Robust Predictive Control for Uncertain Nonlinear Systems via Funnel Synthesis

Taewan Kim

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

Doctor of Philosophy

University of Washington

2025

Reading Committee:

Behçet Açıkmeşe, Chair

Mehran Mesbahi

Maryam Fazel

Program Authorized to Offer Degree:
William E. Boeing Department of Aeronautics & Astronautics

University of Washington

Abstract

Robust Predictive Control for Uncertain Nonlinear Systems via Funnel Synthesis

Taewan Kim

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

This thesis studies robust predictive control for uncertain nonlinear systems, with an emphasis on constrained control synthesis through convex optimization-based approaches. Ensuring constraint satisfaction and robustness under disturbances and model uncertainty is a central challenge in the design of control systems for safety-critical applications. The control framework consists of two modules: a trajectory generation module that computes a nominal state and an open-loop input, and a state feedback module that compensates for deviations from the nominal. The combined input is applied to the system to ensure robustness and constraint satisfaction. This thesis develops novel methods and theoretical results that enable the design of such systems under broader classes of nonlinearities and uncertainties, while improving computational efficiency and enhancing the accuracy of constraint satisfaction.

A central element is funnel synthesis, in which time-varying invariant funnels and associated feedback controllers are synthesized around nominal trajectories. New formulations are introduced using time-varying incremental quadratic constraints, enabling the handling of nonlinearities beyond Lipschitz continuity. The funnel invariance condition is derived via a differential linear matrix inequality using Lyapunov theory. To solve the continuous-time funnel synthesis problem, an optimal control framework supporting higher-order funnel representations is proposed.

Three convex approaches are developed to address continuous-time constraint

satisfaction (CTCS) for funnel synthesis. The first introduces intermediate constraint-checking points without increasing the number of decision variables. The second reformulates continuous-time linear matrix inequalities as nodal constraints via an exterior penalty on constraint violations, evaluated at discrete time points and solved using a subgradient-based successive convexification algorithm. The third approach, based on a matrix copositivity condition, achieves CTCS without additional intermediate checking points or constraint reformulation, offering a structured but conservative alternative.

Finally, joint synthesis algorithms are proposed to compute both the nominal trajectory and the associated funnel within a unified framework, reducing the conservatism that arises when they are optimized separately. The proposed methods are demonstrated on systems including a unicycle, a 6-degree-of-freedom (6-DoF) free-flyer, a 6-DoF quadrotor, and a powered descent guidance scenario for a 6-DoF rocket.

DEDICATION

To Seungbin, whose love made my world stable and converge.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ACKNOWLEDGMENTS

Arguably, the most important things I gained during my PhD study were not the degree itself, but the relationships I built with people. Nothing in this thesis would have been possible without the help and support of the people acknowledged here.

Foremost, I would like to express my sincere gratitude to my advisor, Professor Behçet Açıkmeşe. Your endless passion for control and optimization has been a constant source of motivation for me. Our meetings have always been highly instructive, and through them, I have learned how to approach problems with clear, plausible reasoning, free from unnecessary assumptions. You have shown me what it means to be a thoughtful researcher.

I would also like to thank Professor Mehran Mesbahi. Every opportunity I have had to communicate with you has been truly valuable to me. Your thoughtful comments and insightful questions have played an important role in shaping my research. To me, you have embodied the very standard of what it means to be a great professor.

My sincere thanks also go to Professors Maryam Fazel, Abhishek Gupta, and Dana Dabiri for serving on my final exam committee. In particular, I am deeply grateful to Professor Fazel. It has been an honor to discuss research with you, and your ongoing work in control, optimization, and learning has been a profound inspiration to me.

The memories of working with wonderful colleagues, both personally and technically outstanding, will stay with me for a long time. I am grateful to the members of the Autonomous Controls Lab (ACL): Purnanand Elango, Samet Uzun, Skye Mceowen, Abhinav Kamath, Govind Chari, Dayou Luo, Kazuya Echigo, Justin Ganiban, Jason Zhou, Samuel Buckner, Chris Hayner, Natalia Pavlasek, Fabio Spada, Aman Tiwary, Avi Mittal, Sarah Li, and many others. I am also grateful to fellow UW PhD students outside the ACL, including Eddie Ting, Nick Andrews, and Danny Broyles. My heartfelt thanks also go to

our great UW alumni: Danylo Malyuta, Michael Szmuk, and Taylor Reynolds. I could write at length about the wonderful memories I have shared with each of you mentioned in this paragraph, but I will keep this section brief and convey my gratitude to you personally.

I am also grateful to my industry colleagues, Stefano Di Cairano at MERL and Jasper Corleis at Boeing. My internship with Stefano was a wonderful and formative experience, and my collaboration with Jasper has been an invaluable and rewarding part of my research journey.

I could not have begun my research journey without the guidance and support of my master's advisor, Professor H. Jin Kim at Seoul National University. Her extensive expertise in the field of control and robotics has inspired me to aspire to follow in her footsteps. I am also deeply thankful to Professor Jonghyeon Park at Hanyang University, Professor Jongho Shin at Chungbuk National University, and Myungjun Lee at Samsung Electronics for their support.

I would like to thank my parents and two sisters for their unconditional love and encouragement throughout this journey. I hold in my heart the memory of my beloved dog Dol-dol, who has crossed the rainbow bridge. The happiness and comfort you gave me will stay with me forever. To my family, I owe my deepest gratitude; your love has been the foundation of my journey.

As I wrote in the Dedication, all my love goes to my wife, Seungbin, and I thank my son, Doyoung, for the happiness he has brought into my life.

NOMENCLATURE

CIF: Controlled invariant funnel.

CTCS: Continuous-time constraint satisfaction.

DDP: Differential dynamic programming.

DLME: Differential linear matrix equality.

DLMI: Differential linear matrix inequality.

DME: Differential matrix equality.

FOH: First-order hold.

HJ: Hamilton-Jacobi.

IQC: Integral quadratic constraints.

LICQ: Linear independence constraint qualification.

LMI: Linear matrix inequality.

LP: Linear programming.

LPV: Linear parameter-varying.

LTV: Linear time-varying.

MPC: Model predictive control.

PDP: Positive definite.

PMP: Pontryagin's maximum principle.

PSD: Positive semidefinite.

PTR: Penalized trust-region.

SCP: Sequential convex programming.

SCVX: Successive convexification.

SDP: Semidefinite program.

SOS: Sums-of-squares.

ZOH: Zeroth-order hold.

δQC: Incremental quadratic constraint.

δQS: Incrementally quadratically stable.

Chapter 1
INTRODUCTION

This thesis studies optimization-based constrained robust predictive control for uncertain nonlinear systems. The control of nonlinear systems under uncertainty has long been a fundamental challenge in systems and control engineering. *Uncertain systems* are those whose exact behavior cannot be fully characterized, often due to unknown parameters, external disturbances, or incomplete models. *Robust control* seeks to guarantee stability and safety despite such uncertainties. A control method is considered to be *predictive* if it uses a model of the system to simulate or forecast future behavior, allowing current control actions to be determined based on predicted future outputs. *Constrained control* addresses the need to respect physical, safety, or operational limits on system states and inputs, which is essential in practical applications. By integrating these aspects, this thesis develops an *optimization-based* control framework in which control laws are obtained by solving optimization problems.

The control framework employed in this research is illustrated in Figure 1.1. The pair $(\bar{x}, \bar{u})$ denotes the nominal trajectory, consisting of an open-loop state $\bar{x}$ and input $\bar{u}$, generated by a trajectory optimization module. The feedback control input, denoted by u_{fb}, is a function of the current system state x and the nominal state $\bar{x}$, and is designed to compensate for deviations caused by uncertainties and disturbances. The operator Δ represents an uncertainty mapping that receives an input signal q from the system and produces an output p, which perturbs the system dynamics and captures unmodeled effects. The external disturbance is denoted by w. This control architecture is referred to as *predictive compensation*.

This thesis focuses on two modules within the proposed architecture: trajectory optimization and state feedback control. The trajectory optimization, which is a subclass of trajectory generation or motion planning, aims to generate a nominal trajectory by solving

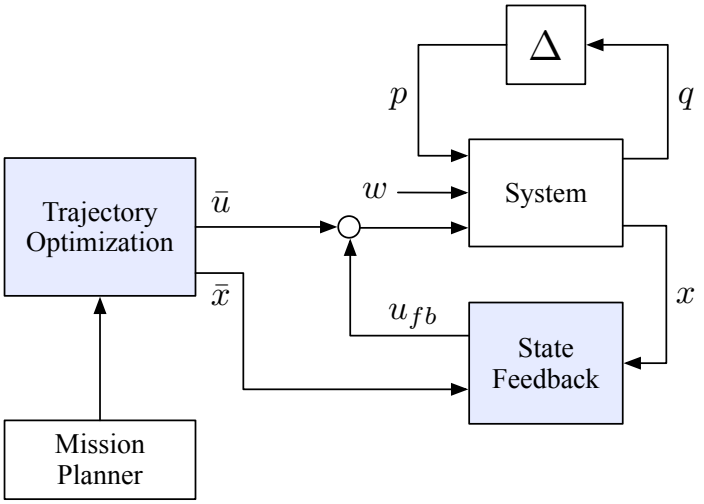

Figure 1.1: The control architecture employed in this thesis.

a constrained optimization problem. The key properties the nominal trajectory should have are its ability to satisfy mission constraints and optimize key objectives such as minimizing time or fuel consumption. To consider a wide range of constraints and objectives, numerous studies have investigated various modeling, formulation, and solution approaches [72, 77].

The generated nominal trajectory alone is not sufficient to control systems subject to disturbances or uncertainties. This is because disturbances and uncertainties can easily cause the system to deviate from the nominal trajectory, and an open-loop control alone is not able to bring the system back. In this context, feedback control plays a critical role in compensating for these deviations. With feedback, the closed-loop system can achieve *robustness* to disturbances and uncertainties.

