $\mathcal{L}_w(k)u_2 = \lambda_2(\mathcal{L}_w(k))u_2$ and unit eigenvector $z_i \in \mathbb{R}^m$ such that $\mathcal{L}_{c_1} z_i = \lambda_i(\mathcal{L}_{c_1})z_i$ for all $i \in \mathcal{I}_1^m$. Note that $\mathbf{1}$ is the eigenvector of these Laplacian matrices with the associated eigenvalue of 0. Since these Laplacian matrices are symmetric, all other eigenvectors of them are orthogonal to $\mathbf{1}$. Then, $u_2^T z_1 = u_2^T \mathbf{1} = 0$ and $u_2 \in \mathrm{span}(z_2, z_3, ..., z_m)$, which means $u_2 = \sum_{i \in \mathcal{I}_2^m} \alpha_i z_i$, where $\sum_{i \in \mathcal{I}_2^m} \alpha_i^2 = 1$. Thus,

$$
\begin{aligned}
0 &\leq u_2^T(\mathcal{L}_w(k) - \mathcal{L}_{c_1})u_2 \\
&= \lambda_2(\mathcal{L}_w(k)) - u_2^T \mathcal{L}_{c_1} u_2 \\
&= \lambda_2(\mathcal{L}_w(k)) - \sum_{i \in \mathcal{I}_2^m} \alpha_i z_i^T \lambda_i(\mathcal{L}_{c_1}) \alpha_i z_i \\
&= \lambda_2(\mathcal{L}_w(k)) - \sum_{i \in \mathcal{I}_2^m} \alpha_i^2 \lambda_i(\mathcal{L}_{c_1}) \\
&\leq \lambda_2(\mathcal{L}_w(k)) - \lambda_2(\mathcal{L}_{c_1}).
\end{aligned}
$$

Hence, Eq. (6.5) can be provided for the second smallest eigenvalue of $\mathcal{L}_w(k)$,

$$
c_1 \frac{4}{\mathrm{diam}(\mathcal{G})m} \leq \lambda_2(\mathcal{L}_w(k)). \tag{6.5}
$$

Since $0 < \mathrm{diam}(\mathcal{G}) \leq m - 1$ for a connected graph and due to Eq. (6.4), the following equation represents the eigenvalues of the matrix $\mathcal{L}_w(k)$,

$$
\sigma(\mathcal{L}_w(k)) \subseteq [c_3, (2 - c_3)] \cup \{0\},
$$

where $0 < c_3 = \min\left(c_1 \frac{4}{m(m-1)}, 2c_2\right) \leq \min(2c_1, 2c_2) \leq \min(2 - 2c_2, 2c_2) < 1$. ■

The following theorem shows that values of vertices of $\mathcal{G}_w(k)$ converge to 0 under specific value transition dynamics.

Theorem 6.7. *Suppose that Assumption 6.2 and Condition 6.4 are satisfied, and $e(k)$ vector in Definition 6.1 evolves according to*

$$
e[i](k+1) = e[i](k) + \sum_{j \in \mathcal{I}_1^m} \Big(e[j](k) - e[i](k)\Big) A_{a_w}[i,j](k), \tag{6.6}
$$

for all $i \in \mathcal{I}_1^m$*, where* $A_{a_w}[i,j](k) = A_a[i,j]w_{i,j}(k),\ 0 \leq w_{i,j}(k) \leq (1-c_2)/(\max(\mathcal{D}[i,i],\mathcal{D}[j,j])),\ 0 < c_2 < 1$*. Then* $e(k)$ *exponentially convergences to* $\mathbf{0}$*, i.e.* $||e(k+1)||/||e(k)|| \leq \gamma$*, where* $0 \leq \gamma < 1$*.*

Proof. Eq. (6.6) is equivalent to following equation,

$$\begin{aligned} e(k+1) &= F(k)e(k), \text{ where} \\ F(k) &= I - \mathcal{L}_w(k). \end{aligned}$$

Note that $F(k)$ is a doubly stochastic symmetric matrix, where $\mathbf{1}^T F(k) = \mathbf{1}^T$, $F(k)\mathbf{1} = \mathbf{1}$, $F(k)^T = F(k)$ and non-zero, non-diagonal elements of $F(k)$ represents the weights of activated edges.

Condition 6.4(a) forces weights of the edges connecting non-negative valued vertices to other vertices to be at least c_1. However, the weights of some other edges may be 0. Thus, the weighted graph may not be connected, which means $\mathcal{L}_w(k)$ and $F(k)$ matrices may be reducible. Hence, $\mathcal{L}_w(k)$ may have repeated eigenvalues at 0, and $F(k)$ may have repeated eigenvalues at 1.

Even if $\mathcal{G}_w(k)$ may not be connected, due to Lemma 6.5, all non-negative valued vertices belong to a connected sub-graph in $\mathcal{G}_w(k)$, which consists of both non-negative and negative valued vertices. Note that in these connected sub-graphs, there is no edge between two negative valued vertices due to Condition 6.4(b). Therefore, weights of all edges are at least c_1 due to Condition 6.4(a), which enables us to apply Lemma 6.6 in later parts. Let us denote the number of these connected sub-graphs as $s(k) \geq 1$ and denote the reordered $F(k)$ matrix and $e(k)$ vector as

$$\begin{aligned}
\widehat{F}(k) &= P(k)F(k)P(k)^T \\
&= \begin{bmatrix} \widehat{F}_0(k) & & & \\ & \widehat{F}_1(k) & & \\ & & \ddots & \\ & & & \widehat{F}_{s(k)}(k) \end{bmatrix}, \\
\hat{e}(k) &= P(k)e(k) \\
&= (\hat{e}_0(k), \hat{e}_1(k), ..., \hat{e}_{s(k)}(k)),
\end{aligned}$$

where $P(k)$ is a permutation matrix, $\widehat{F}_i(k)$, $i \in I_s(k) = \{1, ..., s(k)\}$ are irreducible matrices that correspond to connected sub-graphs and $\widehat{F}_0(k)$ is a matrix that corresponds to remaining sub-graph. Note that there are both non-negative and negative elements in $\hat{e}_i(k)$, $i \in I_s(k)$ and all elements of $\hat{e}_0(k)$ are negative.

Recall that, in Example 1, $s(k) = 1$, values of v_1, v_2 and v_4 vertices belong to the $\hat{e}_1(k)$ vector and values of v_3, v_5 and v_6 vertices belong to the $\hat{e}_0(k)$ vector.

Suppose that, $\alpha_i(k)$ is the average of elements in $\hat{e}_i(k)$ such that $\alpha_i(k) = \mathbf{1}^T\hat{e}_i(k)/m_i(k)$, where $m_i(k)$ is the number of elements in $\hat{e}_i(k)$ and $\boldsymbol{\alpha}_i(k) = \mathbf{1}\alpha_i(k)$, $\boldsymbol{\alpha}_i(k) \in \mathbb{R}^{m_i(k)}$. Lemma 6.6 implies that only one eigenvalue of $\widehat{F}_i(k)$, $i \in I_s(k)$ is 1 with associated eigenvector $\mathbf{1}$, and other eigenvalues are in $[-(1 - c_{3_i}), (1 - c_{3_i})]$, $0 < c_{3_i} \leq 1$ since $\widehat{F}_i(k)$, $i \in I_s(k)$ matrices corresponds to a connected sub-graphs in which weights of all edges are at least $c_1 > 0$. Since the values of all vertices are negative in the remaining sub-graph that corresponds to the $\widehat{F}_0(k)$ matrix, there is no lower-bound for the weights of the edges, while there is an upper bound provided in Eq. (6.6). Hence, $\sigma(\widehat{F}_0(k)) \in [-(1 - c_{3_0}), 1]$, $0 < c_{3_0} \leq 1$. Therefore, the following equation holds,

$$\begin{aligned}
&||\widehat{F}_i(k)\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)||^2 = \\
&\qquad = ||\widehat{F}_i(k)\hat{e}_i(k) - \widehat{F}_i(k)\boldsymbol{\alpha}_i(k)||^2 \\
&\qquad = ||\widehat{F}_i(k)(\hat{e}_i(k) - \boldsymbol{\alpha}_i(k))||^2 \\
&\qquad = (\hat{e}_i(k) - \boldsymbol{\alpha}_i(k))^T \widehat{F}_i(k)^T \widehat{F}_i(k)(\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)) \\
&\qquad \leq c_4^2(\hat{e}_i(k) - \boldsymbol{\alpha}_i(k))^T(\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)) \\
&\qquad = c_4^2||\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)||^2,
\end{aligned} \tag{6.7}$$

where $c_4^2 = (1 - \min_i c_{3_i})^2 < 1$ for all $i \in I_s(k)$.

Here, we will show that $||\widehat{F}_i(k)\hat{e}_i(k)||^2$ is also smaller than or equal to $c_5^2||\hat{e}_i(k)||^2$ for all $i \in \mathcal{I}_1^m$, where $c_5^2 < 1$. Note that if $\mathbf{1}^T\boldsymbol{\alpha}_i(k) = \mathbf{1}^T\hat{e}_i(k)$, then $\boldsymbol{\alpha}_i(k)$ and $(\hat{e}_i(k) - \boldsymbol{\alpha}_i(k))$ are orthogonal to each other,

$$\begin{aligned}
\boldsymbol{\alpha}_i(k)^T(\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)) &= \boldsymbol{\alpha}_i(k)^T\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)^T\boldsymbol{\alpha}_i(k) \\
&= \alpha_i(k)\mathbf{1}^T\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)^T\boldsymbol{\alpha}_i(k) \\
&= \alpha_i(k)\mathbf{1}^T\boldsymbol{\alpha}_i(k) - \boldsymbol{\alpha}_i(k)^T\boldsymbol{\alpha}_i(k) \\
&= \boldsymbol{\alpha}_i(k)^T\boldsymbol{\alpha}_i(k) - \boldsymbol{\alpha}_i(k)^T\boldsymbol{\alpha}_i(k) \\
&= 0.
\end{aligned}$$

Hence, it can be shown that,

$$\begin{aligned}
||\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)||^2 &= ||\hat{e}_i(k)||^2 - 2\boldsymbol{\alpha}_i(k)^T\hat{e}_i(k) + ||\boldsymbol{\alpha}_i(k)||^2 \\
&= ||\hat{e}_i(k)||^2 - 2\boldsymbol{\alpha}_i(k)^T\boldsymbol{\alpha}_i(k) + ||\boldsymbol{\alpha}_i(k)||^2 \\
&= ||\hat{e}_i(k)||^2 - ||\boldsymbol{\alpha}_i(k)||^2,
\end{aligned}$$

and $||\widehat{F}_i(k)\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)||^2 =$

$$
\begin{aligned}
&= ||\widehat{F}_i(k)\hat{e}_i(k)||^2 - 2\boldsymbol{\alpha}_i(k)^T\widehat{F}_i(k)\hat{e}_i(k) + ||\boldsymbol{\alpha}_i(k)||^2 \\
&= ||\widehat{F}_i(k)\hat{e}_i(k)||^2 - 2\alpha_i(k)\mathbf{1}^T\widehat{F}_i(k)\hat{e}_i(k) + ||\boldsymbol{\alpha}_i(k)||^2 \\
&= ||\widehat{F}_i(k)\hat{e}_i(k)||^2 - 2\alpha_i(k)\mathbf{1}^T\hat{e}_i(k) + ||\boldsymbol{\alpha}_i(k)||^2 \\
&= ||\widehat{F}_i(k)\hat{e}_i(k)||^2 - 2\alpha_i(k)\mathbf{1}^T\boldsymbol{\alpha}_i(k)^T + ||\boldsymbol{\alpha}_i(k)||^2 \\
&= ||\widehat{F}_i(k)\hat{e}_i(k)||^2 - 2\boldsymbol{\alpha}_i(k)^T\boldsymbol{\alpha}_i(k) + ||\boldsymbol{\alpha}_i(k)||^2 \\
&= ||\widehat{F}_i(k)\hat{e}_i(k)||^2 - ||\boldsymbol{\alpha}_i(k)||^2.
\end{aligned}
$$

If $i \in I_s(k)$, then $\hat{e}_i(k) \neq \mathbf{0}$ because there are both non-negative and negative values in $\hat{e}_i(k)$. Therefore, Eq. (6.7) implies that

$$
\begin{aligned}
||\widehat{F}_i(k)\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)||^2 &\leq c_4^2||\hat{e}_i(k) - \boldsymbol{\alpha}_i(k)||^2 \\
||\widehat{F}_i(k)\hat{e}_i(k)||^2 - ||\boldsymbol{\alpha}_i(k)||^2 &\leq c_4^2(||\hat{e}_i(k)||^2 - ||\boldsymbol{\alpha}_i(k)||^2) \\
||\widehat{F}_i(k)\hat{e}_i(k)||^2 &\leq c_4^2||\hat{e}_i(k)||^2 + (1-c_4^2)||\boldsymbol{\alpha}_i(k)||^2 \\
\frac{||\widehat{F}_i(k)\hat{e}_i(k)||^2}{||\hat{e}_i(k)||^2} &\leq c_4^2 + (1-c_4^2)\frac{||\boldsymbol{\alpha}_i(k)||^2}{||\hat{e}_i(k)||^2}.
\end{aligned}
\tag{6.8}
$$

Here, we will show that $\frac{||\boldsymbol{\alpha}_i(k)||^2}{||\hat{e}_i(k)||^2} < \frac{m-1}{m}$ for all $i \in I_s(k)$. Suppose that, $\hat{e}_i(k) = P_{\hat{e}_i}(k)(\hat{e}_{i_p}(k), \hat{e}_{i_n}(k))$, where $P_{\hat{e}_i}(k)$ is a permutation matrix such that $\hat{e}_{i_p}(k) \geq 0$ and $\hat{e}_{i_n}(k) < 0$. Let $m_{i_p}(k)$ and $m_{i_n}(k)$ represent the number of elements in $\hat{e}_{i_p}(k)$ and $\hat{e}_{i_n}(k)$, where $m_{i_p}(k) + m_{i_n}(k) = m_i(k)$. Since there are both non-negative and negative values in

$\hat{e}_i(k)$, $m_{i_p}(k) \geq 1$, $m_{i_n}(k) \geq 1$ and $\mathbf{1}^T \hat{e}_i(k) < ||\hat{e}_i(k)||_1$, then

$$\begin{aligned}
\frac{||\boldsymbol{\alpha}_i(k)||^2}{||\hat{e}_i(k)||^2} &= \frac{\boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i}{||\hat{e}_i(k)||^2} \\
&= \frac{\mathbf{1}^T \mathbf{1} (\mathbf{1}^T e_i(k))^2}{m_i^2(k) ||\hat{e}_i(k)||^2} \\
&= \frac{(\mathbf{1}^T e_i(k))^2}{m_i(k) ||\hat{e}_i(k)||^2} \\
&< \frac{(\mathbf{1}^T \hat{e}_{i_p}(k))^2 + (\mathbf{1}^T \hat{e}_{i_n}(k))^2}{m_i(k) ||\hat{e}_i(k)||^2} \\
&= \frac{||\hat{e}_{i_p}(k)||_1^2 + ||\hat{e}_{i_n}(k)||_1^2}{m_i(k) ||\hat{e}_i(k)||^2} \\
&\leq \frac{m_{i_p}(k) ||\hat{e}_{i_p}(k)||^2 + m_{i_n}(k) ||\hat{e}_{i_n}(k)||^2}{m_i(k) ||\hat{e}_i(k)||^2} \\
&\leq \frac{\max(m_{i_p}(k), m_{i_n}(k)) (||\hat{e}_{i_p}(k)||^2 + ||\hat{e}_{i_n}(k)||^2)}{m_i(k) ||\hat{e}_i(k)||^2} \\
&= \frac{\max(m_{i_p}(k), m_{i_n}(k))}{m_i(k)} \\
&\leq \frac{m_i(k) - 1}{m_i(k)} \leq \frac{m-1}{m}
\end{aligned}$$

for all $i \in I_s(k)$. It then follows that Eq. (6.8) implies

$$||\widehat{F}_i(k) \hat{e}_i(k)||^2 \leq c_5^2 ||\hat{e}_i(k)||^2 \ \text{ for all } \ i \in I_s(k),$$

where $c_5^2 < c_4^2 + (1 - c_4^2) \frac{m-1}{m} < 1$.

Here, we will present that $||e(k+1)|| \leq c_6 ||e(k)||$, where $0 < c_6 < 1$. Since $e(k) \neq \mathbf{0}$,

$$\frac{||e(k+1)||^2}{||e(k)||^2} = \frac{||\widehat{F}(k) \hat{e}(k)||^2}{||\hat{e}(k)||^2} = \frac{\sum_{i=0}^{s(k)} ||\widehat{F}_i(k) \hat{e}_i(k)||^2}{\sum_{i=0}^{s(k)} ||\hat{e}_i(k)||^2}.$$

Since $\sigma(\widehat{F}_0(k)) \subseteq [-(1 - c_{3_0}), 1]$, $0 < c_{3_0} \leq 1$, the equation is derived as

$$\frac{||e(k+1)||^2}{||e(k)||^2} \leq \frac{||\hat{e}_0(k)||^2 + c_5^2 \sum_{i=1}^{s(k)} ||\hat{e}_i(k)||^2}{||\hat{e}_0(k)||^2 + \sum_{i=1}^{s(k)} ||\hat{e}_i(k)||^2}. \tag{6.9}$$

Here, we derive an upper bound for $||\hat{e}_0(k)||^2$ with respect to other terms to show that, there is an upper bound for the right side of Eq. (6.9). Since $\mathbf{1}^T e(k) = 0$ and $\hat{e}_0(k) < \mathbf{0}$,

$||\hat{e}_0(k)||_1 + \sum_{i=1}^{s(k)} ||\hat{e}_{i_n}(k)||_1 = \sum_{i=1}^{s(k)} ||\hat{e}_{i_p}(k)||_1$. Note that $||\hat{e}_{i_n}(k)||_1 > 0$ for any $i \in I_s(k)$ due to Lemma 6.5. Then,

$$\begin{aligned} ||\hat{e}_0(k)||_1 &< \sum_{i=1}^{s(k)} ||\hat{e}_{i_p}(k)||_1 + \sum_{i=1}^{s(k)} ||\hat{e}_{i_n}(k)||_1 \\ &= \sum_{i=1}^{s(k)} ||\hat{e}_i(k)||_1. \end{aligned} \tag{6.10}$$

The following inequalities can be provided using equivalence of norms,

$$||\hat{e}_0(k)||^2 \leq ||\hat{e}_0(k)||_1^2 \leq m_0||\hat{e}_0(k)||^2,$$
$$\sum_{i=1}^{s(k)} ||\hat{e}_i(k)||^2 \leq \sum_{i=1}^{s(k)} ||\hat{e}_i(k)||_1^2 \leq (m - m_0) \sum_{i=1}^{s(k)} ||\hat{e}_i(k)||^2.$$

Thus, Eq. (6.10) implies that

$$||\hat{e}_0(k)||^2 < (m - m_0) \sum_{i=1}^{s(k)} ||\hat{e}_i(k)||^2.$$

The strict upper-bound for $||\hat{e}_0(k)||^2$ can be inserted into Eq. (6.9) as

$$\begin{aligned} \frac{||e(k+1)||^2}{||e(k)||^2} &\leq 1 - \frac{(1 - c_5^2) \sum_{i=1}^{s(k)} ||\hat{e}_i(k)||^2}{||\hat{e}_0(k)||^2 + \sum_{i=1}^{s(k)} ||\hat{e}_i(k)||^2} \\ &< 1 - \frac{(1 - c_5^2) \sum_{i=1}^{s(k)} ||\hat{e}_i(k)||^2}{(m - m_0 + 1) \sum_{i=1}^{s(k)} ||\hat{e}_i(k)||^2} \\ &= \frac{m - m_0 + c_5^2}{m - m_0 + 1}. \end{aligned}$$

Since $0 \leq c_5^2 < 1$ and $0 \leq m_0 \leq m - 2$,

$$\frac{||e(k+1)||}{||e(k)||} \leq \gamma,$$

where $\gamma < \sqrt{\max_{0 \leq m_0 \leq m-2} \frac{m - m_0 + c_5^2}{m - m_0 + 1}} = \sqrt{\frac{m + c_5^2}{m+1}} < 1$. Therefore, $||e(k)||$ exponentially converges to 0. ■

6.3 Decentralized State-Dependent Markov Chain Synthesis

Based on the consensus protocol developed in Section 6.2, we propose the decentralized state-dependent Markov chain synthesis (DSMC) algorithm that achieves convergence to the desired distribution with an exponential rate and minimal state transitions. Additionally, we present a shortest path algorithm that can be integrated with the DSMC algorithm, as utilized in [3, 21, 141], to further enhance the convergence rate.

6.3.1 The DSMC Algorithm

We define a finite-state discrete-time Markov chain evolving on the vertices of the uniform graph $\mathcal{G}$ in Definition 6.1.

Definition 6.8. *(Markov chain) Let $\mathcal{V} = \{\mathrm{v}_1, ..., \mathrm{v}_m\}$, which is the vertices of the graph in Definition 6.1, be the finite set of states of the Markov chain and $\mathcal{X}(k) \in \mathcal{V}$ be the random variable of the Markov chain for time $k \in \mathbb{Z}_+$. The connectivity of the states is determined by the adjacency matrix A_a of the uniform graph $\mathcal{G}$. Additionally, we define:*

- *(Probability distribution) Let $x(k) \in \mathbb{R}^m$ be the probability distribution at time k : $x[i](k) = \mathcal{P}(\mathcal{X}(k) = \mathrm{v}_i)$.*
- *(Markov matrix) Markov matrix determines the state transitions of the Markov chain $M(k) \in \mathbb{R}^{m\times m}$, where $M[i,j](k) = \mathcal{P}(\mathcal{X}(k+1) = \mathrm{v}_i|\mathcal{X}(k) = \mathrm{v}_j)$ for all $i, j \in \mathcal{I}_1^m$. Hence $x(k+1) = M(k)x(k)$ for $k \in \mathbb{Z}_+$.*
- *(Desired steady-state distribution) There exists a prescribed desired steady-state probability distribution $v \in \mathbb{R}^m$ such that*

$$\lim_{k\to\infty} x(k) = v.$$

Let the error vector $e(k)$ be the difference between the probability distribution at time k and the desired steady-state distribution $e(k) = x(k) - v$. The DSMC algorithm is designed to ensure that the dynamics of the error vector are identical to the dynamics of the value vector described in Theorem 6.7. This design guarantees consistency between the two, ensuring desirable convergence properties and performance in the DSMC algorithm. Similar to the consensus protocol in Section 6.2, transitions in the DSMC algorithm occur from the states with higher error values to their adjacent states with lower error values to equalize the error values across all states of the Markov chain. Since the sum of these error values is equal to 0, the error values at all states will eventually become balanced at 0, resulting in the convergence of the probability distribution to the desired distribution.

We present the synthesis of the Markov matrix in Algorithm 3, and its convergence proof is presented in Theorem 6.12. To this end, let I_{ij} be the set of indices of states adjacent to either v_i or v_j, such that $I_{ij} = \{l : A_a[l,i] = 1 \text{ or } A_a[l,j] = 1\}$. It should be noted that $e[\ell] < 0$ for all $\ell \in \{i,j\}$ and $e[l](k) \geq 0$ for some $l \in I_{ij}$ if and only if $\{i,j\} \in I_{n_p,n}(k)$, which is introduced in Definition 6.3. To determine the transition probability from state v_j to state v_i for time k, the required inputs of the algorithm are $x[l](k)$ and $v[l]$ for all $l \in I_{ij}$, the corresponding rows of the adjacency matrix for states v_i and v_j, and a hyper-parameter $c_2 \in (0,1)$ that scales the transition probabilities. In Line 1 of Algorithm 3, the algorithm computes the error values at the states associated with the set I_{ij}. From the state v_j, there will be a transition to any adjacent state v_i only if $e[j](k) > e[i](k)$. Also, the transition probabilities vary proportionally with the differences in error values. Therefore, the difference of the error value $e[j](k)$ from the error value $e[i](k)$ is calculated in Line 2 if $e[j](k) > e[i](k)$, otherwise it is set to 0. As in Theorem 6.7, appropriately scaling the differences between error values relative to the number of adjacent states is crucial for the stability of the convergence. Also, it is required to scale these differences with respect to the probabilities of the corresponding states. For this purpose, a scaling factor is determined in Line 4 for the stability of the convergence, while another scaling factor is determined by the probability of the state v_j in Line 5. Furthermore, if $\sum_{l \in \mathcal{I}_1^m} E[l,j](k) = 0$, then there will be no transition from state v_j to any adjacent state v_i. In this case, the value of $s_2[j](k)$ becomes irrelevant and is set to 1. If $e[i](k) < 0$ and $e[j](k) < 0$, and either v_i or v_j has an adjacent state with a positive error value, then another scaling factor $s_3[i,j](k)$ is set to 0 in Line 6 to satisfy Condition 6.4(b). The minimum of these three scaling factors is chosen in Line 7 to satisfy all conditions simultaneously. Hence, the scaled difference of the error value $e[j](k)$ from the error value $e[i](k)$ is calculated in Line 8. Note that $T[i,j](k)$ represents the amount of increase in the error value at state v_i due to the transition from state v_j. Equivalently, it represents the amount of decrease of the error value at state v_j due to the transition to the state v_i. This amount is divided by the probability of the state v_j in Line 9 to decide the transition probability from state v_j to state v_i. In Line 10, the remaining probabilities are added to the diagonal elements, finalizing the synthesis of the Markov matrix. The complexity of the DSMC algorithm is $O(m^2)$, where m is the number of states.

The following proposition shows that the matrix synthesized by Algorithm 3 is a column stochastic Markov matrix.

Proposition 6.9. *The matrix synthesized by Algorithm 3 satisfies $M(k) \geq \mathbf{0}$ and $\mathbf{1}^T M(k) = \mathbf{1}^T$ constraints.*

Proof. (Proof of $M(k) \geq \mathbf{0}$) $W(k) \geq \mathbf{0}$ since $s_1[i,j] > 0$, $s_2[j](k) \geq 0$ and $s_3[i,j](k) \geq 0$, $\forall i,j \in \mathcal{I}_1^m$. Since $E(k) \geq \mathbf{0}$ and $W(k) \geq \mathbf{0}$, $T(k) \geq \mathbf{0}$. $x(k) \geq \mathbf{0}$ and $T(k) \geq \mathbf{0}$ imply that $M[i,j](k) = F[i,j](k) \geq 0$, $\forall i,j \in \mathcal{I}_1^m$ if $i \neq j$.

If $x[j](k) = 0$ or $\sum_{i\in\mathcal{I}_1^m} E[i,j](k) = 0$, then $\sum_{i\in\mathcal{I}_1^m} F[i,j](k) = 0$ and $M[j,j](k) = 1$. If $x[j](k) \neq 0$ and $\sum_{i\in\mathcal{I}_1^m} E[i,j](k) \neq 0$, then

$$\begin{aligned}
\sum_{i\in\mathcal{I}_1^m} F[i,j](k) &= \frac{\sum_{i\in\mathcal{I}_1^m} T[i,j](k)}{x[j](k)} \\
&= \frac{\sum_{i\in\mathcal{I}_1^m} E[i,j](k) W[i,j](k)}{x[j](k)} \\
&\leq \frac{\sum_{i\in\mathcal{I}_1^m} E[i,j](k)}{x[j](k)} s_2[j](k) \\
&= \frac{\sum_{i\in\mathcal{I}_1^m} E[i,j](k)}{x[j](k)} \frac{x[j](k)}{\sum_{i\in\mathcal{I}_1^m} E[i,j](k)} \\
&= 1.
\end{aligned}$$

Therefore, $\sum_{i\in\mathcal{I}_1^m} F[i,j](k) \leq 1$ and $M[j,j](k) \geq 0$.

(Proof of $\mathbf{1}^T M(k) = \mathbf{1}^T$) $\mathbf{1}^T M(k) = \mathbf{1}^T$ requires that $\sum_{i\in\mathcal{I}_1^m} M[i,j](k) = 1$, $\forall j \in \mathcal{I}_1^m$. Note that $\sum_{i\in\mathcal{I}_1^m\setminus\{j\}} F[i,j](k) = \sum_{i\in\mathcal{I}_1^m} F[i,j](k)$ since $F[i,i](k) = 0$. Then, $\sum_{i\in\mathcal{I}_1^m} M[i,j](k) = \sum_{i\in\mathcal{I}_1^m\setminus\{j\}} F[i,j](k) + 1 - \sum_{i\in\mathcal{I}_1^m} F[i,j](k) = 1$. This shows that $\mathbf{1}^T M(k) = \mathbf{1}^T$. ∎

The following proposition shows that the error dynamics in Algorithm 3 are identical to the evolution of the value vector in the consensus protocol presented in Theorem 6.7.

Proposition 6.10. *The error dynamics of Algorithm 3 satisfy Eq. (6.6).*

Proof. In Algorithm 3, the increase in the error value at state v_i caused by state v_j is denoted by $T[i,j](k)$ in Line 8. Equivalently, $T[i,j](k)$ represents the decrease in the error value at state v_j caused by state v_i. Since $\sum_{i\in\mathcal{I}_1^m} F[i,j](k) \leq 1$ in Line 9 of Algorithm 3, the amount of change determined in Line 8 is not suppressed by $x(k)$. Thus, the error value at the state v_i changes as

$$
\begin{aligned}
e[i](k+1) &= e[i](k) + \sum_{j\in\mathcal{I}_1^m} T[i,j](k) - \sum_{j\in\mathcal{I}_1^m} T[j,i](k) \\
&= e[i](k) + \sum_{j\in\mathcal{I}_1^m} E[i,j](k)W[i,j](k) - \sum_{j\in\mathcal{I}_1^m} E[j,i](k)W[j,i](k) \\
&= e[i](k) + \sum_{\substack{j\in\mathcal{I}_1^m \\ e[j](k)\geq e[i](k)}} \Big(e[j](k) - e[i](k)\Big) A_a[i,j]W[i,j](k) \\
&\quad - \sum_{\substack{j\in\mathcal{I}_1^m \\ e[j](k)< e[i](k)}} \Big(e[i](k) - e[j](k)\Big) A_a[j,i]W[j,i](k).
\end{aligned}
\tag{6.11}
$$

Let us denote $w_{i,j}(k)$ as

$$
w_{i,j}(k) = W[i,j](k) \text{ if } e[j](k) \geq e[i](k). \tag{6.12}
$$

Note that $0 \leq W[i,j](k) \leq s_1[i,j] = (1-c_2)/\max(d[i], d[j])$, for all $i,j \in \mathcal{I}_1^m$, where $0 < c_2 < 1$ due to Line 4 and Line 7 of the Algorithm 3. Since $d[i]$ in Line 3 of the Algorithm 3 represents the degree similar to $\mathcal{D}[i,i]$ in Definition 6.1, $w_{i,j}(k)$ satisfies the constraints given in Theorem 6.7. Thus, Eq. (6.11) implies that

$$
e[i](k+1) = e[i](k) + \sum_{j\in\mathcal{I}_1^m} \Big(e[j](k) - e[i](k)\Big) A_{a_w}[i,j](k) \tag{6.13}
$$

for all $i \in \mathcal{I}_1^m$, where $A_{a_w}[i,j](k) = A_a[i,j]w_{i,j}(k)$, $0 \leq w_{i,j}(k) < (1-c_2)/(\max(\mathcal{D}[i,i],\mathcal{D}[j,j]))$, $0 < c_2 < 1$. Since Eq. (6.13) is identical to Eq. (6.6), the error dynamics of Algorithm 3 are identical to the evolution of the value vector in Theorem 6.7. ■

Note that Assumption 6.2 is already satisfied for the DSMC algorithm due to Assumption 6.18. We now show that Condition 6.4, which is crucial for the convergence of the proposed consensus protocol, is satisfied by Algorithm 3.

Proposition 6.11. *Condition 6.4 is satisfied by Algorithm 3.*

Proof. To ensure Condition 6.4(a) it must be shown that there exists a c_1 such that $0 < c_1 \leq w_{i,j}(k)$ if $0 \leq e[i](k)$. Then, due to Eq. (6.12), it is sufficient to show that there exists a c_1 such that $0 < c_1 \leq W[i,j](k)$ if $0 \leq e[j](k)$.

(Proof of Condition 6.4(a)) Since $s_1[i,j] > 0$ and it is not time-varying, there exists a c_1' such that $0 < c_1' \leq s_1[i,j]$. The inequality $0 \leq e[j](k)$ implies that $s_3[i,j] = 1$ for all $i \in \mathcal{I}_1^m$ due to Line 6 of the Algorithm 3. Note that

$$\begin{aligned}\sum_{i\in\mathcal{I}_1^m} E[i,j](k) &= \sum_{i\in\mathcal{I}_1^m}\Big(0, \Big(e[j](k) - e[i](k)\Big)A_a[i,j]\Big)\\ &\leq \sum_{i\in\mathcal{I}_1^m} |e[j](k) - e[i](k)|\\ &\leq (m-1)|e[j](k)| + \sum_{i\in\mathcal{I}_1^m\setminus\{j\}} |e[i](k)|.\end{aligned}$$

Since $||e[i](k)||_\infty \leq 1$, $\sum_{i\in\mathcal{I}_1^m} |e[i](k)| \leq 2$, and $m \geq 2$, then

$$\begin{aligned}\sum_{i\in\mathcal{I}_1^m} E[i,j](k) &\leq (m-1)|e[j](k)| + (2 - |e[j](k)|)\\ &= m|e[j](k)| + 2 - 2|e[j](k)|\\ &\leq m.\end{aligned}$$

It then follows that Line 5 of the Algorithm 3 implies

$$\frac{x[j](k)}{m} = \min\left(\frac{x[j](k)}{m}, 1\right) \leq \min\left(\frac{x[j](k)}{\sum_{i\in\mathcal{I}_1^m} E[i,j](k)}, 1\right) \leq s_2[j](k).$$

Let $v_{\min} = \min_{i \in \mathcal{I}_1^m} v[i]$. Then, the inequality $0 \leq e[j](k)$ implies that $v_{\min} \leq x[j](k)$. Thus, $v_{\min}/m \leq x[j](k)/m \leq s_2[j](k)$. Since $v_{\min}/m$ is not time-varying, there exists a c_1'' such that $0 < c_1'' \leq v_{\min}/m \leq s_2[j](k)$. Hence, if $0 \leq e[j](k)$, then there exist a c_1 such that $0 < c_1 \leq \min\Big(s_1[i,j], s_2[j](k), s_3[i,j]\Big) = W[i,j](k)$, where $c_1 = \min(c_1', c_1'', 1)$.

(Proof of Condition 6.4(b)) Condition 6.4(b) requires that $w_{i,j} = 0$ if $\{i,j\} \in I_{n_p,n}$, where $I_{n_p,n}$ is the set defined in Definition 6.3. Note that $\{i,j\} \in I_{n_p,n}$ if and only if $e[\ell] < 0$ for all $\ell \in \{i,j\}$ and $e[l] \geq 0$ for some $l \in I_{ij}$. Therefore, if $\{i,j\} \in I_{n_p,n}$, both $s_3[i,j](k)$ and $s_3[j,i](k)$ are set to 0, which imply that both $W[i,j](k)$ and $W[j,i](k)$ are 0. Hence, $w_{i,j}(k) = 0$ if $\{i,j\} \in I_{n_p,n}$. ■

The following theorem shows that Algorithm 3 achieves the desired convergence.

Theorem 6.12. *The probability distribution $x(k)$ exponentially converges to the desired distribution v if the corresponding Markov matrix is synthesized by Algorithm 3.*

Proof. In Proposition 6.10, it is proven that the dynamics of the error vector in Algorithm 3 are identical to the dynamics of the value vector in Theorem 6.7. Condition 6.4 is used in Theorem 6.7 to prove that the value vector exponentially converges to $\mathbf{0}$. In Proposition 6.11, it is proven that Algorithm 3 satisfies Condition 6.4 which means that probability distribution $x(k)$ exponentially converges to the desired distribution v. ■

6.3.2 The Modified DSMC Algorithm

In this section, we introduce a shortest-path algorithm that is proposed as a modification to the Metropolis-Hastings algorithm in [3, Section V-E] and integrated with the Markov chain synthesis methods described in [21] and [141]. This algorithm can also be integrated with the DSMC algorithm to further increase the convergence rate. Our approach begins by categorizing the states of the desired distribution.

Definition 6.13. *(Recurrent and transient states) The states with non-zero elements in the desired distribution v are called recurrent states. The other states with zero elements in the desired distribution v are called transient states.*

The shortest-path algorithm is used for the synthesis of the Markov chain for the transient states while the Markov chain for the recurrent states is synthesized by the DSMC algorithm. Let us split the desired distribution and the Markov matrix to synthesize the Markov matrix for recurrent and transient states separately. Assume that there are m_r recurrent states and $m_t = m - m_r$ transient states of the desired distribution. Then the Markov matrix and the desired distribution are split as

$$v = (v_t, v_r), \quad M(k) = \begin{bmatrix} M_1 & \mathbf{0} \\ M_2 & M_3(k) \end{bmatrix}_{m\times m}, \tag{6.14}$$

where $v_t = \mathbf{0}$, $v_t \in \mathbb{R}^{m_t}$, $v_r > \mathbf{0}$, $v_r \in \mathbb{R}^{m_r}$, $M_1 \in \mathbb{R}^{m_t\times m_t}$ and $M_3(k) \in \mathbb{R}^{m_r\times m_r}$. Since the desired distribution and the Markov matrix are partitioned by renumbering the elements, the probability distribution and the adjacency matrix are also partitioned using the same renumbering as

$$x(k) = (x_t(k), x_r(k)), \quad A_a = \begin{bmatrix} A_{a_1} & \mathbf{0} \\ A_{a_2} & A_{a_3} \end{bmatrix}_{m\times m}.$$

Note that while the time-invariant matrices M_1 and M_2 in Eq. (6.14) are synthesized by the shortest-path algorithm, the time-varying Markov matrix $M_3(k)$ is synthesized by the DSMC algorithm. If the shortest-path is to be combined with another Markov chain synthesis algorithm, it is also necessary to assume that the recurrent states are connected among themselves, which requires the following assumption.

Assumption 6.14. *The graph defined by the recurrent states is connected, which means there exists a path between all recurrent states, or equivalently, $(I + A_{a_3})^{m_r-1} > \mathbf{0}$ for $A_{a_3} \in \mathbb{R}^{m_r\times m_r}$ [201, section 2.1].*

Furthermore, we demonstrate that the condition $\mathbf{1}^T x_r(k) = 1$, which is crucial for the Markov chain synthesis algorithm of the recurrent states, is satisfied for all $k \geq K$, $K \in \mathbb{Z}_+$ in the shortest-path algorithm.

Let us first define the following index sets to present the algorithm.