While feedback control enhances robustness, imposing constraints on the feedback control is not straightforward. In trajectory optimization, constraints are imposed directly on a single nominal trajectory, making enforcement conceptually simple. However, feedback control governs an entire set of possible trajectories that may arise from different initial conditions and disturbances. As a result, enforcing constraints under feedback requires

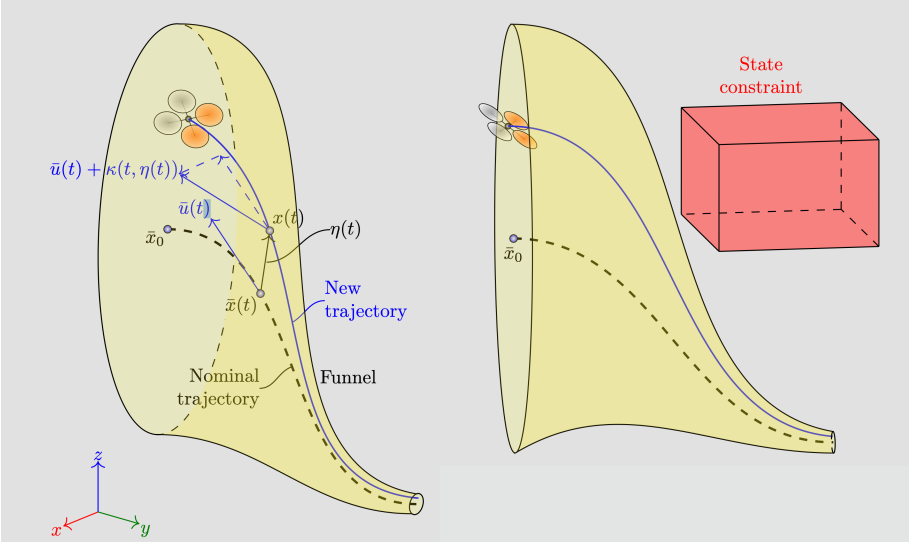

Figure 1.2: Funnel synthesis: a procedure for computing both the funnel and the associated feedback control. Figure adapted from [76].

ensuring that all possible closed-loop trajectories remain within the allowable bounds.

One possible approach to remedy this issue is to compute a *funnel*, which represents an invariant set of the closed-loop system, and impose constraints on this funnel. By doing so, it becomes possible to ensure constraint satisfaction for all possible trajectories resulting from the feedback control policy. Motivated by this, this thesis primarily studies *funnel synthesis* that refers to a procedure for computing both controlled invariant funnels and associated feedback control [97]. Then, the computed funnels serve as a certificate that, for any state within them, there exists a control signal, provided by the associated controller, that ensures constraint satisfaction, and consequently, the system's safety, under the presence of disturbances. An illustration of funnels is given in Figure 1.2.

The computed funnels and associated controllers can be used for various purposes. First, they enable constrained robust control for uncertain systems, as they guarantee the closed-loop system's constraint satisfaction under the presence of uncertainties [4]. Second, the funnels can be used to generate feasible trajectories [97] by numerically integrating the

system dynamics with the synthesized controller. This eliminates the need to repeatedly solve the trajectory optimization problem when problem data, such as the initial condition, changes. This can address a common limitation of many nonconvex trajectory optimization methods, which often lack guarantees that the converged solution satisfies constraints [77]. Furthermore, the computed funnels can be used to ensure recursive feasibility of model predictive control (MPC) for nominal trajectory tracking. By enforcing a terminal constraint that requires the predicted final state to lie within the funnel, the method provides a natural mechanism to maintain feasibility across MPC iterations. In addition, the computed funnels can be used for feasible-trajectory data generation, which is useful for training neural network controllers under safety constraints [58].

Due to these benefits of funnels, the problem of computing funnels has been actively studied in both the control [4, 83] and robotics [24, 74] communities. Despite this progress, several key challenges remain. First, extending funnel synthesis to broader classes of nonlinear systems requires going beyond linear or polynomial dynamics by exploiting structural properties of nonlinearities. Second, ensuring invariance, which is typically derived through Lyapunov methods [30], Hamilton-Jacobi (HJ) reachability analysis [13], or direct uncertainty propagation [112], is often computationally expensive, particularly for high-dimensional or nonlinear systems. Third, achieving constraint satisfaction for all states and inputs within the associated state and input funnels is nontrivial and is often handled indirectly by adjusting the nominal trajectory, rather than explicitly enforcing constraints on the funnel itself. Finally, because funnels are inherently time-varying, they are commonly represented using basis function approximations, such as zero-order hold or first-order hold interpolations, which can limit expressiveness.

The design of funnels for uncertain systems traditionally follows a two-step (or separate) scheme; initially, nominal trajectory planning [18, 77] is performed to compute the open-loop control and the corresponding state trajectory, followed by the synthesis of the feedback control and the associated funnel [22, 74] based on the analysis of the perturbed system around the nominal trajectory. In aerospace applications, the former is often termed as *guidance* and the latter is referred to as *control*. However, such a two step scheme has a potential drawback; the resulting control law, consisting of both the open-loop and closed-

loop control, may be overly conservative. This conservatism stems from the lack of joint consideration of the open-loop and closed-loop control computations for given constraints such as actuator limits and obstacles.

This thesis aims to address the aforementioned challenges in funnel computation. The summary of the problem and the approach taken in each chapter is as follows. Chapter 3 formulates a continuous-time constrained funnel synthesis problem for uncertain nonlinear systems. The formulation considers uncertain nonlinear systems subject to external bounded disturbances, where uncertainties and nonlinearities satisfy incremental quadratic constraints (δQCs) [6, 33, 127]. The expression with δQC can characterize a broader class of uncertainties/nonlinearities, including Lipschitz, L-smooth, and sector-bounded nonlinearities. The state funnels are defined as time-varying ellipsoidal sets around a nominal trajectory, described by a quadratic Lyapunov function with a positive definite (PD) matrix-valued function, as decision variables. Alongside the nominal input, a linear time-varying (LTV) feedback controller is applied, with the LTV gain matrices serving as other decision variables. Funnel invariance is ensured by Lyapunov theory, resulting in a differential linear matrix inequality (DLMI). To ensure constraint satisfaction, all states and inputs within the funnel must satisfy state and input constraints. When these constraints are linear or locally linearized around the nominal trajectory, they can be encoded as pointwise-in-time LMIs. The resulting funnel synthesis problem, involving a DLMI for invariance and pointwise in time LMIs for state and input constraints, is an infinite-dimensional problem due to the time-varying nature of the decision variables.

Chapter 4 develops a solution method for the continuous-time funnel synthesis problem formulated in Chapter 3. To make the problem tractable, the DLMI is reformulated using a slack variable, transforming it into a differential linear matrix equality (DLME) and a pointwise-in-time LMI. We refer to this DLME as funnel dynamics, as it behaves analogously to system dynamics in an optimal control framework. This reformulation enables us to cast the funnel synthesis problem into a standard form of numerical optimal control so that we can apply numerical optimal control techniques such as multiple-shooting [16] and control parameterization [72]. Furthermore, the proposed framework provides two ways to aim to ensure continuous-time constraint satisfaction (CTCS). The first enforces

pointwise-in-time LMIs at additional intermediate time points within each subinterval, improving constraint coverage without increasing the number of decision variables. The second approach follows a numerical optimal control technique commonly used in prior works [40, 72] where continuous-time constraints are reformulated by integrating their violations using an exterior penalty function. This reformulation retains convexity but results in constraints that are not in a standard conic form. Thus, the proposed solution method employs a successive convexification (SCvx) approach.

Chapter 5 provides a computational funnel synthesis method for nonlinear systems under bounded disturbances over a finite-time horizon. The main contribution is proving that the computed funnel is indeed invariant over the entire time interval. To achieve this, we first derive an incremental dynamical system representing the behavior relative to the given nominal trajectory, modeled as an uncertain LTV system. The key idea is to approximate this LTV system to an uncertain linear parameter-varying (LPV) system where the approximation error is treated as a state-, input-, and disturbance-dependent uncertainty. This enables the derivation of a finite number of LMIs whose satisfaction guarantees the required continuous-time DLMI for the invariance of the funnel. In this derivation, we employ LMI conditions that ensure matrix copositivity [93]. Including linear state and input constraints together, the funnel synthesis problem can be finally formulated as a finite-dimensional semidefinite program (SDP). Since both the system's nonlinearity and LPV approximation errors are explicitly accounted for, the resulting funnel is guaranteed to be invariant for the original nonlinear system.

Chapter 6 proposes a joint synthesis algorithm that jointly synthesizes the nominal trajectory and the funnel ensuring robustness for nonlinear systems with locally Lipschitz nonlinearities. The problem formulation can be viewed as a robust trajectory optimization in which we optimize both the trajectory and the funnel that consists of the forward invariant set and the corresponding feedback gain. The proposed method has the following steps in each iteration: First, we update the nominal trajectory while ensuring the feasibility of the funnel. The next step estimates local Lipschitz constants of the nonlinearity in the system by sampling state space inside the funnel. With the trajectory computed in the first step, the third step then constructs a semidefinite programming (SDP) problem derived

with funnel constraints and a Lyapunov condition to ensure the invariance property of the funnel. These steps are repeated until the convergence of both the trajectory and the funnel synthesis. We validate the proposed method through a numerical simulation.

Chapter 7 further develops the joint synthesis framework by formulating the combined trajectory and funnel design as an augmented optimal control problem. To solve this nonconvex formulation, a SCvx method based on the penalized trust-region (PTR) approach is applied, whereby each iteration solves a convex SDP.

1.1 Literature Review

1.1.1 Analysis and stabilization of uncertain nonlinear systems

The computation of the funnel proposed in this paper relies on Lyapunov theory. The Lyapunov theory was originally developed by Russian Scientist for system analysis and design [71]. The theory later gained widespread recognition in the Western publications, such as an article by Kalman and Bertram [50]. The ideas in Lyapunov theory have played a significant role in analyzing and designing control systems with uncertainty [30, 48, 54, 104, 105]. Based on Lyapunov theory, we can find an invariant set as a sublevel set of a Lyapunov function, provided the function satisfies certain properties.

The main difficulty in applying Lyapunov theory is that one should find a *Lyapunov function* satisfying certain properties to prove the stability. However, there is no general procedure to obtain such functions for systems. One approach is to assume a specific function based on mathematical intuition and understanding of the system, and then use trial and error to verify whether it satisfies the given properties. Alternatively, one can derive insight from the physics of the system, such as using the energy function as a starting point in mechanical systems.

The proposed funnel synthesis in this thesis considers a specific class of Lyapunov functions, namely quadratic Lyapunov functions, which are quadratic forms in the system state variables. The quadratic Lyapunov function is motivated by the concept of quadratic stability [29, 31, 32, 55, 68, 138]. The necessary and sufficient conditions for the quadratic stability of uncertain systems are examined in [14]. This thesis is particularly related to the

study of necessary and sufficient LMI conditions for the quadratic stability through the use of multiplier matrices [5].

The funnel is defined as a time-varying set around the nominal trajectory. It is essential to characterize the system's behavior relative to the nominal trajectory rather than its behavior relative to a fixed equilibrium point. Hence, this thesis is related to the concept of incremental stability that concerns the convergence between arbitrary trajectories of the system rather than convergence to a fixed point or nominal trajectory. The work in [73] studied incremental stability based on the theory of contraction analysis for nonlinear systems. The existence of an equilibrium state for autonomous systems was investigated in [2] through the quadratic stability of the associated derivative system, which is equivalent to the incremental quadratic stability of the original system. The paper shows that if the derivative system is quadratically stable, then the original system has an equilibrium state that is globally exponentially stable. The work in [9] analyzed incremental stability for autonomous systems in the context of input-to-state stability. They showed a result on the existence of globally attractive solutions for input-to-state systems subject to constant or periodic inputs. Incremental quadratic stability was further investigated in [33], where the concept of an incremental multiplier matrix is provided and used to derive necessary and sufficient LMI conditions for the incremental stability of systems. This approach was utilized in [6] for observer design, and further extended to the simultaneous design of both observer and controller in [127].

The use of LMIs has gained significant attention as a systematic method for constructing Lyapunov functions. In this approach, a Lyapunov function, typically of a specific form such as a quadratic form, can be obtained by solving LMIs. With the development of efficient interior-point methods [89], LMIs can now be solved in polynomial time, which has further increased the popularity of this technique. As Stephen Boyd famously stated in [19], the "grandfather" of this field is Lyapunov, and the "father" is Yakubovich. Many of the early works in this area were developed based on Yakubovich's foundational contributions [128, 129, 130, 132]. This thesis specifically adopts LMI-based approaches incorporating the concept of multipliers [11, 12], as well as their extension to controller synthesis [2, 3]. More recently, sums-of-squares (SOS) methods have been actively studied as a means to

construct Lyapunov functions beyond the quadratic form [94, 122].

1.1.2 Integral quadratic constraints

Integral quadratic constraints (IQCs) provide a unified and powerful framework for robustness analysis of systems with nonlinearities, time variations, and uncertain parameters. Originating from the work of Yakubovich [131] and influenced by both Western and Russian control theory traditions, IQCs generalize classical concepts such as passivity and dissipativity [85, 86]. The primary strengths of the IQC framework are: i) its ability to encompass classical passivity and dissipativity arguments, ii) its simplification of the use of multipliers, which are a major hurdle in μ-analysis [37], and iii) its support for computational tools through convex optimization using LMIs.

The integral constraints employed in the IQC framework can be interpreted in either the frequency-domain or the time-domain. For the time-domain analysis, the work in [86] introduces the notions of hard and soft IQCs. A hard IQC requires the integral constraint to hold over all finite time intervals, while a soft IQC requires the integral constraint to hold over the infinite horizon and may not hold on finite intervals. The work in [109] unifies IQC analysis and the dissipativity theory [125, 126] by constructing a quadratic storage function for hard-IQC multipliers and, via a strict-positivity factorization, extending the same finite-horizon dissipation inequality to soft IQCs. Building on this approach, [107] demonstrates that, with a suitable hard factorization of the IQC multiplier, the finite-horizon dissipation-inequality test is exactly equivalent to the classical frequency-domain IQC stability criterion. A comprehensive overview of the idea and examples of the connection between the IQC framework and dissipativity theory can be found in [106] and references therein. Additionally, in recent years the IQC framework has been extended to exponential stability analysis [17, 47].

Among IQC-related papers, the work most closely related to the analysis and synthesis methods developed in this thesis focuses on uncertain LTV systems [22, 107, 108, 110]. The approach in [108] presents a computational framework for assessing finite-horizon robust performance of uncertain LTV systems with uncertainties described by IQCs. Building upon

this, [22] extends the results to the synthesis of robust output-feedback controllers where Lyapunov parameters and control parameters are iteratively updated. The key distinction with the thesis's focus on funnel synthesis is invariant set computation, which necessitates explicit consideration of state and input constraints. In addition, we employ state-feedback and formulate the problem as a convex SDP where both funnel and control parameters are computed together.

1.1.3 Trajectory optimization by sequential convex programming

Trajectory optimization methods have been developed to solve (possibly) nonconvex trajectory generation problems, e.g., optimal control problems with nonlinear dynamics and nonconvex constraints [72, 77]. Many different algorithms for trajectory optimization have been developed, such as sequential convex programming (SCP) [18, 51, 60, 116], differential dynamics programming (DDP) [62, 118, 119], and indirect methods based on Pontryagin's maximum principle (PMP) [56, 114]. With these developed methods, numerous studies have applied trajectory optimization to various applications, including the control of ground vehicles [61, 92, 134], autonomous robotic manipulation [64], and aerospace vehicles [60, 116].

This thesis is particularly related to the class of nonconvex optimization methods known as SCP. In particular, it builds upon a variant called SCvx that has been developed primarily for trajectory optimization of aerospace vehicles. SCvx methods solve a sequence of convex subproblems around a nonlinear reference trajectory, incorporating mechanisms to ensure convergence despite nonconvexity in dynamics or constraints. SCvx algorithms can be broadly categorized into two classes based on how they manage trust regions: one enforces an explicit constraint on the trust region radius [79, 80, 81], while the other incorporates the radius into the cost function via a penalty term [84, 98, 117]. The latter is often referred to as the PTR method. In recent studies [40], this PTR formulation has been shown to align structurally with the prox-linear method [39], a general framework for solving composite optimization problems with convergence guarantees.

1.1.4 Controlled invariant funnel computation

The computation of funnels has been studied in both robotics and control communities. To the best of our knowledge, the notion of the funnel was first introduced in [82] where the term was used as a geometric metaphor to describe how mechanical interactions can guide a range of initial states toward a desired final configuration. The Lyapunov-theoretic interpretation where funnels represent invariant sets was later adopted in [24]. Following this direction, a significant body of work has focused on offline synthesis of funnels as time-varying controlled invariant sets. Recent approaches rely on sum-of-squares (SOS) programming, where the system dynamics are approximated by polynomials, and high-order polynomial Lyapunov functions are employed [74, 120, 121]. This direction of research was later extended to the computation of the funnel for piecewise polynomial systems in [49]. To improve computational efficiency, the work in [113] formulates an optimization problem that establishes the funnel using linear programming (LP), which is significantly less computationally demanding than SOS programming.

The concept of *funnel synthesis* was first introduced in [100] and later extended in [97]. They employ the quadratic Lyapunov function that is optimized by LMIs to guarantee the invariance and the feasibility of the funnel. Although the quadratic Lyapunov function is simpler than those considered in [74, 120, 121], the distinctive point in [97] compared to existing studies is that they impose the constraints on the funnel for the safety of the systems.

The existing studies in funnel synthesis can be separated into two categories depending on whether they aim to maximize [41, 97, 121] or minimize the size of the funnel [74]. The funnel computation inherently aims to maximize the size of the funnel to have a larger controlled invariant set in the state space. On the other hand, when it comes to systems under uncertainty or disturbances, the funnel has been computed in a way that minimizes the size of the funnel to bound the effect of the uncertainty. For example, the work in [74] minimizes the size of the funnel to prohibit collision with obstacles instead of imposing obstacle avoidance constraints directly. However, minimizing the size of the funnel is against the original purpose of having a large controlled invariant set in the state space. A recent

approach in [59] aims to balance maximizing the size of the funnel and minimizing the effect of the bounded disturbance. To this end, they exploited invariance and attractivity conditions derived from Lyapunov theory and imposing state and input constraints directly on the funnel.

1.1.5 Tube-based predictive control

In the control community, controlled invariant funnels, also referred to as *tubes*, have been studied extensively in the context of tube-based offline finite-horizon predictive control and online MPC. Earlier work in this area focused on linear systems subject to disturbances and stability around an equilibrium point [65, 83, 95]. As the field evolved, approaches have extended toward trajectory tracking for nonlinear systems [4]. In these problems, the main solution variables include the nominal trajectory, the invariant set (funnel), and the feedback controller, typically leading to nonconvex formulations. For tractability in MPC settings, some of these variables are fixed in advance and assumed to be time-invariant for simplicity. For example, a tube-based MPC scheme is developed for Lipschitz nonlinear systems subject to bounded disturbances in [135]. With the precomputed feedback gain and the tube set, the work optimizes the nominal trajectory online. The work in [63] computes an incremental Lyapunov function and a corresponding feedback gain offline, and then optimizes the nominal trajectory and the support value of the invariant set online. In [124], they obtain both the nominal trajectory and the invariant set by solving an SDP with the precomputed feedback gain.

The work in [42] proposes a robust tube-based MPC algorithm for obstacle avoidance. They provide a framework that optimizes the nominal trajectory and the corresponding funnel based on the derived ellipsoid uncertainty propagation. However, this propagation provides only an approximate guarantee, that is, the resulting funnel might not be invariant under the bounded disturbance. Also, they assume that the feedback controllers are precomputed and fixed. For further examples of obstacle avoidance using MPC, we refer the reader to [34, 102, 111]. The most relevant results in robust MPC literature to the proposed work appear in [87] and [78], which jointly synthesize the nominal trajectory and

the funnel. The major difference compared to the joint synthesis method in this thesis is that the resulting funnels in [87] and [78] are only robust for the linearized closed-loop system (they ignored the higher-order terms due to nonlinearities).

In contrast, several offline approaches jointly optimize the nominal trajectory, the feedback policy, and the associated funnel, leveraging this additional flexibility to improve overall performance [57, 66, 67]. Building on the system level synthesis framework [8], the authors of [66] formulate a method to simultaneously compute a nominal trajectory and a time-varying linear feedback gain for linear time-varying (LTV) systems that capture the error dynamics around the nominal trajectory. This joint formulation allows for tighter integration between planning and control, leading to improved robustness and constraint handling. The approach was further extended in [67] to account for parametric uncertainties in the system dynamics.

1.2 Contributions

In this section, we highlight the key contributions of the thesis, organized by chapters.

- The funnel synthesis problem in Chapter 3 considers a broader class of nonlinear systems by incorporating time-varying δQC. In contrast to previous works limited to Lipschitz nonlinearities [97], the proposed work considers more general sector-bounded nonlinearities. In particular, Chapter 3 demonstrates how L-smooth nonlinearities can be explicitly handled within the δQC framework for reduced conservativeness.

- The proposed work derives a DLMI that implies a dissipation inequality in the form of an input-to-state Lyapunov condition, ensuring the funnel's invariance. This generalizes existing LMI-based robust state-feedback synthesis methods [4] by allowing the nonlinearity to depend explicitly on external bounded disturbances.

- Chapter 4 presents a numerical optimal control-based solution method for solving the continuous-time funnel synthesis problem, in which the DLMI appears as one of the constraints. Unlike existing approaches that are restricted to represent funnels using zeroth-order hold or first-order hold interpolations, the proposed method supports

higher-order representations, including second-order hold. Furthermore, it allows the funnel profile to be aligned with the solution of a Lyapunov differential equation.

- The proposed solution method in Chapter 4 systematically addresses CTCS using two convex approaches. While prior work often increases discretization nodes [74, 121], which leads to more decision variables, the first approach adds intermediate constraint-checking points without increasing variable count. The second reformulates the continuous-time LMI using an integral representation of constraint violation using an exterior penalty function, leading to nondifferentiable constraints involving the maximum eigenvalue function. To solve this, a subgradient-based SCvx method is introduced, along with theoretical support for the existence of suitable subgradients.