Definition 6.15. *(Index sets with respect to recurrent states) The index sets I_r and I_t contain the indices of the recurrent and transient states, respectively. Indices of transient states can be split into subsets as I_s, $I_{s-1}, \ldots$, and so on. The index set I_s consists of the indices of the states that are adjacent to the states corresponding to I_r and $I_s \cap I_r = \emptyset$, the index set I_{s-1} consists of the indices of states that are adjacent to the states corresponding to I_s and $I_{s-1} \cap (I_r \cup I_s) = \emptyset$, and so on.*

The synthesis of the Markov matrix using the DSMC algorithm and the shortest-path algorithm for the recurrent and transient states of the desired distribution is given as

$$M[i,j] = \begin{cases} \text{as in Algorithm 3} & \text{if } i \in I_r,\ j \in I_r \\ 0 & \text{if } i \in I_{s-n},\ j \in I_r \text{ for } n \in \mathbb{Z}_+ \\ 1/(\sum_{l \in I_r} A_a[j,l]) & \text{if } i \in I_r,\ j \in I_s,\ A_a[j,i] = 1 \\ 1/(\sum_{l \in I_{s-n}} A_a[j,l]) & \begin{array}{l} \text{if } i \in I_{s-n},\ j \in I_{s-n-1},\ A_a[j,i] = 1 \\ \text{for } n \in \mathbb{Z}_+ \end{array} \end{cases}$$

In the proposed shortest-path algorithm, the transition probability from any state in set I_{s-k-1} to any state in set I_{s-k}, $k \in \mathbb{Z}_+$, and from any state in set I_s to any state in set I_r is 1. Therefore, the total probability of the transient states becomes zero in a finite time. In [3], it is shown that the condition $\rho(M_1) < 1$ is satisfied using the properties of M-matrices, which are shown in Theorem 2.5.3 (parts 2.5.3.2 and 2.5.3.12) of [134].

6.4 An application of the DSMC Algorithm on Probabilistic Swarm Guidance

In this section, we apply the DSMC algorithm to the probabilistic swarm guidance problem and provide numerical simulations that show the convergence rate of the DSMC algorithm is considerably faster as compared to the previous Markov chain synthesis algorithms in [3] and [21].

6.4.1 Probabilistic Swarm Guidance

Most of the underlying definitions and baseline algorithms in this section are based on [3]. We briefly review this material for completeness.

Definition 6.16. *(Bins) The operational region is denoted as $\mathcal{R}$. The region is assumed to be partitioned as the union of m disjoint regions, which are referred to as bins R_i, $i = 1, ..., m$, such that $\mathcal{R} = \cup_{i=1}^{m} R_i$, and $R_i \cap R_j = \varnothing$ for $i \neq j$. Each bin can contain multiple agents.*

Bins, as in the states of a Markov chain, in the operational region are represented as vertices of a graph. The allowed transitions between bins are specified by the adjacency matrix of the graph.

Definition 6.17. *(Transition constraints) An adjacency matrix $A_a \in \mathbb{R}^{m\times m}$ is used to restrict the allowed transitions between bins. $A_a[i,j] = 1$ if the transition from R_i to R_j is allowable, and is 0 otherwise. The bins R_i and R_j are called neighboring bins if $A_a[i,j] = 1$.*

Assumption 6.18. *The graph describing the bin connections is connected, that is, $(I + A_a)^{m-1} > \mathbf{0}$ for $A_a \in \mathbb{R}^{m\times m}$ [201, section 2.1].*

Consider a swarm comprised of N agents. Let $n[i](k)$ be the number of agents in bin R_i at time k. Then $x(k) = n(k)/N$ represents the swarm density distribution at time k where $x(k) \geq \mathbf{0}$ and $\mathbf{1}^T x(k) = 1$. It is desired to guide the swarm density distribution to a *desired steady-state distribution* $v \in \mathbb{R}^m$, where $v \geq \mathbf{0}$ and $\mathbf{1}^T v = 1$. The distance between the swarm density distribution and the desired distribution is monitored via the total variation between the swarm density distribution at time k and the desired distribution

$$T_{x,v}(k) = \frac{1}{2}||x(k) - v||_1.$$

Assumption 6.19. *All agents can determine their current bins, and they can use this information to move between bins.*

Each swarm agent changes its bin at time k by using a column stochastic, Markov, matrix $M(k) \in \mathbb{R}^{m\times m}$ called a Markov matrix [135]. The entries of matrix $M(k)$ define the transition probabilities between bins, that is, for any $k \in \mathbb{Z}_+$ and $i, j \in \mathcal{I}_1^m$,

$$M[i,j](k) = \mathcal{P}(r(k+1) \in R_i | r(k) \in R_j),$$

i.e., an agent in R_j moves to R_i with probability $M[i,j](k)$ at time k. Algorithm 4 is an implementation of this probabilistic guidance algorithm. The first step of the algorithm is to determine the agent's current bin. In the following steps, each agent samples from a uniform discrete distribution and goes to another bin depending on the column of the Markov matrix, which is determined by the agent's current bin.

The main idea of the probabilistic swarm guidance is to drive the propagation of density distribution vector $x(k)$, instead of individual agent positions $\{r_l(k)\}_{l=1}^N$. Swarms are viewed as a statistical ensemble of agents to facilitate the swarm guidance problem. The density distribution of the swarm $x(k)$ satisfies the properties of a probability distribution, which are $x(k) \geq \mathbf{0}$, and $\mathbf{1}^T x(k) = 1$. However, the number of agents is finite and, using the law of large numbers [28, Section V], the Algorithm 4 ensures that $x(k+1) = M(k)\, x(k)$ is satisfied as $N \to \infty$. Furthermore, any given desired distribution $v \in \mathbb{R}^m$ cannot always be accurately represented by the finite number of agents of the swarm. For example, if $v = [0.5, 0.5]^T$ and $N = 1$, then the best-quantized representation of the distribution that can be obtained is either $[1, 0]^T$ or $[0, 1]^T$. In this case, the total variation between the density distribution of the swarm and the desired distribution cannot be less than 0.5.

Definition 6.20. *(Quantization error) The quantization error is the minimum total variation value that can be achieved by the density distribution of the swarm $x(k) = n(k)/N$ to represent a given desired distribution v. It is denoted as $q_N(v)$ and is the solution to the following problem*

$$q_N(v) = \min_n \left\{ \frac{1}{2} ||n/N - v||_1 : n \in \mathbb{Z}_+^m, \mathbf{1}^T n = N \right\}.$$

Let Q_N be the maximum value of the quantization error that considers the worst possible desired distribution v for a given number of agents N such that

$$Q_N = \max_v \{q_N(v) : v \in \mathbb{R}^m_+, \mathbf{1}^T v = 1\}.$$

The following theorem provides the maximum quantization error value for a given number of agents N.

Lemma 6.21. *For any given desired distribution $v \in \mathbb{R}^m_+$, $\mathbf{1}^T v = 1$, the maximum quantization error value between $x(k) = n(k)/N$ and v is not greater than $m/(4N)$, where N is the total number of agents.*

Proof. Let us split the integer and fractional part of Nv such that $Nv = t + f$, where $t \in \mathbb{Z}^m_+$ and $\mathbf{0} \leq f < \mathbf{1}$. Let $s = \mathbf{1}^T f$. Note that $s \in \mathbb{Z}_+$ and $s = N - \mathbf{1}^T t$. Let $n^*[i]$ be the optimal value of $n[i]$ that minimizes $|n[i] - Nv[i]|$. Then, $n^*[i] = t[i]$ or $n^*[i] = t[i] + 1$. Without loss of generality, suppose that $f[1] \geq f[2] \geq \cdots \geq f[m]$. Then $n^*[i] = \begin{cases} t[i] + 1 & \text{if } i \leq s \\ t[i] & \text{otherwise} \end{cases}$, where $\mathbf{1}^T n = \mathbf{1}^T t + s = N$. Therefore $||n - vN||_1 = (1 - f[1]) + \cdots + (1 - f[k]) + f[k+1] + \cdots + f[m]$. Since $s = \mathbf{1}^T f$, $||n - vN||_1 = 2(f[k+1] + \cdots + f[m])$. Also, $f[k+1] \leq s/m$ since $f[i] \geq f[i+1]$ for all $i \in \mathcal{I}^m_1$ and $\mathbf{1}^T f = s$. Therefore, $||n - vN||_1 \leq 2(m - s)(s/m)$. The maximum value of $2(m - s)(s/m)$ is $m/4$, where $s = m/2$. Hence, $||n - vN||_1 \leq m/2$ and $\frac{1}{2}||n/N - v||_1 \leq m/(4N)$. ■

6.4.2 Synthesis of the Markov Matrix

The DSMC algorithm presented in Section 6.3.1 is employed for synthesizing the Markov chain, which serves as the stochastic policy for the swarm agents. It is worth noting that the bins comprising the operational region, as defined in Definition 6.16, determine the vertices of the uniform graph in Definition 6.1. Consequently, these vertices correspond to the states of the Markov chain defined in Definition 6.8. Similarly, the transition constraints of the swarm, defined by an adjacency matrix in Definition 6.17, determine the adjacency matrix of the uniform graph in Definition 6.1, which subsequently corresponds to the adjacency matrix of the Markov chain in Definition 6.8. The connectivity requirement for the vertices of the consensus protocol, as outlined in Assumption 6.2, is satisfied by Assumption 6.18. Furthermore, if the recurrent states of the desired distribution are also connected among themselves, then Assumption 6.14 is satisfied, allowing the utilization of the Modified DSMC algorithm presented in Section 6.3.2 for Markov chain synthesis. Since the DSMC algorithm is state-dependent, we use the density distribution of the swarm for feedback, which requires the following assumption.

Assumption 6.22. *All agents know the density values of their own and neighboring bins.*

A complex communication architecture is not required since communication only with neighboring bins is sufficient for an agent to determine its transition probabilities. If agents have only access to the number of agents of their own and neighboring bins, then they also need to know the total number of agents in the swarm, which is global information, to determine their density values. Distributed consensus algorithms, which work under the strongly-connected communication network topology assumption, can be used to estimate the total number of agents of the swarm. The consensus protocol that is used to estimate the density distribution of the swarm in [21, Remark 13] can also be used to estimate the total number of agents of the swarm.

6.4.3 Numerical Simulations

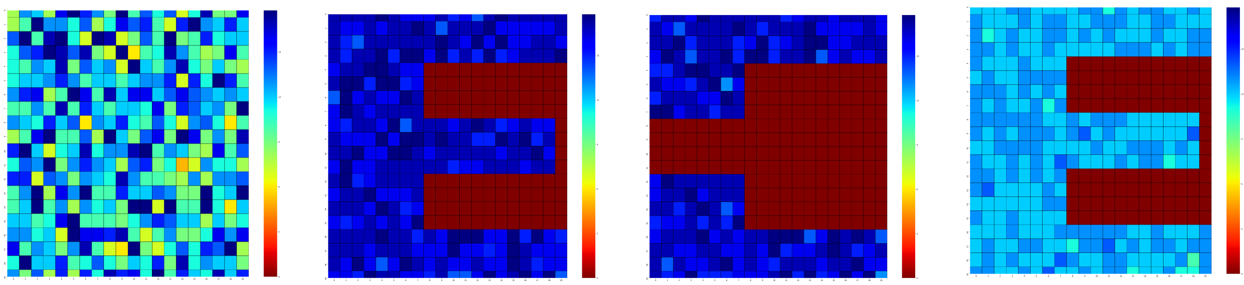
The convergence performance of the DSMC algorithm is demonstrated on the probabilistic swarm guidance application with numerical simulations shown in Figure 6.2 and 6.5. We monitor the total variation between the density distribution of the swarm and the desired distribution. The finite number of agents causes the quantization error outlined in Definition 6.20. Theorem 6.21 limits the quantization error to $m/(4N)$, where m is the number of bins and N is the number of agents.

In the first simulation, which is the same numerical example presented in [3], agents converge to the letter “E" in 250 time-steps. Then approximately 1/3 agents are removed and the remaining agents converge to the desired distribution again in 500 time-steps, demonstrating the “self-repair" property of the proposed algorithm for swarm guidance. Comparisons of the total variation and the total number of transitions for the M-H, PSG-IMC, and the DSMC algorithms are given in Figures 6.3, 6.4, and Tables 6.1, 6.2 for two different cases. In the first case, the adjacency matrix only allows transitions to the bins, which are above, below, right, and left to a given bin, i.e., the bins that are 1-step away. In the second case, the adjacency matrix allows transitions to 10-step away bins. When compared to the M-H and PSG-IMC algorithms, in total variation at the 750^{th} time-step, the DSMC algorithm provides approximately 1.26 and 60.67 times improvement in the speed of convergence (the ratios of the total amount of changing of the total variations in the given time-step) in the first case and 1.51, and 1.25 times improvement in the second case. The PSG-IMC algorithm does not perform as well in the first case because of the issues caused by having a sparse adjacency matrix as discussed in related work. In the second case, which has a much denser adjacency matrix, the PSG-IMC algorithm provides approximately 1.21 times improvement in total variation compared to the M-H algorithm but the DSMC algorithm still provides faster convergence than the PSG-IMC algorithm. Furthermore, the total variation value of the DSMC algorithm converges to a value below the maximum value of the quantization error provided by Theorem 6.21. In the DSMC algorithm, as the distribution converges, the Markov matrix turns into an identity matrix, and unnecessary transitions are avoided as in the PSG-IMC algorithm. Since the transition of agents rarely occurred in the PSG-IMC algorithm in the first case due to the sparse adjacency matrix, the total number of transitions of the DSMC algorithm is approximately 49.56 times less than the M-H algorithm, and 4.93 times more than the PSG-IMC algorithm. For the second case, the total number of transitions of the DSMC algorithm is approximately 74.96 times less than the M-H algorithm, and 1.17 times less than the PSG-IMC algorithm.

In the second simulation, a more comprehensive comparison is provided. The swarm distribution converges to various multimodal Gaussian distributions (i.e., a mixture of multiple Gaussian distributions). The desired distribution is changed to another multimodal Gaussian distribution every 40 time-steps, except for the time-steps between 160 and 240. Up to the 160^{th} time-step, agents converge to the 4 different multimodal Gaussian distributions. To show the self-repair property of the swarm, approximately 1/3 of the agents are removed at the 161^{st} time-step and the remaining agents converge to the same desired distribution in 40 time-steps. Moreover, 7500 agents are uniformly added to the operational region at 201^{st} time-step and all agents converge to the same desired distribution again in 40 time-steps. After the 240^{th} time-step, agents converge to the 3 new multimodal Gaussian distributions up to the 360^{th} time-step. Comparisons of the total variation and the total number of transitions for the M-H, PSG-IMC, and DSMC algorithms are given in Figure 6.6 and Table 6.3. In this case, the adjacency matrix allows transitions to 10-step away bins. When compared to the M-H and PSG-IMC algorithms, the DSMC algorithm provides a significant improvement in the speed of convergence for each desired multimodal Gaussian distribution, as well as causing fewer transitions compared to both algorithms. Similar to the first simulation, the total variation value of the DSMC algorithm reaches the maximum value of the quantization error given in Theorem 6.21. Furthermore, when some agents are removed or new agents are added to the operational region, the DSMC algorithm recovers the desired distribution faster than the previous algorithms.

Table 6.1: Comparison of change of the total variation and the total number of transitions with time for the algorithms. In this case, the adjacency matrix only allows agents to transition to 1-step away bins.

	Total Variations				Ratios of change of Total Variation			Total Number of Transitions	
Time	0	250	251	750	0-250	250-750	0-750	750	750 *(Ratios)*
M-H Algorithm	0.268	0.107	0.273	0.123	1.27	1.16	**1.26**	2282394	**49.56**
PSG-IMC Algorithm	0.260	0.169	0.282	0.257	2.25	6.96	**60.67**	9348	**1/(4.93)**
DSMC Algorithm	0.261	0.056	0.253	0.079	1.00	1.00	**1.00**	46051	**1.00**



(a) $t = 0$ (b) $t = 250$ (c) $t = 251$ (d) $t = 750$

Figure 6.2: Representation of the distribution of the swarm for the time-steps 0, 250, 251, and 750, respectively. There are 400 (20×20) bins and 5000 agents in the operational region at the beginning of the simulation. The agents converge to the "E" letter in 250 time-steps and approximately 1/3 agents are removed from the operational space. Then, the remaining agents converge to the desired distribution again in 500 time-steps.

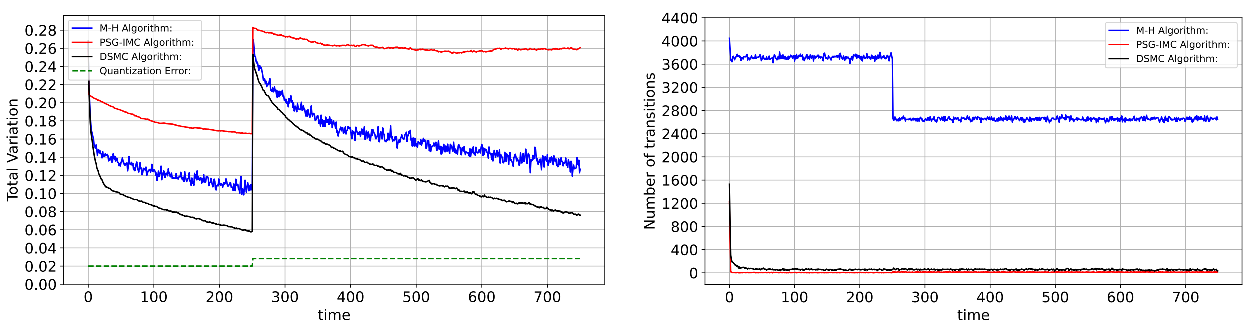


Figure 6.3: Comparison of change of the total variation and the number of transitions with time for the algorithms. In this case, the adjacency matrix only allows agents to transition to 1-step away bins.

Algorithm 3 Decentralized state-dependent Markov chain synthesis algorithm

Input: $x[l](k), v[l]$, $\forall l \in I_{ij}$, $A_a[\ell, \cdot]$, $\forall \ell \in \{i, j\}$, $c_2 \in (0, 1)$

Output: $M[i, j](k)$

1: Compute error values at the states associated with set I_{ij}:

$e[l](k) := x[l](k) - v[l]$, $\forall l = I_{ij}$

2: Calculate the non-negative error differences between adjacent states:

$E[i, j](k) := \max\Big(0, \Big(e[j](k) - e[i](k)\Big)A_a[i, j]\Big)$

3: Determine the total number of adjacent states:

$d[\ell] := \sum_{l \in \mathcal{I}_1^m \setminus \ell} A_a[\ell, l]$, $\forall \ell \in \{i, j\}$

4: Calculate the scaling factor to ensure algorithm stability:

$s_1[i, j] := \frac{1-c_2}{\max(d[i], d[j])}$

5: Set the scaling factor based on the probability of state

$$\mathrm{v}_j\text{: } s_2[j](k) := \begin{cases} \frac{x[j](k)}{\sum_{l \in \mathcal{I}_1^m} E[l,j](k)} & \text{if } \sum_{l \in \mathcal{I}_1^m} E[l, j](k) \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

6: Determine the scaling factor to satisfy Condition 6.4(b):

$$s_3[i, j](k) := \begin{cases} 0 & \text{if } e[\ell] < 0 \text{ for all } \ell \in \{i, j\} \text{ and } e[l](k) \geq 0 \text{ for some } l \in I_{ij} \\ 1 & \text{otherwise} \end{cases}$$

7: Obtain the resulting scaling factor:

$W[i, j](k) := \min\Big(s_1[i, j], s_2[j](k), s_3[i, j](k)\Big)$

8: Calculate the scaled non-negative error differences between adjacent states:

$T[i, j](k) := E[i, j](k)W[i, j](k)$

9: Compute the transition probability from v_j to v_i:

$$F[i, j](k) := \begin{cases} \frac{T[i,j](k)}{x[j](k)} & \text{if } x[j](k) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

10: Synthesize the Markov matrix:

$$M[i, j](k) := \begin{cases} F[i, j](k) & \text{if } i \neq j \\ 1 - \sum_{l \in \mathcal{I}_1^m} F[l, j](k) & \text{if } i = j \end{cases}$$

Algorithm 4 Probabilistic Guidance Algorithm [3]

1: Each agent determines its current bin, e.g., $r_l(k) \in R_i$.

2: Each agent generates a random number z that is uniformly distributed in $[0, 1]$.