- Chapter 5 proposes a funnel synthesis method that can guarantee the continuous-time invariance of the funnel without increasing constraint-checking points or involving subgradients. Furthermore, the method employs a matrix copositivity condition [93] to convert the DLMI into a finite set of LMIs. Compared to existing methods [97], this conversion is less conservative, allowing the computation of more optimal funnels while still satisfying the original DLMI condition.

- Chapter 6 proposes a novel algorithm that jointly synthesizes the nominal trajectory and the funnel for discrete-time nonlinear systems with locally Lipschitz nonlinearities. By optimizing the trajectory and funnel together, the algorithm mitigates conservatism that can arise from treating them separately. Chapter 7 extends the joint synthesis framework to continuous-time systems. It further introduces the use of the prox-linear method [39] supported by its convergence guarantees.

Chapter 2
THEORETICAL BACKGROUND

The theoretical foundations of this thesis lie in quadratic stability, incremental quadratic stability, and numerical optimal control. This chapter briefly covers the main idea of each theory and technique. This chapter does not include any contributions of the author and serves solely as background material. More details could be found in [5, 33, 39, 40, 72, 127] and references therein.

2.1 Quadratic stability

Consider the following nonlinear uncertain systems described by

$$\dot{x}(t) = F(t, x(t), w(t)), \tag{2.1}$$

where $t \geq 0$ is the time variable, $x(t) \in \mathbb{R}^{n_x}$ is the state, and $w(t) \in \mathcal{W} \subset \mathbb{R}^{n_w}$ is the external inputs, such as exogenous disturbance or a reference state, where the set $\mathcal{W} \subseteq \mathbb{R}^{n_w}$ are assumed to be compact. We assume that $F : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is continuous with respect to its second and third arguments.

Definition 2.1. *System in* (2.1) *is quadratically stable about the origin with a decay rate* $\alpha > 0$ *and a Lyapunov matrix* $P = P^\top \succ 0$ *if*

$$x^\top P F(t, x, w) \leq -\alpha x^\top P x, \tag{2.2}$$

for all $t \geq 0$, $x \in \mathbb{R}^{n_x}$, *and* $w \in \mathcal{W}$.

Lemma 2.2. *If a system is quadratically stable, one can readily show that every trajectory of the system* (2.1) *satisfies*

$$\|x(t)\| \leq \beta \|x(t_0)\| e^{-\alpha(t-t_0)} \quad \textit{for all } t \geq t_0,$$

with $\beta = \sqrt{\lambda_{max}(P)/\lambda_{min}(P)}$ *where* $\lambda_{min}(P)$ *and* $\lambda_{max}(P)$ *are the smallest and the maximum eigenvalues of* P, *respectively.*

Proof. Define a Lyapunov function $V(x) = x^\top Px$ where the time argument is dropped. It follows from (2.2) that $\dot{V}(x) \leq -2\alpha V(x)$. This implies that $\frac{\dot{V}(x)}{V(x)} \leq -2\alpha$. By integrating both sides from t_0 to t, we obtain $\ln \frac{V(x(t))}{V(x(t_0))} \leq -2\alpha(t-t_0)$, which is equivalent to $V(x(t)) \leq e^{-2\alpha(t-t_0)}V(x(t_0))$. This inequality leads to the following inequalities:

$$\lambda_{min}(P)\|x(t)\|_2^2 \leq x^\top Px \leq e^{-2\alpha(t-t_0)}V(x(t_0)) \leq e^{-2\alpha(t-t_0)}\lambda_{max}(P)\|x(t_0)\|_2^2.$$

The desired result can be constructed with the above inequalities as follows:

$$\|x(t)\|^2 \leq \beta\|x(t_0)\|e^{-\alpha(t-t_0)}.$$

This result implies that the system is globally uniformly exponentially stable with decay rate α. □

2.1.1 Invariance and attractivity conditions

Lemma 2.3. *Suppose that there exist a decay rate $\alpha > 0$ and a Lyapunov matrix $P = P^\top \succ 0$ such that*

$$x^\top PF(t,x,w) \leq -\alpha x^\top Px \quad \textit{when} \quad x^\top Px \geq \|w\|_2^2, \tag{2.3}$$

for all $x \in \mathbb{R}^{n_x}$, $w \in \mathcal{W}$, and $t \geq t_0$. Then, for every bounded disturbance signal $w(\cdot)$, the ellipsoid $\mathcal{E} \coloneqq \{x \mid x^\top Px \leq \|w(\cdot)\|_\infty^2\}$ where $\|w(\cdot)\|_\infty = \operatorname{ess\,sup}_{t\geq t_0}\|w(t)\|_2$ is invariant and attractive for the system (2.1) in the following sense.

- Invariance: The inclusion $x(t_0) \in \mathcal{E}$ implies $x(t) \in \mathcal{E}$ for all $t \geq t_0$ where $x(\cdot)$ is a solution of the system (2.1).

- Attractivity: For every solution $x(\cdot)$ of the system (2.1), there exists $t_1 \in [t_0, \infty)$ such that $x(t) \in \mathcal{E}$ for all $t \geq t_1$.

Proof. We prove the invariance condition by contradiction. Let V be the Lyapunov function defined as $V(x) = x^\top Px$. Suppose there exists a time $t_2 \geq t_0$ such that $x(t_2) \notin \mathcal{E}$. Notice that the solution $x(t)$ of (2.1) is continuous with the bounded disturbance signal $w(\cdot)$. Therefore, $V(x)$ is also continuous in t. By the intermediate value theorem, there exists a

time $t_1 \in [t_0, t_2)$ such that $V(x(t_1)) = \|w(\cdot)\|_\infty^2$ and $V(x(t)) > \|w(\cdot)\|_\infty^2$ for all $t \in (t_1, t_2]$. It follows from (2.3) that

$$\frac{d}{dt}V(x(t)) \leq -\alpha x(t)^\top P x(t) < 0,$$

for all $t \in (t_1, t_2]$. Now, we can deduce that

$$V(x(t_2)) = V(x(t_1)) + \int_{t_1}^{t_2} \frac{d}{dt}V(x(t))\mathrm{d}t \leq V(x(t_1)) = \|w(\cdot)\|_\infty^2.$$

This implies that $x(t_2) \in \mathcal{E}$, which contradicts our hypothesis. Hence, $x(t) \in \mathcal{E}$ for all $t \geq t_0$.

To show the attractivity, consider the case where the initial condition $x(t_0)$ starts in $\mathcal{E}$. In this case, by the invariant property, $x(t) \in \mathcal{E}$ for all $t \geq t_0$. Hence, the attractivity condition trivially follows. Now, suppose that $x(t_0) \notin \mathcal{E}$. To prove by contradiction, consider that $x(t) \notin \mathcal{E}$ for all $t \geq t_0$. The condition $x(t) \notin \mathcal{E}$ implies that $V(x(t)) > \|w(\cdot)\|_\infty^2$, so $V(x(t)) \geq \|w(t)\|_2^2$ for all $t \geq t_0$. By (2.3), we have $\dot{V}(t) \leq -\alpha V(t)$ for all $t \geq t_0$. This Lyapunov condition implies that $V(t) \leq V(0)e^{-\alpha(t-t_0)}$. Hence, with sufficiently large t such that $t \geq t_0 + \frac{1}{\alpha}(\ln V(0) - \ln\|w(\cdot)\|_\infty^2)$, it follows that $V(t) \leq \|w(\cdot)\|_\infty^2$. This contradicts our hypothesis that $x(t) \notin \mathcal{E}$ for all $t \geq t_0$. Therefore, there exists $t_1 \geq t_0$ such that $x(t_1) \in \mathcal{E}$. Given the previously established invariance property, it follows by the same reasoning that $x(t) \in \mathcal{E}$ for all $t \geq t_1$. □

2.2 Incremental quadratic stability

Definition 2.4. *The system* (2.1) *is incrementally quadratically stable (δQS) with a decay rate* $\alpha > 0$ *and a Lyapunov matrix* $P = P^\top \succ 0$ *if*

$$(x - \tilde{x})P[F(t, x, w) - F(t, \tilde{x}, w)] \leq -\alpha(x - \tilde{x})^\top P(x - \tilde{x}), \tag{2.4}$$

for all $t \geq 0$, $x \in \mathbb{R}^n$, $\tilde{x} \in \mathbb{R}^n$, *and* $w \in \mathcal{W}$.

Lemma 2.5. *Suppose that the system* (2.1) *is δQS with a decay rate* α *and a Lyapunov matrix* $P = P^\top \succ 0$. *Then, if* $x(\cdot)$ *and* $\tilde{x}(\cdot)$ *are any two solutions of system, we have*

$$\|x(t) - \tilde{x}(t)\| \leq \beta\|x(t_0) - \tilde{x}(t_0)\|e^{-\alpha(t-t_0)}, \tag{2.5}$$

for all $t \geq t_0$, *where* $\beta = \sqrt{\lambda_{max}(P)/\lambda_{min}(P)}$.

Proof. When $x(\cdot)$ and $\tilde{x}(\cdot)$ are any two solutions of system (2.1), consider a new variable $\delta(t) = x(t) - \tilde{x}(t)$. From the inequality given in (2.4), there exists a matrix $P = P^\top \succ 0$ and a scalar $\alpha > 0$ such that

$$\delta(t)P\dot{\delta}(t) \leq -\alpha\delta(t)P\delta(t).$$

This represents the quadratic stability of $\delta(t)$ given in (2.2) at the origin. This turns out

$$\|x(t) - \tilde{x}(t)\| \leq \beta\|x(t_0) - \tilde{x}(t_0)\|e^{-\alpha(t-t_0)},$$

for all $t \geq t_0$, where $\beta = \sqrt{\lambda_{max}(P)/\lambda_{min}(P)}$. □

2.2.1 Relationship to quadratic stability of the derivative system

Definition 2.6. *A derivative system corresponding to the system described by* (2.1) *for which* $\frac{\partial F}{\partial x}(t, x, w)$ *exists for all* $t \geq t_0$, $x \in \mathbb{R}^{n_x}$, *and* $w \in \mathcal{W}$ *is defined by*

$$\dot{\eta} = \frac{\partial F}{\partial x}(t, \psi(t), w(t))\eta, \tag{2.6}$$

where $\psi(\cdot)$ *and* $w(\cdot)$ *are any continuous functions mapping* $[0, \infty)$ *into* $\mathbb{R}^{n_x}$ *and* $\mathcal{W}$, *respectively.*

Definition 2.7. *The derivative system* (2.6) *is quadratically stable with a decay rate* $\alpha > 0$ *and a Lyapunov matrix* $P = P^\top \succ 0$ *if for all* $t \geq 0$, $w \in \mathcal{W}$, $\psi \in \mathbb{R}^{n_x}$, *and* $\eta \in \mathbb{R}^{n_x}$, *the following inequality holds:*

$$\eta^\top P\frac{\partial F}{\partial x}(t, \psi, w)\eta \leq -\alpha\eta^\top P\eta. \tag{2.7}$$

Lemma 2.8. *Consider the system in* (2.1) *and suppose* F *is differentiable with respect to its second argument. Then, the system is* δQS *with a decay rate* α *and Lyapunov matrix* $P = P^\top \succ 0$ *if and only if the corresponding derivative system* (2.6) *is quadratically stable.*

Proof. We first show that if the corresponding derivative system is quadratically stable, then the original system is incrementally quadratically stable. Suppose that the system is described by (2.1) and its derivative system given by (2.6) is quadratically stable in the sense of (2.7). Consider any time $t \geq 0$, any disturbance $w \in \mathcal{W}$, and any two states $x, \tilde{x} \in \mathbb{R}^{n_x}$.

It follows from mean value theorem that for any vector $z \in \mathbb{R}^{n_x}$, there exists $\hat{\psi} \in \mathbb{R}^{n_x}$ such that

$$z^\top F(t,x,w) - z^\top F(t,\tilde{x},w) = z^\top \frac{\partial F}{\partial x}(t,\hat{\psi},w)(x-\tilde{x}).$$

Using the fact that z is arbitrary, consider $z = (x-\tilde{x})P$. Since the derivative system is quadratically stable with a decay rate α and a Lyapunov matrix P, it follows that

$$\begin{aligned}(x-\tilde{x})^\top P[F(t,x,w) - F(t,\tilde{x},w)] &= (x-\tilde{x})^\top P \frac{\partial F}{\partial x}(t,\hat{\psi},w)(x-\tilde{x}),\\ &\leq -\alpha(x-\tilde{x})^\top P(x-\tilde{x}),\end{aligned}$$

where the last inequality comes from (2.7) by taking $\eta = x - \tilde{x}$. By the definition of (2.4), we can conclude that the system is δQS with the rate of convergence α and the Lyapunov matrix P.

Conversely, suppose now that the system (2.1) is incrementally quadratically stable with decay rate α and Lyapunov matrix P. It follows from (2.4) that

$$(x-\tilde{x})^\top P[F(t,x,w) - F(t,\tilde{x},w)] \leq -\alpha(x-\tilde{x})^\top P(x-\tilde{x}),$$

for all $t \geq 0$, $w \in \mathcal{W}$ and $x, \tilde{x} \in \mathbb{R}^n$. Choose any $\tilde{x}, \eta \in \mathbb{R}^{n_x}$ and $\lambda \in \mathbb{R}$ such that $\lambda \neq 0$, and define x as follows:

$$x = \tilde{x} + \lambda\eta.$$

Then, the above inequality can convert to

$$\frac{\eta^\top P[F(t,\tilde{x}+\lambda\eta,w) - F(t,\tilde{x},w)]}{\lambda} \leq -\alpha\eta^\top P\eta.$$

Since λ is arbitrary in $\mathbb{R}\backslash\{0\}$, we can further deduce

$$\begin{aligned}\eta^\top P\frac{\partial F}{\partial x}(t,\tilde{x},w)\eta &= \eta^\top P\left(\lim_{\lambda\to 0}\frac{F(t,\tilde{x}+\lambda\eta,w) - F(t,\tilde{x},w)}{\lambda}\right),\\ &= \lim_{\lambda\to 0}\eta^\top P\frac{F(t,\tilde{x}+\lambda\eta,w) - F(t,\tilde{x},w)}{\lambda},\\ &\leq -\alpha\eta^\top P\eta.\end{aligned}$$

This above inequality holds for all $t \geq 0, w \in \mathcal{W}$ and $\tilde{x}, \eta \in \mathbb{R}^n$, so the derivative system is quadratically stable at the origin by (2.7). □

2.3 Incremental multiplier matrices and an LMI characterization of incremental quadratic stability

Consider the following uncertain nonlinear systems:

$$\dot{x} = Ax + Bp(t, x, w) + g(t, w), \tag{2.8a}$$

$$q = Cx + Dp, \tag{2.8b}$$

$$p = \phi(t, q, w), \tag{2.8c}$$

where $t \geq 0$ is the time variable, $x(t) \in \mathbb{R}^{n_x}$ is the state, and $w(t) \in \mathcal{W} \subset \mathbb{R}^{n_w}$ represents a system disturbance, a reference state, and/or an input. We arrange the system so that all nonlinearities depending on the state x are captured by the variable $p(t, x, w) \in \mathbb{R}^{n_p}$, while all other components independent of the state are grouped into $g(t, w) \in \mathbb{R}^n$. The auxiliary variable $q(t, x, w) \in \mathbb{R}^{n_q}$ is introduced to define an explicit function ϕ that represents the system's nonlinearities. The matrices A, B, C, D are constant and of appropriate dimensions.

Definition 2.9. *A symmetric matrix $M \in \mathbb{R}^{(n_q+n_p)\times(n_q+n_p)}$ is an incremental multiplier matrix for ϕ if and only if the following incremental quadratic inequality holds:*

$$\begin{bmatrix} q - \tilde{q} \\ \phi(t, q, w) - \phi(t, \tilde{q}, w) \end{bmatrix}^\top M \begin{bmatrix} q - \tilde{q} \\ \phi(t, q, w) - \phi(t, \tilde{q}, w) \end{bmatrix} \geq 0, \tag{2.9}$$

where $t \geq 0$, $w \in \mathcal{W}$ and $q, \tilde{q} \in \mathbb{R}^{n_q}$.

The incremental quadratic inequality in (2.9) is also referred to as an incremental quadratic constraint [46, 88, 127].

Lemma 2.10. *For the system described by (2.8), if there exists a matrix $P = P^\top \succ 0$, a scalar $\alpha > 0$, and an incremental multiplier matrix $M = M^\top$ for ϕ such that*

$$\begin{bmatrix} PA + A^\top P + 2\alpha P & PB \\ B^\top P & 0 \end{bmatrix} + \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^\top M \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \leq 0, \tag{2.10}$$

then the system is δQS with a decay rate α and a Lyapunov matrix P.

Proof. Apply pre- and post-multiplication of the LMI in (2.10) by $[(x-\tilde{x})^\top, (p-\tilde{p})^\top]$ and its transpose, respectively, where $p = \phi(t,q,w)$ and $\tilde{p} = \phi(t,\tilde{q},w)$. This yields:

$$\mathcal{L}_0(x,\tilde{x},w) + \mathcal{L}_1(x,\tilde{x},w) \leq 0,$$

where

$$\mathcal{L}_0(x,\tilde{x},w) = \begin{bmatrix} x-\tilde{x} \\ p-\tilde{p} \end{bmatrix}^\top \begin{bmatrix} PA + A^\top P + 2\alpha P & PB \\ B^\top P & 0 \end{bmatrix} \begin{bmatrix} x-\tilde{x} \\ p-\tilde{p} \end{bmatrix},$$

and

$$\mathcal{L}_1(x,\tilde{x},w) = \begin{bmatrix} x-\tilde{x} \\ p-\tilde{p} \end{bmatrix}^\top \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^\top M \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \begin{bmatrix} x-\tilde{x} \\ p-\tilde{p} \end{bmatrix}.$$

By the definition of the incremental multiplier matrix M in (2.9), we have $\mathcal{L}_1(x,\tilde{x},w) \geq 0$ for all $x,\tilde{x} \in \mathbb{R}^{n_x}$ and $w \in \mathcal{W}$. Therefore, it must be that $\mathcal{L}_0(x,\tilde{x},w) \leq 0$ for all $x,\tilde{x} \in \mathbb{R}^n$ and $w \in \mathcal{W}$, which implies

$$(x-\tilde{x})^\top P[A(x-\tilde{x}) + B(p-\tilde{p})] \leq -\alpha(x-\tilde{x})^\top P(x-\tilde{x}).$$

It follows then from (2.4) that the system (2.8) is δQS with a decay rate α and a Lyapunov matrix P. □

Example 2.11. *(Lipschitz continuity) A function $\phi(t,q,w)$ is said to be globally Lipschitz continuous with respect to its second argument if there exists a constant $\gamma > 0$ such that*

$$\|\phi(t,q,w) - \phi(t,\tilde{q},w)\|_2 \leq \|q-\tilde{q}\|_2,$$

for all $q,\tilde{q} \in \mathbb{R}^{n_q}$. This condition can be expressed in the form of (2.9) *with the multiplier*

$$M = \begin{bmatrix} \gamma^2 I & 0 \\ 0 & -I \end{bmatrix}.$$

Example 2.12. *(Incrementally sector-bounded nonlinearity) Let $p = \phi(t,q,w)$ and $\tilde{p} = \phi(t,\tilde{q},w)$. Define the incremental variables $\delta p := p - \tilde{p}$ and $\delta q := q - \tilde{q}$. The*

nonlinearity/uncertainty $\phi(t, q, w)$ is said to be incrementally sector-bounded with respect to its second argument if it satisfies

$$(\delta p - K_1 \delta q)^\top S^{-1} (\delta p - K_2 \delta q) \leq 0$$

for all $q, \tilde{q} \in \mathbb{R}^{n_q}$, where $S = S^\top$ and K_1 and K_2 are given matrices. This condition describes a nonlinearity $\phi(t, q, w)$ whose incremental behavior lies within the sector defined by K_1 and K_2 with respect to its second argument.

This condition can be represented in the form of (2.9) *using the multiplier*

$$M = \begin{bmatrix} -K_1(t)^\top S^{-1} K_2(t) - K_2(t)^\top S^{-1} K_1(t) & \star \\ S^{-1}(K_1(t) + K_2(t)) & -2S^{-1} \end{bmatrix},$$

where $\star$ represents the transpose of the corresponding off-diagonal block, i.e., the transpose of $S^{-1}(K_1(t) + K_2(t))$.

See [6, 33] for other examples of nonlinearities and uncertainties that can be characterized by incrementally quadratic inequality.

2.3.1 Block diagonal parameterization

This subsection considers a block diagonal parameterization of incremental multiplier matrices [1, 127]. This technique imposes a structural assumption on the multiplier matrices M. By doing so, it facilitates the derivation of an LMI formulation for robust controller synthesis in which both the Lyapunov matrix and the feedback gain are treated as decision variables. to derive an LMI for robust controller synthesis. This subsection considers a block diagonal parameterization of incremental multiplier matrices [1, 127]. This technique imposes a structural assumption on the multiplier matrices M. By doing so, it facilitates the derivation of an LMI formulation for robust controller synthesis in which both the Lyapunov matrix and the feedback gain are treated as decision variables.