3: Each agent goes to bin j, *i.e.*, $r_l(k+1) \in R_j$, if $\left\{ \sum_{s=1}^{j-1} M[s,i](k) \leq z \leq \sum_{s=1}^{j} M[s,i](k). \right.$

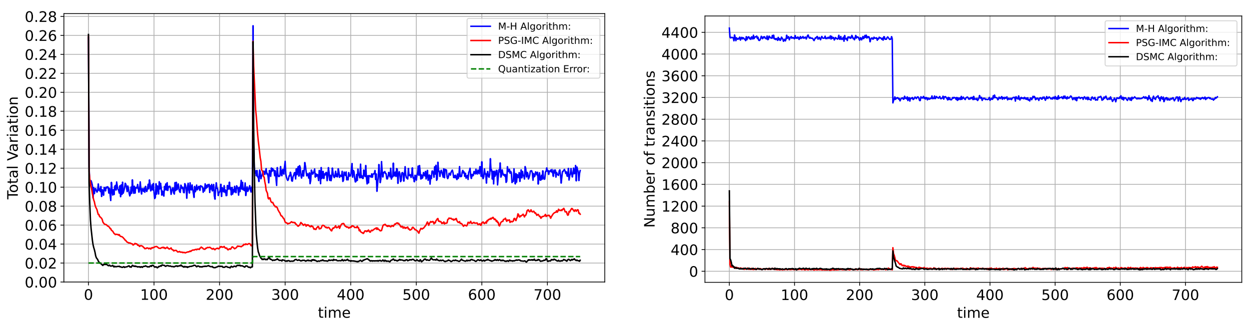
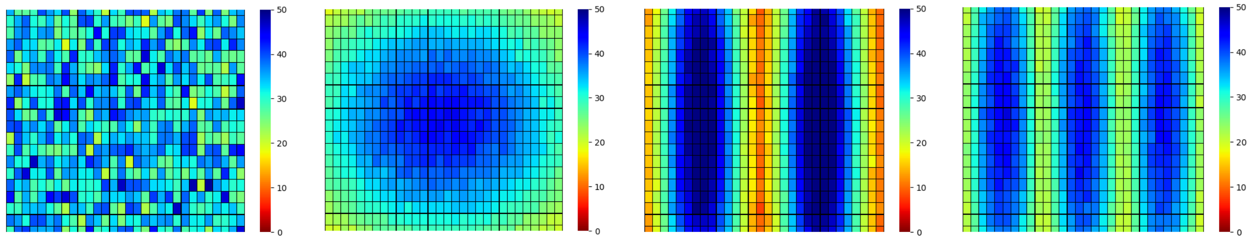


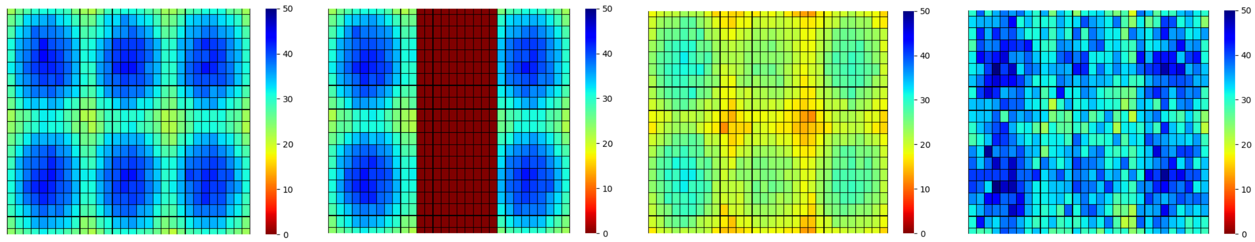
Figure 6.4: Comparison of change of the total variation and the number of transitions with time for the algorithms. In this case, the adjacency matrix allows agents to transition to 10-step away bins.

Table 6.2: Comparison of change of the total variation and the total number of transitions with time for the algorithms. In this case, the adjacency matrix allows agents to transition to 10-step away bins.

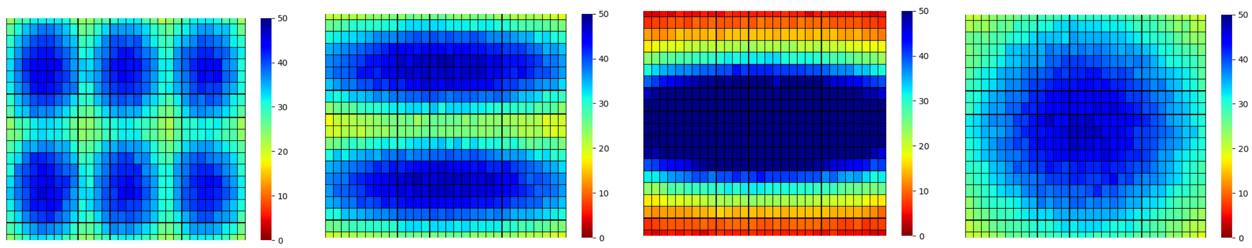
	Total Variations				Ratios of change of Total Variation			Total Number of Transitions	
Time	0	250	251	750	0-250	250-750	0-750	750	750 *(Ratios)*
M-H Algorithm	0.271	0.101	0.269	0.113	1.44	1.48	**1.51**	2663105	**74.96**
PSG-IMC Algorithm	0.263	0.039	0.253	0.072	1.09	1.28	**1.25**	41483	**1.17**
DSMC Algorithm	0.261	0.017	0.253	0.022	1.00	1.00	**1.00**	35529	**1.00**



(a) $t = 0$ (b) $t = 40$ (c) $t = 80$ (d) $t = 120$



(e) $t = 160$ (f) $t = 161$ (g) $t = 200$ (h) $t = 201$



(i) $t = 240$ (j) $t = 280$ (k) $t = 320$ (l) $t = 360$

Figure 6.5: Representation of the distribution of the swarm for several time-steps. There are 600 (20×30) bins and 20000 agents in the operational region at the beginning of the simulation. Swarm distribution converges to a different multimodal Gaussian distribution in every 40 time-step except the time-steps between 160 and 240. Agents converge to the 4 different multimodal Gaussian distributions up to the 160^{th} time-step. At the 161^{st} time-step, approximately 1/3 agents are removed and the remaining agents converge to the same desired distribution in 40 time-steps. Then, 7500 agents are uniformly added to the operational region at 201^{st} time-step and all agents converge to the same desired distribution again in 40 time-steps. After the 240^{th} time-step, agents converge to the 3 new multimodal Gaussian distributions up to the 360^{th} time-step.

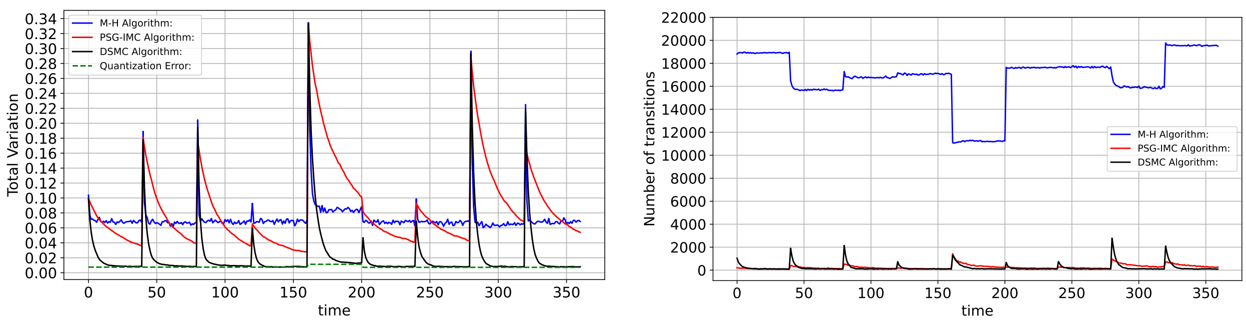


Figure 6.6: Comparison of change of the total variation and the number of transitions with time for the algorithms.

Table 6.3: Comparison of change of the total variation and the total number of transitions with time for the algorithms.

	Ratios of change of Total Variation									Total Number of Transitions	
Time	0-40	40-80	80-120	120-160	161-200	201-240	240-280	280-320	320-360	360	360 *(Ratios)*
M-H Algorithm	2.57	1.47	1.34	2.00	1.29	7.06	1.77	1.24	1.34	6058502	77.09
PSG-IMC Algorithm	1.37	1.24	1.36	1.36	1.38	0.94	1.26	1.25	1.89	95277	1.21
DSMC Algorithm	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	78592	1.00

Chapter 7

CONCLUSIONS AND FUTURE DIRECTIONS

7.1 Conclusions

This dissertation first developed smooth and exact modeling tools for mission-level specifications in trajectory optimization. Building on generalized mean-based smooth robustness, it introduced $\mathcal{C}^1$-smooth parameterizations of signal temporal logic requirements that preserve the satisfaction logic of the original specifications. These constructions enable continuous-time satisfaction of `always`, `eventually`, and `until`-type requirements while mitigating the locality and masking behavior of standard quantitative semantics.

The dissertation then developed prox-convex, a structure-preserving sequential convex programming algorithm for the nonconvex composite problems induced by these modeling choices and exact-penalty optimal-control transcriptions. The method preserves existing convex objectives, constraints, and inner convex structure while locally approximating only the smooth nonconvex components, yielding a numerical framework well matched to the resulting trajectory-generation problems.

These modeling and algorithmic ingredients were integrated in several applications, including illustrative continuous-time STL examples, nonlinear model predictive control with continuous-time constraint satisfaction, perception-constrained motion planning for information acquisition, rocket landing with compound state-triggered constraints, and nonlinear quadrotor flight with moving predicates and terminal requirements.

As a complementary contribution, the dissertation also developed a decentralized state-dependent Markov-chain synthesis framework for swarm guidance. A state-dependent consensus protocol was introduced on dynamic graphs and used to construct a decentralized Markov-chain synthesis algorithm with exponential convergence guarantees.

Overall, the dissertation shows that reliable autonomous guidance benefits from the joint design of expressive models and structure-preserving algorithms, both for individual trajectory generation and for large-scale swarm guidance.

7.2 Future Directions

Several future directions emerge naturally from this dissertation.

A first direction concerns *solver-aware problem formulation.* One recurring lesson throughout the dissertation is that the numerical behavior of an optimization problem depends strongly on how it is modeled. Two mathematically equivalent formulations of the same task can lead to very different solver performance because of scaling, conditioning, gradient distribution, active-set structure, and the way convexity is exposed. This suggests a broader research direction in which modeling itself becomes an algorithmic object. Instead of asking only how to solve a given formulation efficiently, one asks how to transform a mission description into a formulation that is both semantically faithful and well matched to the numerical solver that will be used.

This perspective opens the door to model-analysis tools for trajectory optimization and mission-level specifications. Such tools could take a user-specified optimal-control problem, temporal-logic requirement, or composite objective; analyze its scaling, smoothness, convex structure, gradient pathways, constraint activity, and constraint interactions; and then provide feedback on how to reformulate the problem before solving it. In this way, modeling could become a feedback-guided design process, analogous in spirit to how control engineers use analysis tools to tune and diagnose feedback systems.

A second direction concerns *structure-detecting algorithms and solvers for nonconvex optimization.* The prox-convex framework of Chapter 4 shows that preserving convex objectives, convex constraints, and convex inner features can substantially change the optimization problem seen by the solver. A natural next step is to develop parsers and solver stacks that automatically detect such structure from a user-provided model, decide which components should be preserved and which should be linearized, and generate the corresponding convex subproblems without requiring the user to manually expose the decomposition. This would move structure exploitation from a hand-designed modeling choice to an automated part of the trajectory-optimization pipeline. A closely related avenue is solver specialization and code generation for embedded implementations, where the detected composite structure, sparsity, scaling, and convex subproblem form are used to produce real-time solvers automatically.

A third direction is *closed-loop trajectory generation with rich mission specifications*. A natural next step is to extend the methods of Chapter 5 to problems in which task requirements evolve online or depend explicitly on uncertainty, sensing, or environment updates. Examples include nonlinear model predictive control with richer temporal operators beyond `always`-type safety constraints, perception-aware planning with coupled estimation and control objectives, and multi-agent trajectory generation with interacting mission-level requirements. In such settings, the challenge is not merely to solve a sequence of nonconvex problems online, but to do so while preserving the semantic guarantees that motivate the use of temporal-logic specifications in the first place.

Finally, the DSMC chapter opens several avenues for *scalable distributed guidance*. One natural extension is to move beyond finite-state distributions toward continuous-state or hybrid swarm-distribution models while preserving the decentralized and state-dependent character of the synthesis. Another is to incorporate richer communication effects, such as asynchronous updates, delays, packet loss, or imperfect local state estimates, into the convergence analysis. It is also of interest to integrate the distribution-level Markov guidance with local trajectory-generation or collision-avoidance layers, so that one can reason jointly about population shaping and individual agent motion. More broadly, the synthesis of efficient decentralized guidance laws that retain rigorous convergence guarantees remains a central challenge for large-scale autonomy.

In summary, this dissertation suggests that progress in autonomous guidance depends not only on faster solvers or richer task descriptions, but on the joint design of modeling languages, numerical formulations, and algorithms. The hope is that the frameworks developed here provide useful building blocks toward such integrated modeling-and-solution pipelines.

BIBLIOGRAPHY

[1] OpenSCVX: Fast and easy nonconvex trajectory optimization, 2026. URL https://openscvx.github.io/OpenSCvx/latest/. Software documentation.

[2] Behçet Açıkmeşe, John M. Carson III, and Lars Blackmore. Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. *IEEE Transactions on Control Systems Technology*, 21(6):2104–2113, nov 2013. doi: 10.1109/tcst.2012.2237346. URL https://doi.org/10.1109/TCST.2012.2237346.

[3] Behçet Açıkmeşe and David S Bayard. A markov chain approach to probabilistic swarm guidance. In *2012 American Control Conference (ACC)*, pages 6300–6307. IEEE, 2012. doi: 10.1109/acc.2012.6314729. URL https://doi.org/10.1109/ACC.2012.6314729.

[4] Behçet Açıkmeşe and David S Bayard. Markov chain approach to probabilistic guidance for swarms of autonomous agents. *Asian Journal of Control*, 17(4):1105–1124, 2015. doi: 10.1002/asjc.982. URL https://doi.org/10.1002/asjc.982.

[5] Behçet Açıkmeşe and Lars Blackmore. Lossless convexification of a class of optimal control problems with non-convex control constraints. *Automatica*, 47(2):341–347, 2011. doi: 10.1016/j.automatica.2010.10.037. URL https://doi.org/10.1016/j.automatica.2010.10.037.

[6] Behçet Açıkmeşe and Scott R Ploen. Convex programming approach to powered descent guidance for mars landing. *Journal of Guidance, Control, and Dynamics*, 30(5):1353–1366, 2007. doi: 10.2514/1.27553. URL https://doi.org/10.2514/1.27553.

[7] Behçet Açıkmeşe, M Aung, Jordi Casoliva, Swati Mohan, Andrew Johnson, Daniel

Scharf, David Masten, Joel Scotkin, Aron Wolf, and Martin W Regehr. Flight testing of trajectories computed by g-fold: Fuel optimal large divert guidance algorithm for planetary landing. In *AAS/AIAA Spaceflight Mechanics Meeting*, page 386, 2013. URL https://www.researchgate.net/publication/236334043.

[8] Behçet Açıkmeşe, Nazlı Demir, and Matthew W Harris. Convex necessary and sufficient conditions for density safety constraints in markov chain synthesis. *IEEE Transactions on Automatic Control*, 60(10):2813–2818, 2015. doi: 10.1109/TAC.2015.2400712. URL https://doi.org/10.1109/TAC.2015.2400712.

[9] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 10 2023. URL https://github.com/ceres-solver/ceres-solver.

[10] Takumi Akazaki and Ichiro Hasuo. Time robustness in MTL and expressivity in hybrid system falsification. In *International Conference on Computer Aided Verification*, pages 356–374. Springer, 2015. doi: 10.1007/978-3-319-21668-3_21. URL https://doi.org/10.1007/978-3-319-21668-3_21.

[11] Derya Aksaray, Austin Jones, Zhaodan Kong, Mac Schwager, and Calin Belta. Q-learning for robust satisfaction of signal temporal logic specifications. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6565–6570. IEEE, 2016. doi: 10.1109/cdc.2016.7799279. URL https://doi.org/10.1109/CDC.2016.7799279.

[12] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11797. URL https://doi.org/10.1609/AAAI.V32I1.11797.

[13] Rajeev Alur and David L Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994. doi: 10.1016/0304-3975(94)90010-8. URL https://doi.org/10.1016/0304-3975(94)90010-8.

[14] Rajeev Alur, Tomás Feder, and Thomas A Henzinger. The benefits of relaxing punctuality. *Journal of the ACM (JACM)*, 43(1):116–146, 1996. doi: 10.1145/227595.227602. URL https://doi.org/10.1145/227595.227602.

[15] Aleksandr Y Aravkin, James V Burke, and Gianluigi Pillonetto. Sparse/robust estimation and kalman smoothing with nonsmooth log-concave densities: Modeling, computation, and theory. *The Journal of Machine Learning Research*, 14(1):2689–2728, 2013. doi: 10.5555/2567709.2567747. URL http://jmlr.org/papers/v14/aravkin13a.html.

[16] Aleksandr Y Aravkin, Robert Baraldi, and Dominique Orban. A proximal quasi-newton trust-region method for nonsmooth regularized optimization. *SIAM Journal on Optimization*, 32(2):900–929, 2022. doi: 10.1137/21m1409536. URL https://doi.org/10.1137/21M1409536.

[17] Aleksandr Y Aravkin, Robert Baraldi, and Dominique Orban. A levenberg–marquardt method for nonsmooth regularized least squares. *SIAM Journal on Scientific Computing*, 46(4):A2557–A2581, 2024. doi: 10.1137/22m1538971. URL https://doi.org/10.1137/22M1538971.

[18] Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical programming*, 137(1):91–129, 2013. doi: 10.1007/s10107-011-0484-9. URL https://doi.org/10.1007/s10107-011-0484-9.

[19] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT press, 2008. doi: 10.5555/1373322. URL https://mitpress.mit.edu/9780262026499/principles-of-model-checking/.

[20] Saptarshi Bandyopadhyay, Soon-Jo Chung, and Fred Y Hadaegh. Probabilistic swarm guidance using optimal transport. In *2014 IEEE Conference on Control Applications*

(CCA), pages 498–505. IEEE, 2014. doi: 10.1109/CCA.2014.6981395. URL https://doi.org/10.1109/CCA.2014.6981395.

[21] Saptarshi Bandyopadhyay, Soon-Jo Chung, and Fred Y Hadaegh. Probabilistic and distributed control of a large-scale swarm of autonomous agents. *IEEE Transactions on Robotics*, 33(5):1103–1123, 2017. doi: 10.1109/TRO.2017.2705044. URL https://doi.org/10.1109/TRO.2017.2705044.

[22] Agamirza E Bashirov, Emine Mısırlı Kurpınar, and Ali Özyapıcı. Multiplicative calculus and its applications. *Journal of mathematical analysis and applications*, 337(1):36–48, 2008. doi: 10.1016/j.jmaa.2007.03.081. URL https://doi.org/10.1016/j.jmaa.2007.03.081.

[23] Barış Başpınar, Hamsa Balakrishnan, and Emre Koyuncu. Mission planning and control of multi-aircraft systems with signal temporal logic specifications. *IEEE Access*, 7:155941–155950, 2019. doi: 10.1109/access.2019.2949426. URL https://doi.org/10.1109/ACCESS.2019.2949426.

[24] Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer, 2nd edition, 2017. doi: 10.1007/978-3-319-48311-5. URL https://doi.org/10.1007/978-3-319-48311-5.

[25] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009. doi: 10.1137/080716542. URL https://doi.org/10.1137/080716542.