Assume that there exists a set $\mathcal{N}$ of matrix pairs (N_1, N_2), where $N_1 \in \mathbb{R}^{n_q \times n_q}$, $N_2 \in \mathbb{R}^{n_p \times n_p}$, and an invertible matrix $T \in \mathbb{R}^{(n_q+n_p)\times(n_q+n_p)}$ with the block structure

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix},$$

and T_{22} is invertible, such that every valid incremental multiplier matrix M satisfies

$$M = T^\top N T, \text{ with } N = \begin{bmatrix} N_1 & 0 \\ 0 & -N_2 \end{bmatrix}.$$

Example 2.13. *The multiplier matrix M for the global Lipschitz nonlinearity described in Example 2.11 satisfies the above assumption with*

$$T = \begin{bmatrix} \gamma^2 I & 0 \\ 0 & -I \end{bmatrix}, \quad \mathcal{N} = \{(\lambda I, \lambda I) \mid \lambda > 0\}.$$

For the sector-bounded nonlinearity illustrated in Example 2.12, the multiplier matrix M satisfies the assumption with

$$T = \begin{bmatrix} K_2 - K_1 & 0 \\ K_2 + K_1 & -2I \end{bmatrix}, \quad \mathcal{N} = \{(\lambda I, \lambda I) \mid \lambda > 0\}.$$

2.4 Successive convexification by a prox-linear method

This section briefly summarizes a specific variant of the SCvx method, known as the PTR approach [96]. In PTR, the trust region is not enforced as a hard constraint but is instead incorporated into the objective function as a penalty term. This soft penalization indirectly bounds the step size, guiding the solution toward a stationary point. The PTR method can also be interpreted as an instance of the prox-linear method framework [39].

The PTR method considers the following discrete-time optimal control problem:

$$\min_{X,U} J_f(x_N) + \sum_{k=0}^{N-1} J_r(x_k, u_k) \tag{2.11a}$$

$$\text{s.t. } x_{k+1} = f(x_k, u_k), \quad k = \{1, \ldots, N-1\}, \tag{2.11b}$$

$$g(x_k, u_k) \leq 0, \quad k = \{1, \ldots, N-1\}, \tag{2.11c}$$

$$h(x_k, u_k) = 0, \quad k = \{1, \ldots, N-1\}. \tag{2.11d}$$

Here, the state and input sets X and U are defined as

$$X = \{x_1, \ldots, x_N\}, \quad U = \{u_1, \ldots, u_{N-1}\},$$

with $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$ for each $k \in \{1, \ldots, N\}$. The system dynamics is described by a continuously differentiable function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$. The inequality and equality constraints are given by $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_g}$ and $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_h}$, respectively, and are enforced elementwise as written in (2.11d) and (2.11c). The objective consists of the terminal cost $J_f : \mathbb{R}^{n_x} \to \mathbb{R}$ and the running cost $J_r : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$. For simplicity and to avoid unnecessary complexity in this chapter, the terminal condition is assumed to be indirectly enforced through the terminal cost.

The PTR method transforms the constrained optimal control problem in (2.11) into an unconstrained optimization problem by penalizing the constraints with tunable hyperparameters. In particular, it employs the $\mathcal{L}_1$ penalty functions as described in [90, Eq. 17.22]. The resulting unconstrained problem is

$$\begin{aligned}\min_{X,U}\ & J_f(x_N) + \sum_{k=0}^{N-1} J(x_k, u_k) \\ & + w_{vc} \sum_{k=0}^{N-1} \left(\|x_{k+1} - f(x_k, u_k)\|_1 + \max(0, g(x_k, u_k)) + \|h(x_k, u_k)\|_1 \right), \end{aligned} \tag{2.12}$$

where the maximum is applied elementwise. Here, $w_{vc} > 0$ is a penalty weight that governs the degree to which constraint violations are penalized.

To solve the possibly nonconvex unconstrained problem (2.12), the PTR method iteratively updates the following convex subproblem and uses its solution to refine trajectory:

$$\begin{aligned}\min_{X,U}\ & J_f(x_N, u_N) + \sum_{k=0}^{N-1} J(x_k, u_k) + w_{vc} \sum_{k=0}^{N-1} \left(\|v_k\|_1 + \max(0, w_k) + \|r_k\|_1 \right) \\ & + w_{tr} \sum_{k=0}^{N} \left(\|x_k - \bar{x}_k\|_2^2 + \|u_k - \bar{u}_k\|_2^2 \right) && \text{(2.13a)} \\ \text{s.t. } & x_{k+1} = A_k x_k + B_k u_k + z_k + v_k, && \text{(2.13b)} \\ & C_k^g x_k + D_k^g u_k + e_k^g \leq w_k, && \text{(2.13c)} \\ & C_k^h x_k + D_k^h u_k + e_k^h = r_k. && \text{(2.13d)}\end{aligned}$$

Here, $(\bar{x}_k, \bar{u}_k)$ denotes the reference trajectory at iteration k, which is either provided as an initial guess or obtained from the solution of the previous iteration. The matrices $A_k =$

$\frac{\partial f}{\partial x}(\bar{x}_k, \bar{u}_k)$ and $B_k = \frac{\partial f}{\partial u}(\bar{x}_k, \bar{u}_k)$ are the Jacobians of the system dynamics, and the vector $z_k = f(\bar{x}_k, \bar{u}_k) - A_k\bar{x}_k - B_k\bar{u}_k$ ensures consistency with the linearization. Similarly, the constraint functions g and h are linearized around the reference trajectory to yield the matrices C_k^g, D_k^g, e_k^g and C_k^h, D_k^h, e_k^h, respectively. The slack variables v_k, w_k, and r_k represent the violations of the dynamics, inequality constraints, and equality constraints, and are penalized in the cost using the weight w_{vc}. The quadratic term weighted by w_{tr} penalizes deviation from the reference, representing the penalized trust region.

The PTR approach aligns closely with the prox-linear method. Specifically, the formulation of the convex subproblem in PTR mirrors that of prox-linear methods applied to composite optimization problems. Since the prox-linear method comes with theoretical convergence guarantees, the same guarantees carry over to PTR under appropriate conditions. The following subsection briefly summarizes the prox-linear algorithm and its convergence properties, providing a foundation for understanding the theoretical justification behind PTR.

2.4.1 Prox-linear algorithm

The prox-linear method is designed to solve

$$\Theta(y) = J(y) + H(G(y)),$$

where $y \in \mathbb{R}^{n_y}$ represents the decision variable. Here, $H : \mathbb{R}^{n_G} \to \mathbb{R}$ is a convex and α_h-Lipschitz continuous function, and $G : \mathbb{R}^{n_y} \to \mathbb{R}^{n_G}$ is potentially nonconvex continuously differentiable with a β_g-Lipschitz continuous gradient. To optimize $\Theta(y)$, the prox-linear method iteratively minimizes a convex approximation of $\Theta(y)$, given by

$$\Theta^\rho(y; y_i) = J(y) + H(G(y_i) + \nabla G(y_i)(y - y_i)) + \frac{1}{2\rho}\|y - y_i\|_2^2,$$

where the subscript i denotes the current iteration index.

The following result describes the descent property of the prox-linear method:

Lemma 2.14. *(Lemma 5.1 of [38]) At iteration i of the prox-linear method, the following holds*

$$\Theta(y_i) \geq \Theta(y_{i+1}) + \frac{\rho}{2}(2 - \alpha_h\beta_g\rho)\|\mathcal{G}_\rho(y_i)\|^2,$$

where $\mathcal{G}_\rho(\cdot)$ is the prox-gradient mapping defined as

$$\mathcal{G}_\rho(y) = \frac{1}{\rho}\left(z - \operatorname*{argmin}_{y'} \Theta^\rho(y'; y)\right).$$

Hence, $\Theta^\rho(y)$ is monotonically decreasing for $\rho \leq \frac{1}{\alpha_h \beta_g}$.

Chapter 3

CONTINUOUS-TIME CONSTRAINED FUNNEL SYNTHESIS

This chapter introduces a generalized formulation of the funnel synthesis problem using the framework of incremental quadratic constraints (δQCs). The formulation accommodates a broader class of uncertainties and nonlinearities beyond standard Lipschitz continuity, including sector-bounded and L-smooth functions. Funnel invariance is enforced through a differential linear matrix inequality (DLMI) derived via dissipativity theory, specifically input-to-state Lyapunov methods. The constraint is imposed through pointwise-in-time linear matrix inequalities (LMIs), leading to a convex optimization problem in terms of the Lyapunov matrix and feedback controller. This chapter establishes the theoretical foundation for continuous-time funnel synthesis.

Chapter-specific notation. This chapter denotes by $\mathbb{R}$ the set of real numbers, $\mathbb{R}_+$ the set of nonnegative real numbers, $\mathbb{R}_{++}$ the set of positive real numbers, $\mathbb{Z}$ the set of integers, and $\mathbb{R}^n$ the set of n-dimensional real column vectors. The set $\mathbb{Z}_{[a,b)} = \{z \in \mathbb{Z} : a \leq z < b\}$ denotes the integers in the half-closed interval from a to b. The set $\mathbb{S}^n$ denotes the set of $n \times n$ real symmetric matrices, $\mathbb{S}^n_+$ the set of $n \times n$ positive semidefinite (PSD) matrices, $\mathbb{S}^n_{++}$ the set of $n \times n$ PD matrices, and $\mathbb{S}^n$ the set of $n \times n$ real symmetric matrices. The set $\mathcal{L}_2[a,b]$ denotes the space of Lebesgue measurable functions $x(t)$ defined on an interval $[a,b] \subset \mathbb{R}$ such that $\left(\int_a^b x(t)^\top x(t) \mathrm{d}t\right)^{1/2} < \infty$. A property is said to hold almost everywhere (also expressed as holding for almost every $x \in X$, or almost all $x \in X$) if the set of points in a measurable space X where it fails has Lebesgue measure zero. For a measurable function $f : [t_0, t_f] \to \mathbb{R}$, the essential supremum of f, denoted by $\operatorname{ess\,sup}_{t\in[t_0,t_f]} f(t)$, is the smallest real number M such that $f(t) \leq M$ for almost all $t \in [t_0, t_f]$. For notational brevity, we omit repeated symmetric terms and write expressions such as $(\star)^\top PA$ to indicate $A^\top PA$. In block matrices, we write $\begin{bmatrix} a & \star \\ b & c \end{bmatrix}$ where $\star$ represents the transpose of the corresponding

off-diagonal block (e.g., $b^\top$). We use $\oplus$ to denote Minkowski sum and $\times$ to denote the Cartesian product of sets. The zero matrices having $m \times n$ size and identity matrix having $n \times n$ size are denoted by $0_{m\times n}$ and I_n, respectively. The subscript will be omitted when it is clear from the context. This chapter uses $\mathrm{diag}(\cdot)$ to denote a diagonal matrix formed from its arguments, $\mathrm{vec}(\cdot)$ to denote vectorization of a matrix by stacking its columns, and $\mathrm{mat}(\cdot)$ as the inverse operation of $\mathrm{vec}(\cdot)$, reshaping a vector into a matrix. For time-varying signals $a(t)$ and $b(t)$, we write (a, b) to denote the pair of functions $\{a(\cdot), b(\cdot)\}$. The chapter omits the time variable t when it is either clear from context or not essential to the discussion.