[26] Howard E Bell. Gershgorin's theorem and the zeros of polynomials. *The American Mathematical Monthly*, 72(3):292–295, 1965. doi: 10.2307/2313703. URL https://doi.org/10.2307/2313703.

[27] Calin Belta and Sadra Sadraddini. Formal methods for control synthesis: An optimization perspective. *Annual Review of Control, Robotics, and Autonomous Sys-*

tems, 2:115–140, 2019. doi: 10.1146/annurev-control-053018-023717. URL https://doi.org/10.1146/annurev-control-053018-023717.

[28] Dimitri Bertsekas and John N Tsitsiklis. *Introduction to probability*, volume 1. Athena Scientific, 2008.

[29] John T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998. doi: 10.2514/2.4231. URL https://doi.org/10.2514/2.4231.

[30] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Advances in Design and Control. SIAM, 2nd edition, jan 2010. doi: 10.1137/1.9780898718577. URL https://doi.org/10.1137/1.9780898718577.

[31] Shiba Biswal, Karthik Elamvazhuthi, and Spring Berman. Decentralized control of multiagent systems using local density feedback. *IEEE Transactions on Automatic Control*, 67(8):3920–3932, 2021. doi: 10.1109/TAC.2021.3109520. URL https://doi.org/10.1109/TAC.2021.3109520.

[32] Åke Björck. *Numerical methods for least squares problems*. SIAM, 1996. doi: 10.1137/1.9781611971484. URL https://doi.org/10.1137/1.9781611971484.

[33] Lars Blackmore. Autonomous precision landing of space rockets. In *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*, volume 46, pages 15–20. The Bridge Washington, DC, 2016. doi: 10.17226/23659. URL https://www.researchgate.net/publication/316547389.

[34] Lars Blackmore, Behçet Açıkmeşe, and Daniel P Scharf. Minimum-landing-error powered-descent guidance for mars landing using convex optimization. *Journal of Guidance, Control, and Dynamics*, 33(4):1161–1171, 2010. doi: 10.2514/1.47202. URL https://doi.org/10.2514/1.47202.

[35] Lars Blackmore, Behçet Açıkmeşe, and John M Carson III. Lossless convexification of control constraints for a class of nonlinear optimal control problems. *Systems & Control Letters*, 61(8):863–870, 2012. doi: 10.1016/j.sysconle.2012.04.010. URL https://doi.org/10.1016/j.sysconle.2012.04.010.

[36] H G Bock and K J Plitt. A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2):1603–1608, jul 1984. ISSN 1474-6670. doi: 10.1016/s1474-6670(17)61205-9. URL https://doi.org/10.1016/s1474-6670(17)61205-9. 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984.

[37] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146 (1):459–494, 2014. doi: 10.1007/s10107-013-0701-9. URL https://doi.org/10.1007/s10107-013-0701-9.

[38] Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507, 2017. doi: 10.1007/s10107-016-1091-6. URL https://doi.org/10.1007/s10107-016-1091-6.

[39] Giuseppe Bombara, Cristian-Ioan Vasile, Francisco Penedo, Hirotoshi Yasuoka, and Calin Belta. A decision tree approach to data classification using signal temporal logic. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pages 1–10, 2016. doi: 10.1145/2883817.2883843. URL https://doi.org/10.1145/2883817.2883843.

[40] Riccardo Bonalli, Abhishek Cauligi, Andrew Bylard, and Marco Pavone. GuSTO: Guaranteed sequential trajectory optimization via sequential convex programming. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6741–6747.

IEEE, may 2019. doi: 10.1109/icra.2019.8794205. URL https://doi.org/10.1109/ICRA.2019.8794205.

[41] Angelo D Bonzanini, Ali Mesbah, and Stefano Di Cairano. Perception-aware model predictive control for constrained control in unknown environments. *Automatica*, 160: 111418, 2024. doi: 10.1016/j.automatica.2023.111418. URL https://doi.org/10.1016/j.automatica.2023.111418.

[42] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, mar 2004. doi: 10.1017/cbo9780511804441. URL https://doi.org/10.1017/CBO9780511804441.

[43] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. Siam, 1994. doi: 10.1137/1.9781611970777. URL https://doi.org/10.1137/1.9781611970777.

[44] Stephen Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing markov chain on a graph. *SIAM review*, 46(4):667–689, 2004. doi: 10.1137/S0036144503423264. URL https://doi.org/10.1137/S0036144503423264.

[45] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006. doi: 10.1109/TIT.2006.874516. URL https://doi.org/10.1109/TIT.2006.874516.

[46] Stephen Boyd, Persi Diaconis, Pablo Parrilo, and Lin Xiao. Fastest mixing markov chain on graphs with symmetries. *SIAM Journal on Optimization*, 20(2):792–819, 2009. doi: 10.1137/070689413. URL https://doi.org/10.1137/070689413.

[47] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. doi: 10.1561/2200000016. URL https://doi.org/10.1561/2200000016.

[48] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. JAX: composable transformations of Python+ NumPy programs, 2018. URL https://github.com/jax-ml/jax.

[49] Samuel C. Buckner, Joshua Shaffer, John M. Carson, Breanna J. Johnson, Ronald R. Sostaric, and Behçet Açıkmeşe. Constrained visibility guidance for 6-dof powered descent maneuvers with terrain scanning using sequential convex programming. In *AIAA SCITECH 2024 Forum*, page 1759, 2024. doi: 10.2514/6.2024-1759. URL https://doi.org/10.2514/6.2024-1759.

[50] James V. Burke. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33(3):260–279, dec 1985. doi: 10.1007/bf01584377. URL https://doi.org/10.1007/BF01584377.

[51] James V Burke. Second order necessary and sufficient conditions for convex composite ndo. *Mathematical programming*, 38(3):287–302, 1987. doi: 10.1007/bf02592016. URL https://doi.org/10.1007/BF02592016.

[52] James V Burke and Abraham Engle. Line search and trust-region methods for convex-composite optimization. *arXiv preprint arXiv:1806.05218*, 2018. doi: 10.48550/arXiv.1806.05218. URL https://arxiv.org/abs/1806.05218.

[53] James V Burke and Abraham Engle. Strong metric (sub) regularity of KKT mappings for piecewise linear-quadratic convex-composite optimization. *Mathematics of Operations Research*, 45(3), 2020. doi: 10.1287/moor.2019.1027. URL https://doi.org/10.1287/moor.2019.1027.

[54] James V Burke and Michael C Ferris. A Gauss-Newton method for convex composite optimization. *Mathematical Programming*, 71(2):179–194, 1995. doi: 10.1007/bf01585997. URL https://doi.org/10.1007/BF01585997.

[55] Ali Tevfik Buyukkocak, Derya Aksaray, and Yasin Yazıcıoğlu. Planning of heterogeneous multi-agent systems under signal temporal logic specifications with integral predicates. *IEEE Robotics and Automation Letters*, 6(2):1375–1382, 2021. doi: 10.1109/lra.2021.3057049. URL https://doi.org/10.1109/LRA.2021.3057049.

[56] Ali Tevfik Buyukkocak, Derya Aksaray, and Yasin Yazıcıoğlu. Control barrier functions with actuation constraints under signal temporal logic specifications. In *2022 European Control Conference (ECC)*, pages 162–168, 2022. doi: 10.23919/ecc55457.2022.9838028. URL https://doi.org/10.23919/ecc55457.2022.9838028.

[57] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1):427–438, 2012. doi: 10.1109/TII.2012.2219061. URL https://doi.org/10.1109/TII.2012.2219061.

[58] Gustavo A Cardona and Cristian-Ioan Vasile. Planning for heterogeneous teams of robots with temporal logic, capability, and resource constraints. *The International Journal of Robotics Research*, page 02783649241247285, 2024. doi: 10.1177/02783649241247285. URL https://doi.org/10.1177/02783649241247285.

[59] Gustavo A Cardona, Disha Kamale, and Cristian-Ioan Vasile. Mixed integer linear programming approach for control synthesis with weighted signal temporal logic. In *Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control*, pages 1–12, 2023. doi: 10.1145/3575870.3587120. URL https://doi.org/10.1145/3575870.3587120.

[60] John M Carson, Michelle M Munk, Ronald R Sostaric, Jay N Estes, Farzin Amzajerdian, James B Blair, David K Rutishauser, Carolina I Restrepo, Alicia M Dwyer-Cianciolo, George Chen, et al. The SPLICE project: Continuing NASA development of GN&C technologies for safe and precise landing. In *AIAA SciTech 2019 forum*, page 0660, 2019. doi: 10.2514/6.2019-0660. URL https://doi.org/10.2514/6.2019-0660.

[61] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. *SIAM Journal on Optimization*, 21(4):1721–1739, oct 2011. ISSN 1052-6234. doi: 10.1137/11082381x. URL https://doi.org/10.1137/11082381X.

[62] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1–2):89–97, 2004. doi: 10.1023/b:jmiv.0000011325.36760.1e. URL https://doi.org/10.1023/B:JMIV.0000011325.36760.1e.

[63] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011. doi: 10.1007/s10851-010-0251-1. URL https://doi.org/10.1007/s10851-010-0251-1.

[64] Govind M. Chari and Behçet Açıkmeşe. Spacecraft rendezvous guidance via factorization-free sequential convex programming using a first-order method. *arXiv preprint arXiv:2402.04561*, 2024. doi: 10.48550/arXiv.2402.04561. URL https://doi.org/10.48550/arXiv.2402.04561.

[65] Govind M Chari and Behçet Açıkmeşe. QOCO: A quadratic objective conic optimizer with custom solver generation. *arXiv preprint arXiv:2503.12658*, 2025. doi: 10.48550/arxiv.2503.12658. URL https://doi.org/10.48550/arXiv.2503.12658.

[66] Govind M. Chari, Abhinav G. Kamath, Purnanand Elango, and Behçet Açıkmeşe. Fast monte carlo analysis for 6-DoF powered-descent guidance via GPU-accelerated sequential convex programming. In *AIAA SciTech 2024 Forum*, page 1762. AIAA, jan 2024. doi: 10.2514/6.2024-1762. URL https://doi.org/10.2514/6.2024-1762.

[67] Vasileios Charisopoulos, Damek Davis, Matías Díaz, and Dmitriy Drusvyatskiy. Com-

posite optimization for robust blind deconvolution. *arXiv preprint arXiv:1901.01624*, 2019. doi: 10.48550/arXiv.1901.01624. URL https://arxiv.org/abs/1901.01624.

[68] Maria Charitidou and Dimos V Dimarogonas. Barrier function-based model predictive control under signal temporal logic specifications. In *2021 European Control Conference (ECC)*, pages 734–739. IEEE, 2021. doi: 10.23919/ecc54610.2021.9655231. URL https://doi.org/10.23919/ecc54610.2021.9655231.

[69] Soon-Jo Chung, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4): 837–855, 2018. doi: 10.1109/TRO.2018.2857475. URL https://doi.org/10.1109/TRO.2018.2857475.

[70] Radek Cibulka, AL Dontchev, and AY Kruger. Strong metric subregularity of mappings in variational analysis and optimization. *Journal of Mathematical Analysis and Applications*, 457(2):1247–1282, 2018. doi: 10.1016/j.jmaa.2016.11.045. URL https://doi.org/10.1016/j.jmaa.2016.11.045.

[71] Frank H. Clarke. *Optimization and Nonsmooth Analysis*. Classics in Applied Mathematics. John Wiley & Sons, first edition, 1983. doi: 10.1137/1.9781611971309. URL https://doi.org/10.1137/1.9781611971309.

[72] Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011. doi: 10.1007/978-1-4419-9569-8_10. URL https://doi.org/10.1007/978-1-4419-9569-8_10.

[73] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000. doi: 10.1137/1.9780898719857. URL https://doi.org/10.1137/1.9780898719857.

[74] George Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal of parallel and distributed computing*, 7(2):279–301, 1989. doi: 10.1016/0743-7315(89)90021-X. URL https://doi.org/10.1016/0743-7315(89)90021-X.

[75] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004. doi: 10.1002/cpa.20042. URL https://doi.org/10.1002/cpa.20042.

[76] Nazlı Demir, Behçet Açıkmeşe, and Can Pehlivantürk. Density control for decentralized autonomous agents with conflict avoidance. *IFAC Proceedings Volumes*, 47(3):11715–11721, 2014. doi: 10.3182/20140824-6-ZA-1003.01720. URL https://doi.org/10.3182/20140824-6-ZA-1003.01720.

[77] Nazlı Demir, Utku Eren, and Behçet Açıkmeşe. Decentralized probabilistic density control of autonomous swarms with safety constraints. *Autonomous Robots*, 39(4):537–554, 2015. doi: 10.1007/s10514-015-9470-z. URL https://doi.org/10.1007/s10514-015-9470-z.

[78] Nazlı Demirer, Mahmoud El Chamie, and Behçet Açıkmeşe. Safe markov chains for on/off density control with observed transitions. *IEEE Transactions on Automatic Control*, 63(5):1442–1449, 2017. doi: 10.1109/TAC.2017.2755366. URL https://doi.org/10.1109/TAC.2017.2755366.

[79] Stefano Di Cairano, Daniele Bernardini, Alberto Bemporad, and Ilya V Kolmanovsky. Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management. *IEEE Transactions on Control Systems Technology*, 22(3):1018–1031, 2013. doi: 10.1109/tcst.2013.2272179. URL https://doi.org/10.1109/TCST.2013.2272179.

[80] G. Di Pillo and L. Grippo. Exact penalty functions in constrained optimization. *SIAM*

Journal on Control and Optimization, 27(6):1333–1360, nov 1989. ISSN 1095-7138. doi: 10.1137/0327068. URL http://dx.doi.org/10.1137/0327068.

[81] Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016. doi: 10.5555/2946645.3007036. URL https://www.cvxpy.org/.

[82] Moritz Diehl, Hans Georg Bock, and Johannes P Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736, 2005. doi: 10.1137/s0363012902400713. URL https://doi.org/10.1137/S0363012902400713.

[83] Quoc Tran Dinh, Carlo Savorgnan, and Moritz Diehl. Adjoint-based predictor-corrector sequential convex programming for parametric nonlinear optimization. *SIAM Journal on Optimization*, 22(4):1258–1284, 2012. doi: 10.1137/110844349. URL https://doi.org/10.1137/110844349.

[84] Franck Djeumou, Zhe Xu, and Ufuk Topcu. Probabilistic Swarm Guidance Subject to Graph Temporal Logic Specifications. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020. doi: 10.15607/rss.2020.xvi.058. URL https://doi.org/10.15607/rss.2020.xvi.058.

[85] Franck Djeumou, Zhe Xu, Murat Cubuktepe, and Ufuk Topcu. Probabilistic control of heterogeneous swarms subject to graph temporal logic specifications: A decentralized and scalable approach. *IEEE Transactions on Automatic Control*, 68(4):2245–2260, 2022. doi: 10.1109/tac.2022.3176797. URL https://doi.org/10.1109/TAC.2022.3176797.

[86] Nikita Doikov and Yurii Nesterov. High-order optimization methods for fully composite problems. *SIAM Journal on Optimization*, 32(3):2402–2427, 2022. doi: 10.1137/21m1410063. URL https://doi.org/10.1137/21M1410063.

[87] Javier A Doll, Samuel T Wagner, Matthew Fritz, Jiann-woei Jang, Jeanette M Harper, Kyle W Smith, Behçet Açıkmeşe, Samuel M Pedrotty, and Gavin F Mendeck. Performance analysis of a dual quaternion guidance algorithm applicable during lunar approach with a hazard avoidance maneuver. In *AIAA SciTech 2024 Forum*, page 1757, 2024. doi: 10.2514/6.2024-1757. URL https://doi.org/10.2514/6.2024-1757.

[88] Alexander Domahidi, Eric Chu, and Stephen Boyd. ECOS: An SOCP solver for embedded systems. In *2013 European control conference (ECC)*, pages 3071–3076. IEEE, jul 2013. doi: 10.23919/ecc.2013.6669541. URL https://doi.org/10.23919/ECC.2013.6669541.

[89] Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 92–106. Springer, 2010. doi: 10.1007/978-3-642-15297-9_9. URL https://doi.org/10.1007/978-3-642-15297-9_9.

[90] Dmitriy Drusvyatskiy. The proximal point method revisited. *arXiv preprint arXiv:1712.06038*, 2017. doi: 10.48550/arXiv.1712.06038. URL https://arxiv.org/abs/1712.06038.

[91] Dmitriy Drusvyatskiy and Adrian Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018. doi: 10.1287/moor.2017.0889. URL https://doi.org/10.1287/moor.2017.0889.

[92] Dmitriy Drusvyatskiy and Courtney Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178(1):503–558, 2019. doi: 10.1007/s10107-018-1311-3. URL https://doi.org/10.1007/s10107-018-1311-3.

[93] Dmitriy Drusvyatskiy, Adrian S. Lewis, and Alexander D. Ioffe. Nonsmooth optimiza-

tion using Taylor-like models: error bounds, convergence, and termination criteria. *Mathematical Programming*, 185:357–383, 2021. doi: 10.1007/s10107-019-01432-w. URL https://doi.org/10.1007/s10107-019-01432-w.

[94] John C Duchi and Feng Ruan. Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval. *Information and Inference: A Journal of the IMA*, 8(3):471–529, 2019. doi: 10.1093/imaiai/iay015. URL https://doi.org/10.1093/imaiai/iay015.

[95] Daniel Dueri, Jing Zhang, and Behçet Açıkmeşe. Automated custom code generation for embedded, real-time second order cone programming. *IFAC Proceedings Volumes*, 47(3):1605–1612, 2014. doi: 10.3182/20140824-6-za-1003.02736. URL https://doi.org/10.3182/20140824-6-ZA-1003.02736.

[96] Daniel Dueri, Behçet Açıkmeşe, Daniel P. Scharf, and Matthew W. Harris. Customized real-time interior-point methods for onboard powered-descent guidance. *Journal of Guidance, Control, and Dynamics*, 40(2):197–212, feb 2017. ISSN 1533-3884. doi: 10.2514/1.g001480. URL https://doi.org/10.2514/1.G001480.

[97] Matthew Dunbabin and Lino Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics & Automation Mag.*, 19(1):24–39, 2012. doi: 10.1109/MRA.2011.2181683. URL https://doi.org/10.1109/MRA.2011.2181683.

[98] Jonathan Eckstein and Dimitri P. Bertsekas. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1–3):293–318, 1992. doi: 10.1007/bf01581204. URL https://doi.org/10.1007/BF01581204.

[99] Ivar Ekeland. On the variational principle. *Journal of Mathematical Analysis and*

Applications, 47(2):324–353, 1974. doi: 10.1016/0022-247x(74)90025-0. URL https://doi.org/10.1016/0022-247X(74)90025-0.

[100] Mahmoud El Chamie and Behçet Açıkmeşe. Safe metropolis–hastings algorithm and its application to swarm control. *Systems & Control Letters*, 111:40–48, 2018. doi: 10.1016/j.sysconle.2017.10.006. URL https://doi.org/10.1016/j.sysconle.2017.10.006.

[101] Karthik Elamvazhuthi and Spring Berman. Mean-field models in swarm robotics: A survey. *Bioinspiration & Biomimetics*, 15(1):015001, 2019. doi: 10.1088/1748-3190/ab49a4. URL https://doi.org/10.1088/1748-3190/ab49a4.

[102] Karthik Elamvazhuthi, Zahi Kakish, Aniket Shirsat, and Spring Berman. Controllability and stabilization for herding a robotic swarm using a leader: A mean-field approach. *IEEE Transactions on Robotics*, 37(2):418–432, 2020. doi: 10.1109/TRO.2020.3031237. URL https://doi.org/10.1109/TRO.2020.3031237.

[103] Purnanand Elango, Abhinav Kamath, Yue Yu, John M. Carson III, and Behçet Açıkmeşe. A customized first-order solver for real-time powered-descent guidance. In *AIAA SciTech 2022 Forum*, page 0951. American Institute of Aeronautics and Astronautics, jan 2022. doi: 10.2514/6.2022-0951. URL https://doi.org/10.2514/6.2022-0951.

[104] Purnanand Elango, Dayou Luo, Abhinav G Kamath, Samet Uzun, Taewan Kim, and Behçet Açıkmeşe. Continuous-time successive convexification for constrained trajectory optimization. *Automatica*, 180:112464, 2025. doi: 10.1016/j.automatica.2025.112464. URL https://doi.org/10.1016/j.automatica.2025.112464.

[105] Purnanand Elango, Abraham P. Vinod, Stefano Di Cairano, and Avishai Weiss. Continuous-time successive convexification for passively-safe six-degree-of-freedom powered-descent guidance. In *AIAA SciTech 2025*, page 1894. American Institute

of Aeronautics and Astronautics, jan 2025. doi: 10.2514/6.2025-1894. URL https://doi.org/10.2514/6.2025-1894.

[106] Purnanand Elango, Abraham P Vinod, Kenji Kitamura, Behçet Açıkmeşe, Stefano Di Cairano, and Avishai Weiss. Continuous-time successive convexification for passively-safe spacecraft rendezvous on a near rectilinear halo orbit. In *AIAA SCITECH 2026 Forum*, page 2446, 2026. doi: 10.2514/6.2026-2446. URL https://doi.org/10.2514/6.2026-2446.

[107] Utku Eren and Behçet Açıkmeşe. Velocity field generation for density control of swarms using heat equation and smoothing kernels. *IFAC-PapersOnLine*, 50(1):9405–9411, 2017. doi: 10.1016/j.ifacol.2017.08.1454. URL https://doi.org/10.1016/j.ifacol.2017.08.1454.

[108] Utku Eren, Anna Prach, Başaran Bahadır Koçer, Saša V Raković, Erdal Kayacan, and Behçet Açıkmeşe. Model predictive control in aerospace systems: Current state and opportunities. *Journal of Guidance, Control, and Dynamics*, 40(7):1541–1566, 2017. doi: 10.2514/1.g002507. URL https://doi.org/10.2514/1.G002507.

[109] Georgios E Fainekos and George J Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42):4262–4291, 2009. doi: 10.1016/j.tcs.2009.06.021. URL https://doi.org/10.1016/j.tcs.2009.06.021.

[110] Davide Falanga, Philipp Foehn, Peng Lu, and Davide Scaramuzza. PAMPC: Perception-aware model predictive control for quadrotors. In *IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2018. doi: 10.1109/IROS.2018.8593739. URL https://doi.org/10.1109/IROS.2018.8593739.

[111] Robert Falck, Justin S. Gray, Kaushik Ponnapalli, and Ted Wright. Dymos: A python package for optimal control of multidisciplinary systems. *Journal of Open Source Software*, 6(59):2809, 2021. doi: 10.21105/joss.02809.

[112] J Alexander Fax and Richard M Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004. doi: 10.1109/TAC.2004.834433. URL https://doi.org/10.1109/TAC.2004.834433.

[113] George Fishman. *Monte Carlo: concepts, algorithms, and applications*. Springer Science & Business Media, 2013. doi: 10.1007/978-1-4757-2553-7. URL https://doi.org/10.1007/978-1-4757-2553-7.

[114] Francesco Flammini, Concetta Pragliola, and Giovanni Smarra. Railway infrastructure monitoring by drones. In *Int. Transp. Electr. Conf.*, 2016. doi: 10.1109/ESARS-ITEC.2016.7841398. URL https://doi.org/10.1109/ESARS-ITEC.2016.7841398.

[115] Roger Fletcher. A model algorithm for composite nondifferentiable optimization problems. In *Mathematical Programming Study*, pages 17:67–76, Berlin, Heidelberg, 1982. Springer. doi: 10.1007/bfb0120959. URL https://doi.org/10.1007/BFB0120959.

[116] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2000. doi: 10.1002/9781118723203. URL https://doi.org/10.1002/9781118723203.

[117] Fernando A. C. C. Fontes and Luis Tiago Paiva. Guaranteed constraint satisfaction in continuous-time control problems. *IEEE Control Systems Letters*, 3(1):13–18, jan 2019. ISSN 2475-1456. doi: 10.1109/lcsys.2018.2849853. URL http://dx.doi.org/10.1109/LCSYS.2018.2849853.

[118] Anders Forsgren, Philip E Gill, and Margaret H Wright. Interior methods for nonlinear optimization. *SIAM review*, 44(4):525–597, 2002. doi: 10.1137/s00361445024149. URL https://doi.org/10.1137/S00361445024149.

[119] Alex Zheng Frank Allgöwer. *Nonlinear Model Predictive Control*. Birkhäuser Basel, 2000. ISBN 9783034884075. doi: 10.1007/978-3-0348-8407-5. URL https://doi.org/10.1007/978-3-0348-8407-5.

[120] Gianluca Frison and Moritz Diehl. HPIPM: a high-performance quadratic programming framework for model predictive control. *IFAC-PapersOnLine*, 53(2):6563–6569, 2020. doi: 10.1016/j.ifacol.2020.12.073. URL https://doi.org/10.1016/J.IFACOL.2020.12.073.

[121] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995. doi: 10.1201/b14835. URL https://doi.org/10.1201/b14835.

[122] Philip E. Gill, Walter Murray, and Michael A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005. doi: 10.1137/S0036144504446096. URL https://doi.org/10.1137/S0036144504446096.

[123] 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, 2020. doi: 10.1109/lcsys.2020.3001875. URL https://doi.org/10.1109/LCSYS.2020.3001875.

[124] Paul J Goulart and Yuwen Chen. Clarabel: An interior-point solver for conic programs with quadratic objectives. *arXiv:2405.12762*, 2024. doi: 10.48550/arXiv.2405.12762. URL https://doi.org/10.48550/arXiv.2405.12762.

[125] Melissa Greeff, Timothy D Barfoot, and Angela P Schoellig. A perception-aware flatness-based model predictive controller for fast vision-based multirotor flight. *IFAC-PapersOnLine*, 53(2):9412–9419, 2020. doi: 10.1016/j.ifacol.2020.12.2411. URL https://doi.org/10.1016/j.ifacol.2020.12.2411.

[126] Sébastien Gros, Mario Zanon, Rien Quirynen, Alberto Bemporad, and Moritz Diehl. From linear to nonlinear MPC: bridging the gap via the real-time iteration. *International Journal of Control*, 93(1):62–80, 2020. doi: 10.1080/00207179.2016.1222553. URL https://doi.org/10.1080/00207179.2016.1222553.

[127] Felix Gruber and Matthias Althoff. Scalable robust output feedback MPC of linear sampled-data systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, dec 2021. doi: 10.1109/cdc45484.2021.9683384. URL http://dx.doi.org/10.1109/CDC45484.2021.9683384.

[128] Iman Haghighi, Noushin Mehdipour, Ezio Bartocci, and Calin Belta. Control from signal temporal logic specifications with smooth cumulative quantitative semantics. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4361–4366. IEEE, 2019. doi: 10.1109/cdc40024.2019.9029429. URL https://doi.org/10.1109/CDC40024.2019.9029429.

[129] S. P. Han and Olvi L. Mangasarian. Exact penalty functions in nonlinear programming. *Mathematical Programming*, 17:251–269, 1979. doi: 10.1007/bf01588250. URL https://doi.org/10.1007/bf01588250.

[130] Matthew W Harris and Behçet Açıkmeşe. Lossless convexification for a class of optimal control problems with quadratic state constraints. In *2013 American Control Conference*, pages 3415–3420. IEEE, 2013. doi: 10.1109/acc.2013.6580359. URL https://doi.org/10.1109/ACC.2013.6580359.

[131] Matthew W Harris and Behçet Açıkmeşe. Lossless convexification of non-convex optimal control problems for state constrained linear systems. *Automatica*, 50(9):2304–2311, 2014. doi: 10.1016/j.automatica.2014.06.008. URL https://doi.org/10.1016/j.automatica.2014.06.008.

[132] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970. doi: 10.1093/biomet/57.1.97. URL https://doi.org/10.1093/biomet/57.1.97.

[133] Michael A. Henson. Nonlinear model predictive control: current status and future directions. *Computers & Chemical Engineering*, 23(2):187–202, dec 1998. ISSN 0098-

1354. doi: 10.1016/s0098-1354(98)00260-9. URL http://dx.doi.org/10.1016/S0098-1354(98)00260-9.

[134] Roger A Horn and Charles R Johnson. *Topics in matrix analysis*. Cambridge university press, 1994. doi: 10.1017/CBO9780511840371. URL https://doi.org/10.1017/CBO9780511840371.

[135] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012. doi: 10.1017/CBO9781139020411. URL https://doi.org/10.1017/CBO9781139020411.

[136] Boris Houska, Hans Joachim Ferreau, and Moritz Diehl. ACADO toolkit–an open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312, may 2011. doi: 10.1002/oca.939. URL https://doi.org/10.1002/oca.939.

[137] Boris Houska, Hans Joachim Ferreau, and Moritz Diehl. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, 47(10):2279–2285, 2011. doi: 10.1016/j.automatica.2011.08.020. URL https://doi.org/10.1016/j.automatica.2011.08.020.

[138] Yaohua Hu, Chong Li, and Xiaoqi Yang. On convergence rates of linearized proximal algorithms for convex composite optimization with applications. *SIAM Journal on Optimization*, 26(2):1207–1235, 2016. doi: 10.1137/140993090. URL https://doi.org/10.1137/140993090.

[139] Brian E Jackson, Tarun Punnoose, Daniel Neamati, Kevin Tracy, Rianna Jitosho, and Zachary Manchester. ALTRO-C: A fast solver for conic model-predictive control. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7357–7364. IEEE, 2021. doi: 10.1109/icra48506.2021.9561438. URL https://doi.org/10.1109/ICRA48506.2021.9561438.

[140] Ali Jadbabaie, Jie Lin, and A Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003. doi: 10.1109/TAC.2003.812781. URL https://doi.org/10.1109/TAC.2003.812781.

[141] Inmo Jang, Hyo-Sang Shin, and Antonios Tsourdos. Local information-based control for probabilistic swarm distribution guidance. *Swarm Intelligence*, 12(4):327–359, 2018. doi: 10.1007/s11721-018-0160-2. URL https://doi.org/10.1007/s11721-018-0160-2.

[142] Ahmed T Kamal, Jay A Farrell, and Amit K Roy-Chowdhury. Information weighted consensus filters and their application in distributed camera networks. *IEEE Transactions on Automatic Control*, 58(12):3112–3125, 2013. doi: 10.1109/TAC.2013.2277621. URL https://doi.org/10.1109/TAC.2013.2277621.

[143] Abhinav G. Kamath, Purnanand Elango, Skye Mceowen, Yue Yu, John M. Carson III, Mehran Mesbahi, and Behçet Açıkmeşe. Customized real-time first-order methods for onboard dual quaternion-based 6-DoF powered-descent guidance. In *AIAA SciTech 2023 Forum*. American Institute of Aeronautics and Astronautics, jan 2023. doi: 10.2514/6.2023-2003. URL https://doi.org/10.2514/6.2023-2003.

[144] Abhinav G. Kamath, Purnanand Elango, Yue Yu, Skye Mceowen, Govind M. Chari, John M. Carson, III, and Behçet Açıkmeşe. Real-time sequential conic optimization for multi-phase rocket landing guidance. *IFAC-PapersOnLine*, 56(2):3118–3125, 2023. doi: 10.1016/j.ifacol.2023.10.1444. URL https://doi.org/10.1016/j.ifacol.2023.10.1444.

[145] Abhinav G. Kamath, Javier A. Doll, Purnanand Elango, Taewan Kim, Skye McEowen, Yue Yu, Taylor P. Reynolds, Gavin F. Mendeck, John M. Carson III, Mehran Mesbahi, and Behçet Açıkmeşe. Onboard dual quaternion guidance for rocket landing. *arXiv preprint arXiv:2508.10439*, 2025. doi: 10.48550/arXiv.2508.10439. URL https://doi.org/10.48550/arXiv.2508.10439.

[146] Abhinav G Kamath, Purnanand Elango, and Behçet Açıkmeşe. Optimal preconditioning for online quadratic cone programming. *IEEE Control Systems Letters*, 9:108–113, 2025. doi: 10.1109/lcsys.2025.3526134. URL https://doi.org/10.1109/LCSYS.2025.3526134.

[147] Abhinav G. Kamath, Samet Uzun, Govind M. Chari, Behçet Açıkmeşe, Purnanand Elango, and Michael Szmuk. SCvxGEN: Successive convexification with automatic custom code generation for fast and embedded general-purpose trajectory optimization, 2025. URL https://scvxgen.com. Software documentation.

[148] Ruya Karagulle, Nikos Aréchiga, Andrew Best, Jonathan DeCastro, and Necmiye Ozay. A safe preference learning approach for personalization with applications to autonomous vehicles. *IEEE Robotics and Automation Letters*, 9(5):4226–4233, 2024. doi: 10.1109/lra.2024.3375626. URL https://doi.org/10.1109/LRA.2024.3375626.

[149] Sertac Karaman, Ricardo G Sanfelice, and Emilio Frazzoli. Optimal control of mixed logical dynamical systems with linear temporal logic specifications. In *2008 47th IEEE Conference on Decision and Control*, pages 2117–2122. IEEE, 2008. doi: 10.1109/cdc.2008.4739370. URL https://doi.org/10.1109/CDC.2008.4739370.

[150] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing (STOC)*, pages 302–311, 1984. doi: 10.1145/800057.808695. URL https://doi.org/10.1145/800057.808695.

[151] Solmaz S Kia, Jorge Cortés, and Sonia Martínez. Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, 55:254–264, 2015. doi: 10.1016/j.automatica.2015.03.001. URL https://doi.org/10.1016/j.automatica.2015.03.001.

[152] Solmaz S Kia, Bryan Van Scoy, Jorge Cortes, Randy A Freeman, Kevin M Lynch, and

Sonia Martinez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72, 2019. doi: 10.1109/MCS.2019.2900783. URL https://doi.org/10.1109/MCS.2019.2900783.

[153] H Jin Kim, David H Shim, and Shankar Sastry. Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In *Proceedings of the 2002 American Control Conference*, volume 5, pages 3576–3581. IEEE, 2002. doi: 10.1109/acc.2002.1024483. URL https://doi.org/10.1109/ACC.2002.1024483.

[154] Taewan Kim, Abraham P Vinod, and Stefano Di Cairano. Decoupled trajectory planning for monitoring UAVs and UGV carrier by reachable sets. In *American Control Conf.*, pages 587–593, 2024. doi: 10.23919/ACC60939.2024.10644642. URL https://doi.org/10.23919/ACC60939.2024.10644642.

[155] Taewan Kim, Abhinav G Kamath, Niyousha Rahimi, Behçet Açıkmeşe, Mehran Mesbahi, and Jasper Corleis. Six-degree-of-freedom aircraft landing trajectory planning with runway alignment. *Journal of Guidance, Control, and Dynamics*, 48(10):2225–2242, 2025. doi: 10.2514/1.g008521. URL https://doi.org/10.2514/1.G008521.

[156] Yoonsoo Kim and Mehran Mesbahi. On maximizing the second smallest eigenvalue of a state-dependent graph laplacian. *IEEE Transactions on Automatic Control*, 51(1):116–120, 2006. doi: 10.1109/TAC.2005.861710. URL https://doi.org/10.1109/TAC.2005.861710.

[157] Yoonsoo Kim, Mehran Mesbahi, and Fred Y Hadaegh. Multiple-spacecraft reconfiguration through collision avoidance, bouncing, and stalemate. *Journal of Optimization Theory and Applications*, 122(2):323–343, 2004. doi: 10.1023/B:JOTA.0000042524.57088.8b. URL https://doi.org/10.1023/B:JOTA.0000042524.57088.8b.

[158] Allan R Klumpp. Apollo lunar descent guidance. *Automatica*, 10(2):133–146, 1974. doi:

10.1016/0005-1098(74)90019-3. URL https://doi.org/10.1016/0005-1098(74)90019-3.

[159] Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990. doi: 10.1007/bf01995674. URL https://doi.org/10.1007/BF01995674.

[160] Vishaal Krishnan and Sonia Martínez. Distributed optimal transport for the deployment of swarms. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4583–4588. IEEE, 2018. doi: 10.1109/CDC.2018.8619816. URL https://doi.org/10.1109/CDC.2018.8619816.

[161] Vishaal Krishnan and Sonia Martinez. Distributed control for spatial self-organization of multi-agent swarms. *SIAM Journal on Control and Optimization*, 56(5):3642–3667, 2018. doi: 10.1137/16M1080926. URL https://doi.org/10.1137/16M1080926.

[162] Vince Kurtz and Hai Lin. Trajectory optimization for high-dimensional nonlinear systems under STL specifications. *IEEE Control Systems Letters*, 5(4):1429–1434, 2020. doi: 10.1109/lcsys.2020.3038640. URL https://doi.org/10.1109/LCSYS.2020.3038640.

[163] Jason D. Lee, Yifan Sun, and Michael A. Saunders. Proximal newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1441, 2014. doi: 10.1137/130921428. URL https://doi.org/10.1137/130921428.

[164] 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. doi: 10.1177/02783649221082115. URL https://doi.org/10.1177/02783649221082115.

[165] Kenneth Levenberg. A method for the solution of certain non-linear problems in least

squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. doi: 10.1090/qam/10666. URL https://doi.org/10.1090/qam/10666.

[166] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017. doi: 10.1090/mbk/107. URL https://doi.org/10.1090/mbk/107.

[167] Adrian Lewis and Stephen Wright. A proximal method for composite minimization. *Mathematical Programming*, 158(1-2):501–546, jul 2016. doi: 10.1007/s10107-015-0943-9. URL https://doi.org/10.1007/s10107-015-0943-9.

[168] Adrian S Lewis. Active sets, nonsmoothness, and sensitivity. *SIAM Journal on Optimization*, 13(3):702–725, 2002. doi: 10.1137/s1052623401387623. URL https://doi.org/10.1137/S1052623401387623.

[169] Daniel Liberzon. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, dec 2011. ISBN 9780691151878. doi: 10.2307/j.ctvcm4g0s. URL https://doi.org/10.2307/j.ctvcm4g0s.

[170] Qun Lin, Ryan Loxton, and Kok Lay Teo. The control parameterization method for nonlinear optimal control: A survey. *Journal of Industrial and Management Optimization*, 10(1):275–309, 2014. doi: 10.3934/jimo.2014.10.275. URL https://doi.org/10.3934/jimo.2014.10.275.

[171] Xiaoshan Lin, Siddharth Nayak, Stefano Di Cairano, and Abraham P Vinod. Data-driven spatial classification using multi-arm bandits for monitoring with energy-constrained mobile robots. *arXiv preprint arXiv:2501.08222*, 2025. doi: 10.48550/arXiv.2501.08222. URL https://doi.org/10.48550/arXiv.2501.08222.

[172] Lars Lindemann and Dimos V Dimarogonas. Control barrier functions for signal temporal logic tasks. *IEEE Control Systems Letters*, 3(1):96–101, 2018. doi: 10.1109/lcsys.2018.2853182. URL https://doi.org/10.1109/LCSYS.2018.2853182.

[173] Lars Lindemann and Dimos V Dimarogonas. Robust control for signal temporal logic specifications using discrete average space robustness. *Automatica*, 101:377–387, 2019. doi: 10.1016/j.automatica.2018.12.022. URL https://doi.org/10.1016/j.automatica.2018.12.022.

[174] Vladimir J. Lumelsky and KR Harinarayan. Decentralized motion planning for multiple mobile robots: The cocktail party model. *Autonomous Robots*, 4(1):121–135, 1997. doi: 10.1023/A:1008815304810. URL https://doi.org/10.1023/A:1008815304810.