3.1 Nonlinear Systems, Funnels, and Invariance Conditions

3.1.1 Nonlinear systems

Consider continuous-time nonlinear dynamical systems of the form

$$\dot{x}(t) = f(t, x(t), u(t), w(t)), \quad t \in [t_0, t_f], \tag{3.1}$$

where $f : [t_0, t_f] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ is assumed to be continuous in t and continuously differentiable in other arguments. The vectors $x(t) \in \mathbb{R}^{n_x}$ and $u(t) \in \mathbb{R}^{n_u}$ are the state and the control input, and $w(t) \in \mathbb{R}^{n_w}$ is the unknown, but bounded exogenous disturbance. We assume that $u(\cdot) \in \mathcal{L}_2^{n_u}[t_0, t_f]$ is piecewise continuous, and $w(\cdot) \in \mathcal{L}_2^{n_w}[t_0, t_f]$ is continuous almost everywhere and essentially bounded, that is,

$$\|w(\cdot)\|_\infty \coloneqq \operatorname*{ess\,sup}_{t\in[t_0,t_f]} \|w(t)\|_2 \leq w_{max}, \tag{3.2}$$

for some $w_{max} \in \mathbb{R}_+$. The time instances t_0 and t_f are the initial and the final time, respectively, with $t_0 \leq t_f < \infty$.

Without loss of generality, we express the model (3.1) in the linear fractional form:

$$f(t, x, u, w) = A_o(t)x(t) + B_o(t)u(t) + F_o(t)w(t) + E\phi(t, q_o), \tag{3.3a}$$

$$q_o(t) = C_o x(t) + D_o u(t) + G_o w(t), \tag{3.3b}$$

where $A_o : [t_0, t_f] \to \mathbb{R}^{n_x \times n_x}, B_o : [t_0, t_f] \to \mathbb{R}^{n_x \times n_u}, F_o : [t_0, t_f] \to \mathbb{R}^{n_x \times n_w}$ are bounded and continuous in time. A pair (q_o, ϕ) represents the system's nonlinearity, where ϕ :

$[t_0, t_f] \times \mathbb{R}^{n_{q_o}} \to \mathbb{R}^{n_\phi}$ is a known nonlinear function with its argument $q_o(t) \in \mathbb{R}^{n_{q_o}}$ that is a linear function of x, u, and w. The matrices C_o, D_o, G_o, and E are selector matrices composed of 0s and 1s that are set to organize q_o. It is worth noting that the linear fractional form (3.3) is general enough to illustrate (3.1). This is because simply selecting $E = I$, $\phi = f$, $q_o = [x^\top, u^\top, w^\top]^\top$ and setting A_o, B_o, and C_o to zero matrices of appropriate sizes recover the original system representation (3.1).

The nominal trajectory refers to a collection of state and open-loop control input trajectories $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$, compactly denoted by $(\bar{x}, \bar{u})$, satisfying the original system dynamics, that is,

$$\dot{\bar{x}}(t) = f(t, \bar{x}(t), \bar{u}(t), 0),$$

where the nominal disturbance $\bar{w}(t)$ is set as the zero vector for all t.

Now, we focus on the incremental behavior of the system, that is, how the state evolves relative to a nominal trajectory, also referred to as the difference dynamics. This is essential because the funnel is centered around the nominal trajectory, not the origin. To formalize this, define deviation variables as

$$\eta \coloneqq x - \bar{x}, \quad \xi \coloneqq u - \bar{u}, \quad \delta q_o \coloneqq q_o - \bar{q}_o,$$

where $\bar{q}_o = C_o\bar{x} + D_o\bar{u}$. Substituting into the linear fractional form yields:

$$\dot{\eta}(t) = A_o(t)\eta(t) + B_o(t)\xi(t) + F_o(t)w(t) + E\delta\phi(t, \delta q_o(t)), \tag{3.4a}$$

$$\delta q_o(t) = C_o\eta(t) + D_o\xi(t) + G_o w(t), \tag{3.4b}$$

where $\delta\phi(t, \delta q_o) \coloneqq \phi(t, q_o) - \phi(t, \bar{q}_o)$. The expression above shows how the deviation evolves due to input deviations ξ, the disturbance w, and the incremental nonlinearity $\delta\phi$, which captures the difference in the nonlinear term relative to the nominal trajectory. We will further refine the expression (3.4) using the Mean Value Theorem in the subsequent section.

3.1.2 Funnel definition and invariance conditions

We consider a scalar-valued quadratic storage function (Lyapunov function) $V : [t_0, t_f] \times \mathbb{R}^{n_x} \to \mathbb{R}_+$ defined by

$$V(t, \eta(t)) = \eta(t)^\top Q(t)^{-1} \eta(t), \tag{3.5}$$

where $Q : [t_0, t_f] \to \mathbb{S}^{n_x}_{++}$ is piecewise continuous and continuously differentiable. A *state funnel* is defined as the sub-level set of V:

$$\mathcal{E}_\eta(t) = \{\eta \mid \eta^\top Q(t)^{-1} \eta \leq c_Q\}, \tag{3.6}$$

where $c_Q \in \mathbb{R}_+$ is a support value that specifies the level constant of V for $\mathcal{E}_\eta$. We apply a linear time-varying feedback controller for the incremental system (3.4) given by $\xi(t) = K(t)\eta(t)$ with the piecewise continuous feedback gain $K : [t_0, t_f] \to \mathbb{R}^{n_u \times n_x}$. Then, by Lemma A.1, $\eta \in \mathcal{E}_\eta(t)$ implies $K(t)\eta \in \mathcal{E}_\xi(t)$ where the *input funnel* $\mathcal{E}_\xi(t) \subseteq \mathbb{R}^{n_u}$ is defined as

$$\mathcal{E}_\xi(t) = \{(c_Q K(t) Q(t) K(t)^\top)^{\frac{1}{2}} y \mid \|y\|_2 \leq 1, y \in \mathbb{R}^{n_u}\}. \tag{3.7}$$

With the state and input funnels, a full *funnel* is defined as

$$\mathcal{F}(t) = (\{\bar{x}(t)\} \oplus \mathcal{E}_\eta(t)) \times (\{\bar{u}(t)\} \oplus \mathcal{E}_\xi(t)). \tag{3.8}$$

That is, the funnel is a Cartesian product of state and input funnels, shifted to the nominal trajectory.

Definition 3.1. *The funnel defined in* (3.8) *is invariant for the nonlinear system* (3.1) *in a sense that if $x(\cdot)$ is a solution of* (3.1) *with $x(t_0) \in \{\bar{x}(t_0)\} \oplus \mathcal{E}_\eta(t_0)$, the control law*

$$u(t) = \bar{u}(t) + K(t)(x(t) - \bar{x}(t)), \tag{3.9}$$

and $w(\cdot) \in \mathcal{L}_2^{n_w}[t_0, t_f]$ satisfying (3.2)*, then $(x(t), u(t)) \in \mathcal{F}(t)$ for all $t \in [t_0, t_f]$.*

Definition 3.2. *Given a state constraint set $\mathcal{X} \subset \mathbb{R}^{n_x}$ and an input constraint set $\mathcal{U} \subset \mathbb{R}^{n_u}$, the funnel defined in* (3.8) *is feasible if*

$$\mathcal{F}(t) \subseteq \mathcal{X} \times \mathcal{U}, \quad \forall t \in [t_0, t_f]. \tag{3.10}$$

Definition 3.3. *Given a fixed nominal trajectory $(\bar{x}, \bar{u})$, funnel synthesis refers to a procedure for computing the functions $Q : [t_0, t_f] \to \mathbb{S}^{n_x}_{++}$ and $K : [t_0, t_f] \to \mathbb{R}^{n_u \times n_x}$, ensuring the invariance and feasibility of funnel as defined in Definitions 3.1 and 3.2, respectively.*

3.2 Continuous-time Funnel Synthesis Problem

3.2.1 Structured nonlinearity and mean value theorem

Recall that the nonlinear function ϕ in (3.3) is given by

$$\phi(t, q_o) = \begin{bmatrix} \phi_1(t, q_1) \\ \vdots \\ \phi_{n_\phi}(t, q_{n_\phi}) \end{bmatrix},$$

where each $\phi_i : [t_0, t_f] \times \mathbb{R}^{n_{q_i}} \to \mathbb{R}$ corresponds to the i-th row of ϕ, and $q_i \in \mathbb{R}^{n_{q_i}}$ is a subset of the entries of q_o. That is, q_i is not necessarily equal to q_o, but rather represents the specific argument passed to ϕ_i drawn from q_o, since ϕ_i may depend only on part of q_o.

For a more compact representation, we introduce an index $[i]$ for all $i \in \mathbb{Z}_{[1,n_c]}$ to represent the i-th nonlinearity channel, which is a group of component indices treated together in the reformulation. Let $\phi_{[i]}$ denote the stacked function composed of all ϕ_j with $j \in [i]$. The associated argument $q_{[i]} \in \mathbb{R}^{n_{q_{[i]}}}$ is defined as the minimal vector containing all distinct variables that appear as arguments of the functions ϕ_j in channel $[i]$. If two or more ϕ_j in the same channel share the same argument, that argument appears only once in $q_{[i]}$. For example, if ϕ_1 and ϕ_2 are in channel $[1]$ and $q_1 = q_2$, then $q_{[1]} = q_1$ not $[q_1, q_2]^\top$. We define selector matrices C_i, D_i, G_i so that $q_{[i]} = C_i x + D_i u + F_i w$. This reformulation is called structured nonlinearity, whose details can be found in [99, Chapter 6.2.1].