[175] Teppo Luukkonen. Modelling and control of quadcopter, 2011. URL https://sal.aalto.fi/publications/pdf-files/eluu11_public.pdf.

[176] Meiyi Ma, Ji Gao, Lu Feng, and John Stankovic. STLnet: Signal temporal logic enforced multivariate recurrent neural networks. *Advances in Neural Information Processing Systems*, 33:14604–14614, 2020. doi: 10.5555/3495724.3496948. URL https://dl.acm.org/doi/abs/10.5555/3495724.3496948.

[177] L. Magni and R. Scattolini. Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Transactions on Automatic Control*, 49(6):900–906, jun 2004. ISSN 0018-9286. doi: 10.1109/tac.2004.829595. URL http://dx.doi.org/10.1109/TAC.2004.829595.

[178] Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 152–166. Springer, 2004. doi: 10.1007/978-3-540-30206-3_12. URL https://doi.org/10.1007/978-3-540-30206-3_12.

[179] Danylo Malyuta and Behçet Açıkmeşe. Fast homotopy for spacecraft rendezvous trajectory optimization with discrete logic. *Journal of Guidance, Control, and Dynamics*, 46(7):1262–1279, jul 2023. doi: 10.2514/1.g006295. URL https://doi.org/10.2514/1.G006295.

[180] Danylo Malyuta and Behçet Açıkmeşe. Lossless convexification of optimal control problems with semi-continuous inputs. *IFAC-PapersOnLine*, 53(2):6843–6850, 2020. doi: 10.1016/j.ifacol.2020.12.341. URL https://doi.org/10.1016/j.ifacol.2020.12.341.

[181] Danylo Malyuta, Taylor P. Reynolds, Michael Szmuk, Mehran Mesbahi, Behçet Açıkmeşe, and John M. Carson III. Discretization performance and accuracy analysis for the rocket powered descent guidance problem. In *AIAA SciTech 2019 Forum*, Reston, Virginia, jan 2019. doi: 10.2514/6.2019-0925. URL https://doi.org/10.2514/6.2019-0925.

[182] Danylo Malyuta, Taylor P. Reynolds, Michael Szmuk, Behçet Açıkmeşe, and Mehran Mesbahi. Fast trajectory optimization via successive convexification for spacecraft rendezvous with integer constraints. In *AIAA SciTech 2020 Forum*, page 0616, Reston, Virginia, jan 2020. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2020-0616. URL https://doi.org/10.2514/6.2020-0616.

[183] Danylo Malyuta, Yue Yu, Purnanand Elango, and Behçet Açıkmeşe. Advances in trajectory optimization for space vehicle control. *Annual Reviews in Control*, 52:282–315, 2021. doi: 10.1016/j.arcontrol.2021.04.013. URL https://doi.org/10.1016/j.arcontrol.2021.04.013.

[184] Danylo Malyuta, Taylor P Reynolds, Michael Szmuk, Thomas Lew, Riccardo Bonalli, Marco Pavone, and Behçet Açıkmeşe. Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently. *IEEE Control Systems Magazine*, 42(5):40–113, 2022. doi: 10.1109/mcs.2022.3187542. URL https://doi.org/10.1109/MCS.2022.3187542.

[185] Yuanqi Mao, Michael Szmuk, and Behçet Açıkmeşe. Successive convexification of non-convex optimal control problems and its convergence properties. In *2016 IEEE*

55th Conference on Decision and Control (CDC), pages 3636–3641. IEEE, dec 2016. doi: 10.1109/cdc.2016.7798816. URL https://doi.org/10.1109/CDC.2016.7798816.

[186] Yuanqi Mao, Daniel Dueri, Michael Szmuk, and Behçet Açıkmeşe. Convexification and real-time optimization for MPC with aerospace applications. *Handbook of Model Predictive Control*, pages 335–358, 2019. doi: 10.1007/978-3-319-77489-3_15. URL https://doi.org/10.1007/978-3-319-77489-3_15.

[187] Yuanqi Mao, Behçet Açıkmeşe, Pierre-Loic Garoche, and Alexandre Chapoutot. Successive convexification for optimal control with signal temporal logic specifications. In *Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control*, pages 1–7, 2022. doi: 10.1145/3501710.3519518. URL https://doi.org/10.1145/3501710.3519518.

[188] Gregorio Marchesini, Siyuan Liu, Lars Lindemann, and Dimos V. Dimarogonas. Sampling-based planning under stl specifications: A forward invariance approach. *CoRR*, abs/2506.10739, 2025. doi: 10.48550/arXiv.2506.10739. URL https://arxiv.org/abs/2506.10739.

[189] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. doi: 10.1137/0111030. URL https://doi.org/10.1137/0111030.

[190] B. Martinet. R'egularisation d'in'equations variationnelles par approximations successives. *Revue française d'informatique et de recherche op'erationnelle*, 4(R3):154–158, 1970. doi: 10.1051/m2an/197004r301541. URL https://doi.org/10.1051/m2an/197004R301541.

[191] David Q Mayne. Differential dynamic programming—a unified approach to the optimization of dynamic systems. In *Control and Dynamic Systems*, volume 10,

pages 179–254. Elsevier, 1973. doi: 10.1016/b978-0-12-012710-8.50010-8. URL https://doi.org/10.1016/B978-0-12-012710-8.50010-8.

[192] Skye Mceowen and Behçet Açıkmeşe. Hypersonic entry trajectory optimization via successive convexification with abstracted control. In *AIAA SciTech 2022 Forum*, page 0950, 2022. doi: 10.2514/6.2022-0950. URL https://doi.org/10.2514/6.2022-0950.

[193] Skye Mceowen, Abhinav G Kamath, Purnanand Elango, Taewan Kim, Samuel C Buckner, and Behçet Açıkmeşe. High-accuracy 3-dof hypersonic reentry guidance via sequential convex programming. In *AIAA SCITECH 2023 Forum*, page 0300, 2023. doi: 10.2514/6.2023-0300. URL https://doi.org/10.2514/6.2023-0300.

[194] Skye Mceowen, Daniel J Calderone, Aman Tiwary, Jason SK Zhou, Taewan Kim, Purnanand Elango, and Behcet Acikmese. Auto-tuned primal-dual successive convexification for hypersonic reentry guidance. *arXiv preprint arXiv:2411.08361*, page 1317, 2024. doi: 10.48550/arxiv.2411.08361. URL https://doi.org/10.48550/arXiv.2411.08361.

[195] Skye Mceowen, Carlos M Morales, Daniel J Calderone, Samet Uzun, Edgerton M Cook, and Behcet Acikmese. Auto-tuned primal-dual successive convexification for powered descent guidance. In *AIAA SCITECH 2026 Forum*, page 0972, 2026. doi: 10.2514/6.2026-0972. URL https://doi.org/10.2514/6.2026-0972.

[196] Noushin Mehdipour, Cristian-Ioan Vasile, and Calin Belta. Arithmetic-geometric mean robustness for control from signal temporal logic specifications. In *2019 American Control Conference (ACC)*, pages 1690–1695. IEEE, 2019. doi: 10.23919/acc.2019.8814487. URL https://doi.org/10.23919/ACC.2019.8814487.

[197] Noushin Mehdipour, Cristian-Ioan Vasile, and Calin Belta. Average-based robustness for continuous-time signal temporal logic. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 5312–5317. IEEE, 2019. doi: 10.1109/cdc40024.2019.9029989. URL https://doi.org/10.1109/CDC40024.2019.9029989.

[198] Noushin Mehdipour, Cristian-Ioan Vasile, and Calin Belta. Specifying user preferences using weighted signal temporal logic. *IEEE Control Systems Letters*, 5(6):2006–2011, 2020. doi: 10.1109/lcsys.2020.3047362. URL https://doi.org/10.1109/LCSYS.2020.3047362.

[199] Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, 2(4):575–601, 1992. doi: 10.1137/0802028. URL https://doi.org/10.1137/0802028.

[200] Gavin Mendeck and Lynn Craig. Entry guidance for the 2011 Mars Science Laboratory mission. In *AIAA Atmospheric Flight Mechanics Conference*, page 6639, 2011. doi: 10.2514/6.2011-6639. URL https://doi.org/10.2514/6.2011-6639.

[201] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010. doi: 10.1515/9781400835355. URL https://doi.org/10.1515/9781400835355.

[202] Florian Messerer, Katrin Baumgärtner, and Moritz Diehl. Survey of sequential convex programming and generalized gauss-newton methods. *ESAIM: Proceedings and Surveys*, 71:64–88, aug 2021. ISSN 2267-3059,2267-3059. doi: 10.1051/proc/202171107. URL https://doi.org/10.1051/proc/202171107.

[203] Ian M. Mitchell and Shahab Kaynama. An improved algorithm for robust safety analysis of sampled data systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, HSCC '15. ACM, apr 2015. doi: 10.1145/2728606.2728619. URL http://dx.doi.org/10.1145/2728606.2728619.

[204] Bojan Mohar. Eigenvalues, diameter, and mean distance in graphs. *Graphs and combinatorics*, 7(1):53–64, 1991. doi: 10.1007/BF01789463. URL https://doi.org/10.1007/BF01789463.

[205] Jorge J. Moré. The Levenberg–Marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, volume 630 of *Lecture Notes in Mathematics*, pages 105–116. Springer, 1978. doi: 10.1007/bfb0067700. URL https://doi.org/10.1007/BFb0067700.

[206] Luc Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005. doi: 10.1109/TAC.2004.841888. URL https://doi.org/10.1109/TAC.2004.841888.

[207] Daniel Morgan, Soon-Jo Chung, and Fred Y Hadaegh. Model predictive control of swarms of spacecraft using sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 37(6):1725–1740, 2014. doi: 10.2514/1.g000218. URL https://doi.org/10.2514/1.G000218.

[208] Mosek ApS. Mosek optimization toolbox for matlab, 2019. URL https://docs.mosek.com/10.1/toolbox.pdf.

[209] NASA Jet Propulsion Laboratory and Masten Space Systems. 500 meter Xombie divert test flight, 2012. URL http://www.youtube.com/watch?v=1GRwimo1AwY.

[210] NASA Jet Propulsion Laboratory and Masten Space Systems. 750 meter Xombie divert test flight, 2012. URL http://www.youtube.com/watch?v=jl6pw2oossU.

[211] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009. doi: 10.1109/TAC.2008.2009515. URL https://doi.org/10.1109/TAC.2008.2009515.

[212] Yu Nesterov. Modified Gauss–Newton scheme with worst case guarantees for global performance. *Optimisation methods and software*, 22(3):469–483, 2007. doi: 10.1080/08927020600643812. URL https://doi.org/10.1080/08927020600643812.

[213] Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical programming*, 140(1):125–161, 2013. doi: 10.1007/s10107-012-0629-5. URL https://doi.org/10.1007/s10107-012-0629-5.

[214] Yurii Nesterov and Arkadii Nemirovskii. Interior point polynomial algorithms in convex programming. In *SIAM Publications, Philadelphia*, volume 13. SIAM, 1994. doi: 10.1137/1.9781611970791. URL https://doi.org/10.1137/1.9781611970791.

[215] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer, 2 edition, jul 2006. doi: 10.1007/978-0-387-40065-5. URL https://doi.org/10.1007/978-0-387-40065-5.

[216] Peter Ochs, Alexey Dosovitskiy, Thomas Brox, and Thomas Pock. On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. *SIAM Journal on Imaging Sciences*, 8(1):331–372, 2015. doi: 10.1137/140971518. URL https://doi.org/10.1137/140971518.

[217] Peter Ochs, Jalal Fadili, and Thomas Brox. Non-smooth non-convex bregman minimization: Unification and new algorithms. *J. Optim. Theory Appl.*, 181(1): 244–278, apr 2019. ISSN 0022-3239. doi: 10.1007/s10957-018-01452-0. URL https://doi.org/10.1007/s10957-018-01452-0.

[218] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015. doi: 10.1016/j.automatica.2014.10.022. URL https://doi.org/10.1016/j.automatica.2014.10.022.

[219] Reza Olfati-Saber. Distributed kalman filtering for sensor networks. In *2007 46th IEEE Conference on Decision and Control*, pages 5492–5498. IEEE, 2007. doi: 10.1109/CDC.2007.4434303. URL https://doi.org/10.1109/CDC.2007.4434303.

[220] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49

(9):1520–1533, 2004. doi: 10.1109/TAC.2004.834113. URL https://doi.org/10.1109/TAC.2004.834113.

[221] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007. doi: 10.1109/JPROC.2006.887293. URL https://doi.org/10.1109/JPROC.2006.887293.

[222] Alex Olshevsky and John N Tsitsiklis. Degree fluctuations and the convergence time of consensus algorithms. *IEEE Transactions on Automatic Control*, 58(10):2626–2631, 2013. doi: 10.1109/TAC.2013.2257969. URL https://doi.org/10.1109/TAC.2013.2257969.

[223] Yash Vardhan Pant, Houssam Abbas, and Rahul Mangharam. Smooth operator: Control using the smooth robustness of temporal logic. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1235–1240. IEEE, 2017. doi: 10.1109/ccta.2017.8062628. URL https://doi.org/10.1109/CCTA.2017.8062628.

[224] Yash Vardhan Pant, Houssam Abbas, Rhudii A Quaye, and Rahul Mangharam. Fly-by-logic: Control of multi-drone fleets with temporal logic objectives. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 186–197. IEEE, 2018. doi: 10.1109/iccps.2018.00026. URL https://doi.org/10.1109/ICCPS.2018.00026.

[225] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014. doi: 10.1561/2400000003. URL https://doi.org/10.1561/2400000003.

[226] Michael A. Patterson and Anil V. Rao. GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical*

Software, 41(1):1–37, oct 2014. ISSN 0098-3500,1557-7295. doi: 10.1145/2558904. URL https://doi.org/10.1145/2558904.

[227] Natalia Pavlasek, Stefano Di Cairano, and Avishai Weiss. Geostationary satellite station keeping and collocation under high-thrust impulsive control. In *2025 American Control Conference (ACC)*, Denver, CO, USA, 2025. doi: 10.23919/ACC63710.2025.11107882. URL https://doi.org/10.23919/ACC63710.2025.11107882.

[228] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (SFCS 1977)*, pages 46–57. IEEE, 1977. doi: 10.1109/sfcs.1977.32. URL https://doi.org/10.1109/SFCS.1977.32.

[229] Michael JD Powell. A new algorithm for unconstrained optimization. In *Nonlinear programming*, pages 31–65. Elsevier, 1970. doi: 10.1016/b978-0-12-597050-1.50006-3. URL https://doi.org/10.1016/B978-0-12-597050-1.50006-3.

[230] Jiahu Qin, Qichao Ma, Yang Shi, and Long Wang. Recent advances in consensus of multi-agent systems: A brief survey. *IEEE Transactions on Industrial Electronics*, 64 (6):4972–4983, 2016. doi: 10.1109/TIE.2016.2636810. URL https://doi.org/10.1109/TIE.2016.2636810.

[231] R. Quirynen, M. Vukov, M. Zanon, and M. Diehl. Autogenerating microsecond solvers for nonlinear MPC: A tutorial using ACADO integrators. *Optimal Control Applications and Methods*, 36(5):685–704, sep 2015. doi: 10.1002/oca.2152. URL https://doi.org/10.1002/oca.2152.

[232] Rien Quirynen, Milan Vukov, and Moritz Diehl. Multiple shooting in a microsecond. In *Multiple Shooting and Time Domain Decomposition Methods: MuS-TDD, Heidelberg*, pages 183–201. Springer, Springer, 2015. doi: 10.1007/978-3-319-23321-5_7. URL https://doi.org/10.1007/978-3-319-23321-5_7.

[233] 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 *53rd IEEE Conference on Decision and Control*, pages 81–87. IEEE, 2014. doi: 10.1109/cdc.2014.7039363. URL https://doi.org/10.1109/CDC.2014.7039363.

[234] Jaime L Ramirez-Riberos, Marco Pavone, Emilio Frazzoli, and David W Miller. Distributed control of spacecraft formations via cyclic pursuit: Theory and experiments. *Journal of Guidance, Control, and Dynamics*, 33(5):1655–1669, 2010. doi: 10.2514/1.46511. URL https://doi.org/10.2514/1.46511.

[235] James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model Predictive Control: Theory, Computation, and Design*, volume 2. Nob Hill Publishing Madison, WI, 2017. URL https://sites.engineering.ucsb.edu/~jbraw/mpc/.

[236] Wei Ren and Randal W Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, 2005. doi: 10.1109/TAC.2005.846556. URL https://doi.org/10.1109/TAC.2005.846556.

[237] Taylor Reynolds, Danylo Malyuta, Mehran Mesbahi, Behçet Açıkmeşe, and John M. Carson III. A real-time algorithm for non-convex powered descent guidance. In *AIAA SciTech 2020 Forum*, page 0844. AIAA, jan 2020. doi: 10.2514/6.2020-0844. URL https://doi.org/10.2514/6.2020-0844.

[238] Taylor P. Reynolds, Michael Szmuk, Danylo Malyuta, Mehran Mesbahi, Behçet Açıkmeşe, and John M. Carson III. Dual quaternion-based powered descent guidance with state-triggered constraints. *Journal of Guidance, Control, and Dynamics*, 43(9):1584–1599, sep 2020. doi: 10.2514/1.g004536. URL https://doi.org/10.2514/1.G004536.

[239] Arthur Richards, Tom Schouwenaars, Jonathan P How, and Eric Feron. Spacecraft

trajectory planning with avoidance constraints using mixed-integer linear programming. *Journal of Guidance, Control, and Dynamics*, 25(4):755–764, 2002. doi: 10.2514/2.4943. URL https://doi.org/10.2514/2.4943.

[240] Stephen M. Robinson. Strongly regular generalized equations. *Mathematics of Operations Research*, 5(1):43–62, 1980. doi: 10.1287/moor.5.1.43. URL https://doi.org/10.1287/moor.5.1.43.

[241] R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976. doi: 10.1137/0314056. URL https://doi.org/10.1137/0314056.

[242] R Tyrrell Rockafellar. Second-order optimality conditions in nonlinear programming obtained by way of epi-derivatives. *Mathematics of Operations Research*, 14(3):462–484, 1989. doi: 10.1287/moor.14.3.462. URL https://doi.org/10.1287/moor.14.3.462.

[243] R. Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1998. doi: 10.1007/978-3-642-02431-3. URL https://doi.org/10.1007/978-3-642-02431-3.

[244] Alëna Rodionova, Lars Lindemann, Manfred Morari, and George J Pappas. Time-robust control for STL specifications. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 572–579. IEEE, 2021. doi: 10.1109/cdc45484.2021.9683477. URL https://doi.org/10.1109/CDC45484.2021.9683477.

[245] Alëna Rodionova, Lars Lindemann, Manfred Morari, and George J Pappas. Combined left and right temporal robustness for control under STL specifications. *IEEE Control Systems Letters*, 7:619–624, 2022. doi: 10.1109/lcsys.2022.3209928. URL https://doi.org/10.1109/LCSYS.2022.3209928.

[246] Federico Rossi, Saptarshi Bandyopadhyay, Michael Wolf, and Marco Pavone. Review of multi-agent algorithms for collective behavior: a structural taxonomy. *IFAC-*

PapersOnLine, 51(12):112–117, 2018. doi: 10.1016/j.ifacol.2018.07.097. URL https://doi.org/10.1016/j.ifacol.2018.07.097.

[247] 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. doi: 10.1109/allerton.2015.7447084. URL https://doi.org/10.1109/ALLERTON.2015.7447084.

[248] Sadra Sadraddini and Russ Tedrake. Robust output feedback control with guaranteed constraint satisfaction. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, HSCC '20. ACM, apr 2020. doi: 10.1145/3365365.3382211. URL http://dx.doi.org/10.1145/3365365.3382211.

[249] Yunus Emre Sahin, Petter Nilsson, and Necmiye Ozay. Multirobot coordination with counting temporal logics. *IEEE Transactions on Robotics*, 36(4):1189–1206, 2019. doi: 10.1109/tro.2019.2957669. URL https://doi.org/10.1109/TRO.2019.2957669.

[250] Yunus Emre Sahin, Rien Quirynen, and Stefano Di Cairano. Autonomous vehicle decision-making and monitoring based on signal temporal logic and mixed-integer programming. In *2020 American Control Conference (ACC)*, pages 454–459. IEEE, 2020. doi: 10.23919/acc45564.2020.9147917. URL https://doi.org/10.23919/ACC45564.2020.9147917.

[251] Paolo Salaris, Marco Cognetti, Riccardo Spica, and Paolo Robuffo Giordano. Online optimal perception-aware trajectory generation. *IEEE Trans. Robotics*, 35(6):1307–1322, 2019. doi: 10.1109/TRO.2019.2931137. URL https://doi.org/10.1109/TRO.2019.2931137.

[252] A Miguel San Martin, Steven W Lee, and Edward C Wong. The development of the msl guidance, navigation, and control system for entry, descent, and landing. In *AAS/AIAA*

Spaceflight Mechanics Meeting, 2013. URL https://ntrs.nasa.gov/citations/20150008675.

[253] R. W. H. Sargent and G. R. Sullivan. The development of an efficient optimal control package. In *Proceedings of the 8th IFIP Conference on Optimization Techniques, Würzburg, September 5–9, 1977*, volume 7, pages 158–168, Berlin/Heidelberg, 1977. Springer, Springer-Verlag. doi: 10.1007/bfb0006520. URL https://doi.org/10.1007/BFb0006520.

[254] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen. A survey of spacecraft formation flying guidance and control (part 1): guidance. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 2, pages 1733–1739, 2003. doi: 10.1109/ACC.2003.1239845. URL https://doi.org/10.1109/ACC.2003.1239845.

[255] Daniel P Scharf, Martin W Regehr, Geoffery M Vaughan, Joel Benito, Homayoon Ansari, MiMi Aung, Andrew Johnson, Jordi Casoliva, Swati Mohan, Daniel Dueri, et al. Adapt demonstrations of onboard large-divert guidance with a vtvl rocket. In *2014 IEEE Aerospace Conference*, pages 1–18. IEEE, 2014. doi: 10.1109/aero.2014.6836462. URL https://doi.org/10.1109/AERO.2014.6836462.

[256] Daniel P Scharf, Behçet Açıkmeşe, Daniel Dueri, Joel Benito, and Jordi Casoliva. Implementation and experimental demonstration of onboard powered-descent guidance. *Journal of Guidance, Control, and Dynamics*, 40(2):213–229, 2017. doi: 10.2514/1.g000399. URL https://doi.org/10.2514/1.G000399.

[257] David Schlipf, Dominik Johannes Schlipf, and Martin Kühn. Nonlinear model predictive control of wind turbines using lidar. *Wind Energy*, 16(7):1107–1129, 2012. ISSN 1099-1824. doi: 10.1002/we.1533. URL https://doi.org/10.1002/we.1533.

[258] Melanie Schranz, Martina Umlauft, Micha Sende, and Wilfried Elmenreich. Swarm

robotic behaviors and current applications. *Frontiers in Robotics and AI*, 7:36, 2020. doi: 10.3389/frobt.2020.00036. URL https://doi.org/10.3389/frobt.2020.00036.

[259] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015. doi: 10.1137/14096668X. URL https://doi.org/10.1137/14096668X.

[260] Harold Shniad. On the convexity of mean value functions. *Bulletin of the American Mathematical Society*, 54:770–776, 1948. doi: 10.1090/s0002-9904-1948-09077-2. URL https://doi.org/10.1090/S0002-9904-1948-09077-2.

[261] Gurkirpal Singh, Alejandro M SanMartin, and Edward C Wong. Guidance and control design for powered descent and landing on Mars. In *2007 IEEE Aerospace Conference*, pages 1–8. IEEE, 2007. doi: 10.1109/aero.2007.352818. URL https://doi.org/10.1109/AERO.2007.352818.

[262] Ondrej Slučiak and Markus Rupp. Consensus algorithms with state-dependent weights. *IEEE Transactions on Signal Processing*, 64(8):1972–1985, 2016. doi: 10.1109/TSP.2016.2515074. URL https://doi.org/10.1109/TSP.2016.2515074.

[263] Ronald Sostaric and Jeremy Rea. Powered descent guidance methods for the Moon and Mars. In *AIAA guidance, navigation, and control conference and exhibit*, page 6287, 2005. doi: 10.2514/6.2005-6287. URL https://doi.org/10.2514/6.2005-6287.

[264] Fabio Spada, Purnanand Elango, and Behçet Açıkmeşe. Impulsive relative motion control with continuous-time constraint satisfaction for cislunar space missions. In *2025 American Control Conference (ACC)*, pages 1719–1724. IEEE, 2025. doi: 10.23919/ACC63710.2025.11108021. URL https://doi.org/10.23919/ACC63710.2025.11108021.

[265] Dick Stanley. A multiplicative calculus. *Problems, Resources, and Issues in Mathematics Undergraduate Studies*, 9(4):310–326, 1999. doi: 10.1080/10511979908965937. URL https://doi.org/10.1080/10511979908965937.

[266] Lloyd Strohl, Javier Doll, Matthew Fritz, Andrew W Berning, Stephanie White, Stefan R Bieniawski, John M Carson, and Behçet Açıkmeşe. Implementation of a six degree of freedom precision lunar landing algorithm using dual quaternion representation. In *AIAA SCITECH 2022 Forum*, page 1831, 2022. doi: 10.2514/6.2022-1831. URL https://doi.org/10.2514/6.2022-1831.

[267] Aamodh Suresh and Sonia Martínez. Risk-perception-aware control design under dynamic spatial risks. *IEEE Control Sys. Lett.*, 2021. doi: 10.1109/LCSYS.2021.3134942. URL https://doi.org/10.1109/LCSYS.2021.3134942.

[268] Michael Szmuk and Behçet Açıkmeşe. Successive convexification for 6-dof mars rocket powered landing with free-final-time. In *2018 AIAA Guidance, Navigation, and Control Conference*, page 0617. AIAA, jan 2018. doi: 10.2514/6.2018-0617. URL https://doi.org/10.2514/6.2018-0617.

[269] Michael Szmuk, Behçet Açıkmeşe, and Andrew W. Berning. Successive convexification for fuel-optimal powered landing with aerodynamic drag and non-convex constraints. In *AIAA Guidance, Navigation, and Control Conference*, page 0378, Reston, Virginia, jan 2016. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2016-0378. URL https://doi.org/10.2514/6.2016-0378.

[270] Michael Szmuk, Carlo Alberto Pascucci, Daniel Dueri, and Behçet Açıkmeşe. Convexification and real-time on-board optimization for agile quad-rotor maneuvering and obstacle avoidance. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4862–4868. IEEE, IEEE, sep 2017. doi: 10.1109/iros.2017.8206363. URL https://doi.org/10.1109/IROS.2017.8206363.

[271] Michael Szmuk, Carlo Alberto Pascucci, and Behçet Açıkmeşe. Real-time quad-rotor path planning for mobile obstacle avoidance using convex optimization. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*,

pages 1–9. IEEE, IEEE, oct 2018. doi: 10.1109/iros.2018.8594351. URL https://doi.org/10.1109/IROS.2018.8594351.

[272] Michael Szmuk, Danylo Malyuta, Taylor P. Reynolds, Margaret Skye Mceowen, and Behçet Açıkmeşe. Real-time quad-rotor path planning using convex optimization and compound state-triggered constraints. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7666–7673. IEEE, IEEE, nov 2019. doi: 10.1109/iros40897.2019.8967706. URL https://doi.org/10.1109/IROS40897.2019.8967706.

[273] Michael Szmuk, Taylor P. Reynolds, Behçet Açıkmeşe, Mehran Mesbahi, and John M. Carson III. Successive convexification for 6-DoF powered descent guidance with compound state-triggered constraints. In *AIAA SciTech 2019 Forum*, page 0926, Reston, Virginia, jan 2019. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2019-0926. URL https://doi.org/10.2514/6.2019-0926.

[274] Michael Szmuk, Taylor P. Reynolds, and Behçet Açıkmeşe. Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints. *Journal of Guidance, Control, and Dynamics*, 43(8):1399–1413, aug 2020. ISSN 1533-3884. doi: 10.2514/1.g004549. URL http://dx.doi.org/10.2514/1.G004549.

[275] Alireza Tahbaz-Salehi and Ali Jadbabaie. Small world phenomenon, rapidly mixing markov chains, and average consensus algorithms. In *2007 46th IEEE Conference on Decision and Control*, pages 276–281. IEEE, 2007. doi: 10.1109/CDC.2007.4434174. URL https://doi.org/10.1109/CDC.2007.4434174.

[276] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE, 2012. doi: 10.1109/iros.2012.6386025. URL https://doi.org/10.1109/IROS.2012.6386025.

[277] Yuval Tassa, Nicolas Mansard, and Emo Todorov. Control-limited differential dynamic programming. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1168–1175. IEEE, 2014. doi: 10.1109/icra.2014.6907001. URL https://doi.org/10.1109/ICRA.2014.6907001.

[278] Parth Thaker, Stefano Di Cairano, and Abraham P Vinod. Bandit-based multi-agent search under noisy observations. In *Proc. IFAC World Congress*, 2023. doi: 10.1016/j.ifacol.2023.10.1377. URL https://doi.org/10.1016/j.ifacol.2023.10.1377.

[279] Andreas Themelis, Lorenzo Stella, and Panagiotis Patrinos. Forward–backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone line-search algorithms. *SIAM Journal on Optimization*, 28(3):2274–2303, 2018. doi: 10.1137/16m1080240. URL https://doi.org/10.1137/16M1080240.

[280] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996. doi: 10.1111/j.2517-6161.1996.tb02080.x. URL https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[281] Michael Tillerson, Gokhan Inalhan, and Jonathan P How. Co-ordination and control of distributed spacecraft systems using convex optimization techniques. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 12(2-3):207–242, 2002. doi: 10.1002/rnc.683. URL https://doi.org/10.1002/rnc.683.

[282] Jesus Tordesillas and Jonathan P How. PANTHER: Perception-aware trajectory planner in dynamic environments. *IEEE Access*, 10:22662–22677, 2022. doi: 10.1109/ACCESS.2022.3154037. URL https://doi.org/10.1109/ACCESS.2022.3154037.

[283] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer, 2000. doi: 10.1007/3-540-44480-7_21. URL https://doi.org/10.1007/3-540-44480-7_21.

[284] Samet Uzun and Behçet Açıkmeşe. Successive convexification for trajectory optimization with continuous-time satisfaction of signal temporal logic specifications, 2025.

[285] Samet Uzun and Behçet Açıkmeşe. Smooth and exact parameterization of continuous-time signal temporal logic specifications for trajectory optimization. *arXiv preprint arXiv:2604.04245*, 2026. doi: 10.48550/arXiv.2604.04245. URL https://doi.org/10.48550/arXiv.2604.04245.

[286] Samet Uzun, Purnanand Elango, Pierre-Loic Garoche, and Behçet Açıkmeşe. Optimization with temporal and logical specifications via generalized mean-based smooth robustness measures. *arXiv preprint arXiv:2405.10996*, 2024. doi: 10.48550/arxiv.2405.10996. URL https://doi.org/10.48550/arXiv.2405.10996.

[287] Samet Uzun, Purnanand Elango, Abhinav G Kamath, Taewan Kim, and Behçet Açıkmeşe. Successive convexification for nonlinear model predictive control with continuous-time constraint satisfaction. *IFAC-PapersOnLine*, 58(18):421–429, aug 2024. doi: 10.1016/j.ifacol.2024.09.062. URL https://doi.org/10.1016/j.ifacol.2024.09.062.

[288] Samet Uzun, Nazım Kemal Üre, and Behçet Açıkmeşe. Decentralized state-dependent markov chain synthesis with an application to swarm guidance. *IEEE Transactions on Automatic Control*, 2024. doi: 10.1109/tac.2024.3353673. URL https://doi.org/10.1109/TAC.2024.3353673.

[289] Samet Uzun, Behçet Açıkmeşe, and John M Carson. Sequential convex programming for 6-dof powered descent guidance with continuous-time compound state-triggered constraints. In *AIAA SCITECH 2025 Forum*, page 1895. American Institute of Aeronautics and Astronautics, jan 2025. doi: 10.2514/6.2025-1895. URL https://doi.org/10.2514/6.2025-1895.

[290] Samet Uzun, Behçet Açıkmeşe, and Stefano Di Cairano. Motion planning for information

acquisition via continuous-time successive convexification. *IEEE Control Systems Letters*, 2025. doi: 10.1109/lcsys.2025.3576066. URL https://doi.org/10.1109/LCSYS.2025.3576066.

[291] Samet Uzun, Dayou Luo, Behçet Açıkmeşe, and Aleksandr Y Aravkin. A proximal method for composite optimization with smooth and convex components. *arXiv preprint arXiv:2512.20602*, 2025. doi: 10.48550/arXiv.2512.20602. URL https://arxiv.org/abs/2512.20602.

[292] Peter Varnai and Dimos V Dimarogonas. On robustness metrics for learning STL tasks. In *2020 American Control Conference (ACC)*, pages 5394–5399. IEEE, 2020. doi: 10.23919/acc45564.2020.9147692. URL https://doi.org/10.23919/ACC45564.2020.9147692.

[293] Joris Verhagen, Lars Lindemann, and Jana Tumova. Temporally robust multi-agent stl motion planning in continuous time. *arXiv preprint arXiv:2310.10585*, pages 251–258, 2024. doi: 10.23919/acc60939.2024.10644216. URL https://doi.org/10.23919/acc60939.2024.10644216.

[294] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Jonathan Frey, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. Acados-a modular open-source framework for fast embedded optimal control. *Mathematical programming computation*, 14(1):147–183, mar 2022. doi: 10.1007/s12532-021-00208-8. URL https://doi.org/10.1007/s12532-021-00208-8.

[295] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, mar 2006. doi: 10.1007/s10107-004-0559-y. URL https://doi.org/10.1007/s10107-004-0559-y.

[296] Gerhard Wanner and Ernst Hairer. *Solving Ordinary Differential Equations II*, volume

375. Springer Berlin Heidelberg New York, 1996. doi: 10.1007/978-3-642-05221-7. URL https://doi.org/10.1007/978-3-642-05221-7.

[297] David W Way, Paul Brugarolas, Gurkirpal Singh, and Jordi Casoliva. Reconstructed flight performance of the powered descent guidance and control system for the mars 2020 perseverance mission, 2021. URL https://ntrs.nasa.gov/citations/20230005636.

[298] Avishai Weiss and Stefano Di Cairano. Robust dual control MPC with guaranteed constraint satisfaction. In *53rd IEEE Conference on Decision and Control*. IEEE, dec 2014. doi: 10.1109/cdc.2014.7040443. URL http://dx.doi.org/10.1109/CDC.2014.7040443.

[299] Tichakorn Wongpiromsarn, Ufuk Topcu, Necmiye Ozay, Huan Xu, and Richard M Murray. TuLiP: a software toolbox for receding horizon temporal logic planning. In *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, pages 313–314, 2011. doi: 10.1145/1967701.1967747. URL https://doi.org/10.1145/1967701.1967747.

[300] Tichakorn Wongpiromsarn, Ufuk Topcu, and Richard M Murray. Receding horizon temporal logic planning. *IEEE Transactions on Automatic Control*, 57(11):2817–2830, 2012. doi: 10.1109/tac.2012.2195811. URL https://doi.org/10.1109/TAC.2012.2195811.

[301] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004. doi: 10.1016/j.sysconle.2004.02.022. URL https://doi.org/10.1016/j.sysconle.2004.02.022.

[302] Lin Xiao, Stephen Boyd, and Seung-Jean Kim. Distributed average consensus with least-mean-square deviation. *Journal of parallel and distributed computing*, 67(1):33–46, 2007. doi: 10.1016/j.jpdc.2006.08.010. URL https://doi.org/10.1016/j.jpdc.2006.08.010.

[303] Shaobo Xie, Xiaosong Hu, Zongke Xin, and James Brighton. Pontryagin's minimum principle based model predictive control of energy management for a plug-in hybrid electric bus. *Applied Energy*, 236:893–905, 2019. doi: 10.1016/j.apenergy.2018.12.032. URL https://doi.org/10.1016/j.apenergy.2018.12.032.

[304] Guang Yang, Calin Belta, and Roberto Tron. Continuous-time signal temporal logic planning with control barrier functions. In *2020 American Control Conference (ACC)*, pages 4612–4618. IEEE, 2020. doi: 10.23919/acc45564.2020.9147387. URL https://doi.org/10.23919/acc45564.2020.9147387.

[305] Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019. doi: 10.1016/j.arcontrol.2019.05.006. URL https://doi.org/10.1016/j.arcontrol.2019.05.006.

[306] Pian Yu, Xiao Tan, and Dimos V Dimarogonas. Continuous-time control synthesis under nested signal temporal logic specifications. *IEEE Transactions on Robotics*, 40: 2272–2286, 2024. doi: 10.1109/tro.2024.3353081. URL https://doi.org/10.1109/tro.2024.3353081.

[307] Yue Yu, Purnanand Elango, Behçet Açıkmeşe, and Ufuk Topcu. Extrapolated proportional-integral projected gradient method for conic optimization. *IEEE Control Systems Letters*, 7:73–78, 2022. doi: 10.1109/lcsys.2022.3186647. URL https://doi.org/10.1109/LCSYS.2022.3186647.

[308] Yue Yu, Purnanand Elango, Ufuk Topcu, and Behçet Açıkmeşe. Proportional–integral projected gradient method for conic optimization. *Automatica*, 142:110359, aug 2022. doi: 10.1016/j.automatica.2022.110359. URL https://doi.org/10.1016/j.automatica.2022.110359.

[309] Ya-xiang Yuan. Conditions for convergence of trust region algorithms for nonsmooth

optimization. *Mathematical Programming*, 31(2):220–228, 1985. doi: 10.1007/bf02591750. URL https://doi.org/10.1007/BF02591750.

[310] Yating Yuan. Continuous-time control synthesis for multiple quadrotors under signal temporal logic specifications. *CoRR*, abs/2505.07240, 2025. doi: 10.48550/arXiv.2505.07240. URL https://arxiv.org/abs/2505.07240.

[311] Yating Yuan, Thanin Quartz, and Jun Liu. Signal temporal logic planning with time-varying robustness. *IEEE Control Systems Letters*, 8:3015–3020, 2024. doi: 10.1109/lcsys.2024.3520152. URL https://doi.org/10.1109/lcsys.2024.3520152.

[312] Melanie N. Zeilinger, Davide M. Raimondo, Alexander Domahidi, Manfred Morari, and Colin N. Jones. On real-time robust model predictive control. *Automatica*, 50(3): 683–694, mar 2014. ISSN 0005-1098. doi: 10.1016/j.automatica.2013.11.019. URL http://dx.doi.org/10.1016/j.automatica.2013.11.019.

[313] Jian Zhang, Jianbo Hu, Juyu Lian, Zongji Fan, Xuejun Ouyang, and Wanhui Ye. Seeing the forest from drones: Testing the potential of lightweight drones as a tool for long-term forest monitoring. *Biological Conserv.*, 2016. doi: 10.1016/j.biocon.2016.03.027. URL https://doi.org/10.1016/j.biocon.2016.03.027.

[314] Sheng Zhao, Subramanian Ramakrishnan, and Manish Kumar. Density-based control of multiple robots. In *Proceedings of the 2011 American control conference*, pages 481–486. IEEE, 2011. doi: 10.1109/ACC.2011.5990615. URL https://doi.org/10.1109/ACC.2011.5990615.