Let $\bar{q}_{[i]} \coloneqq C_{[i]}\bar{x} + D_{[i]}\bar{u}$ be the nominal value of $q_{[i]}$, and define the increment $\delta q_{[i]} = q_{[i]} - \bar{q}_{[i]}$. Applying the Mean Value Theorem to each channel gives

$$\phi_{[i]}(t, q_{[i]}) = \phi_{[i]}(t, \bar{q}_{[i]}(t)) + \phi'_{[i]}(t, \tilde{q}_{[i]}(t))\delta q_{[i]}(t),$$

where $\tilde{q}_{[i]}(t) \in \mathbb{R}^{n_{q_{[i]}}}$ lies on the line segment between $q_{[i]}(t)$ and $\bar{q}_{[i]}(t)$, i.e., $\tilde{q}_{[i]} \in \mathrm{Co}(\{q_{[i]}, \bar{q}_{[i]}\})$, where $\mathrm{Co}(\cdot)$ denotes the convex hull. The Jacobian $\phi'_{[i]} : [t_0, t_f] \times \mathbb{R}^{n_{q_{[i]}}} \to$

$\mathbb{R}^{1\times n_{q_{[i]}}}$ is taken with respect to the second argument. By adding and subtracting the term $\phi'_{[i]}(t, \bar{q}_{[i]}(t))\delta q_{[i]}(t)$ to the right-hand side, we obtain the followings:

$$\phi_{[i]}(t, q_{[i]}) = \phi_{[i]}(t, \bar{q}_{[i]}(t)) + \phi'_{[i]}(t, \bar{q}_{[i]}(t))\delta q_{[i]}(t) + \delta p_{[i]}(t), \tag{3.11a}$$

$$\delta p_{[i]}(t) = \left(\phi'_{[i]}(t, \tilde{q}_{[i]}(t)) - \phi'_{[i]}(t, \bar{q}_{[i]}(t))\right)\delta q_{[i]}(t), \quad i \in \mathbb{Z}_{[1,n_c]}, \tag{3.11b}$$

where the function $\delta p_{[i]}(t) \in \mathbb{R}^{n_{p_{[i]}}}$ and $\sum_{i=1}^{n_c} n_{p_{[i]}} = n_\phi$.

The incremental form of dynamics (3.4) can then be rewritten as

$$\dot{\eta}(t) = A(t)\eta(t) + B(t)\xi(t) + F(t)w(t) + E\delta p(t, \delta q(t)), \tag{3.12a}$$

$$\delta q(t) = C\eta(t) + D\xi(t) + Gw(t), \tag{3.12b}$$

where the vectors $\delta q \in \mathbb{R}^{n_q}$ and $\delta p \in \mathbb{R}^{n_p}$ are the stacked vector of all channel-wise $\delta q_{[i]}$ and $\delta p_{[i]}$, respectively, with total dimension $n_q = \sum_{i=1}^{n_c} n_{q_{[i]}}$ and $n_p = \sum_{i=1}^{n_c} n_{p_{[i]}}$. The matrices C, D, and G are defined by the vertical stacking of individual selector matrices:

$$C = \begin{bmatrix} C_1 \\ \vdots \\ C_{n_c} \end{bmatrix}, \quad D = \begin{bmatrix} D_1 \\ \vdots \\ D_{n_c} \end{bmatrix}, \quad G = \begin{bmatrix} G_1 \\ \vdots \\ G_{n_c} \end{bmatrix}.$$

The time-varying matrices $A(t), B(t)$, and $F(t)$ are given by

$$A(t) = A_o(t) + E\phi'(t)C_o, \tag{3.13a}$$

$$B(t) = B_o(t) + E\phi'(t)D_o, \tag{3.13b}$$

$$F(t) = F_o(t) + E\phi'(t)G_o, \tag{3.13c}$$

where

$$\phi'(t) \coloneqq \begin{bmatrix} \phi'_{[i]}(t, \bar{q}_{[i]}(t)) \\ \vdots \\ \phi'_{[n_c]}(t, \bar{q}_{[n_c]}(t)) \end{bmatrix}.$$

Note that these matrices $A(t), B(t)$, and $F(t)$ are exactly the Jacobian of f evaluated along the nominal trajectory:

$$A(t) = \left.\frac{\partial f}{\partial x}\right|_{x=\bar{x}(t),u=\bar{u}(t),w=0}, B(t) = \left.\frac{\partial f}{\partial u}\right|_{x=\bar{x}(t),u=\bar{u}(t),w=0}, F(t) = \left.\frac{\partial f}{\partial w}\right|_{x=\bar{x}(t),u=\bar{u}(t),w=0}.$$

Example 3.4. *Consider the dynamics of a unicycle model given by*

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} u_1 \cos x_3 \\ u_1 \sin x_3 \\ u_2 \end{bmatrix},$$

where x_1 and x_2 are x- and y-positions, x_3 is the yaw angle, and u_1 and u_2 are the linear and the angular velocities, respectively. This model can be written in the form of (3.4) *using the following matrices:*

$$A_o = 0_{3\times 3}, \quad B_o = \begin{bmatrix} 0_{2\times 2} \\ 0 \quad 1 \end{bmatrix}, \quad E = \begin{bmatrix} I_2 \\ 0_{1\times 2} \end{bmatrix},$$
$$C_o = \begin{bmatrix} & 1 \\ 0_{2\times 2} & \\ & 0 \end{bmatrix}, \quad D_o = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},$$

and $q_o = [x_3, u_1]^\top$. The nonlinear function is given by $\phi = [u_1 \cos x_3, u_1 \sin x_3]^\top$. Choosing channels $\phi_{[1]} = \phi_1$ and $\phi_{[2]} = \phi_2$, we have $q_o = q_{[1]} = q_{[2]}$, $C_1 = C_2 = C_o$, and $D_1 = D_2 = D_o$. The stacked Jacobian evaluated on the nominal trajectory is

$$\phi'(t) = \begin{bmatrix} \phi'_{[1]}(t, \bar{q}_{[1]}) \\ \phi'_{[2]}(t, \bar{q}_{[2]}) \end{bmatrix} = \begin{bmatrix} -\bar{u}_1 \sin \bar{x}_3 & \cos \bar{x}_3 \\ \bar{u}_1 \cos \bar{x}_3 & \sin \bar{x}_3 \end{bmatrix} \tag{3.14}$$

Thus, the time-varying matrices $A(t)$ and $B(t)$ become

$$A(t) = \begin{bmatrix} & -\bar{u}_1 \sin \bar{x}_3 \\ 0_{3\times 2} & \bar{u}_1 \cos \bar{x}_3 \\ & 0 \end{bmatrix}, \quad B(t) = \begin{bmatrix} \cos \bar{x}_3 & 0 \\ \sin \bar{x}_3 & 0 \\ 0 & 1 \end{bmatrix}.$$

3.2.2 Characterization of nonlinearity using δQC

The incremental form of dynamics system expressed in (3.12) can be viewed as an uncertain LTV system. The pair $(\delta q, \delta p)$, originating from the nonlinearity ϕ in the original system (3.1), now appears as an uncertainty in the incremental system. To characterize this uncertainty, we employ the framework of incremental quadratic constraints [5, 33].

Definition 3.5. *Let $\delta p : [t_0, t_f] \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_p}$ be an uncertain nonlinear function for the incremental system* (3.12)*. A symmetric matrix-valued function $M : [t_0, t_f] \to \mathbb{S}^{(n_p+n_q)}$ is called a time-varying incremental multiplier matrix if it satisfies the following δQC:*

$$\begin{bmatrix} \delta q(t) \\ \delta p(t, \delta q(t)) \end{bmatrix}^\top M(t) \begin{bmatrix} \delta q(t) \\ \delta p(t, \delta q(t)) \end{bmatrix} \geq 0, \quad \forall \delta q \in \mathcal{L}_2^{n_{q_o}}[t_0, t_f], \quad t \in [t_0, t_f]. \tag{3.15}$$

We denote by $\mathcal{M}(t) \subset \mathbb{S}^{n_p+n_q}$ the set of all incremental multiplier matrices that satisfy this condition for the given nonlinearity δp at a fixed time $t \in [t_0, t_f]$.

Example 3.6. *Let $K_1, K_2 : [t_0, t_f] \to \mathbb{R}^{n_p \times n_q}$ be two given matrix-valued functions. The nonlinearity δp_o is said to satisfy a time-varying incremental sector bound if, for some invertible and symmetric weight $S = S^\top \in \mathbb{S}^{n_p}$, the following inequality holds:*

$$(\delta p(t) - K_1(t)\delta q(t))^\top S^{-1}(K_2(t)\delta q(t) - \delta p(t)) \geq 0, \quad \forall t \in [t_0, t_f].$$

This condition admits an δQC representation as in (3.15) *with the time-varying multiplier set*

$$\mathcal{M}(t) = \left\{ \lambda(t) \begin{bmatrix} -K_1(t)^\top S^{-1} K_2(t) - K_2(t)^\top S^{-1} K_1(t) & \star \\ S^{-1}(K_1(t) + K_2(t)) & -2S^{-1} \end{bmatrix} \middle| \lambda(t) \in \mathbb{R}_+ \right\}. \tag{3.16}$$

As a special case, if δp satisfied a norm bound $\|\delta p(t)\| \leq \gamma(t)\|\delta q(t)\|$ with $\gamma(t) > 0$, then (3.16) *holds with $K_1(t) = -\gamma(t)I$, $K_2 = \gamma(t)I$, and $S = I$.*

To enable the convexification of the funnel synthesis problem, we adopt a block-diagonal parameterization of the incremental multiplier matrices [4, 127]. We assume that for every $t \in [t_0, t_f]$, there exist matrices $N_1(t) \in \mathbb{S}^{n_q}$, $N_2(t) \in \mathbb{S}^{n_p}$, and a nonsingular matrix $T(t) \in \mathbb{R}^{(n_q+n_p)\times(n_q+n_p)}$ such that

$$T(t)^\top \begin{bmatrix} N_1(t)^{-1} & 0 \\ 0 & -N_2(t)^{-1} \end{bmatrix} T(t) \in \mathcal{M}(t), \tag{3.17}$$

The transformation matrix $T(t)$ is partitioned as

$$T(t) = \begin{bmatrix} T_{11}(t) & T_{12}(t) \\ T_{21}(t) & T_{22}(t) \end{bmatrix},$$

and we assume that $T_{22}(t)$ is nonsingular for all $t \in [t_0, t_f]$. We define the feasible set-valued function $\mathcal{N} : [t_0, t_f] \to \mathbb{S}_{n_q} \times \mathbb{S}_{n_p}$ as the set of all admissible pairs $(N_1(t), N_2(t))$ that satisfy the block-diagonal parameterization condition (3.17).

Example 3.7. *The incremental multiplier matrices defined in Example 3.6 can be expressed in block-diagonal form using*

$$T(t) = \begin{bmatrix} K_2(t) - K_1(t) & 0 \\ K_2(t) + K_1(t) & -2I \end{bmatrix}, \quad \mathcal{N}(t) = \{(\lambda S, \lambda S) \mid \lambda > 0\}. \tag{3.18}$$

3.2.3 Lipschitz and L-smooth nonlinearities

Here we provide the characterization of the uncertain term δp using the δQC (3.15), based on local Lipschitz and L-smooth properties that are two common assumptions for general nonlinear systems.

Suppose that each nonlinear component $\phi_{[i]}$ is locally pointwise Lipschitz. Then, its Jacobian $\phi'_{[i]}$ is bounded over any compact set $\Omega \subset \mathbb{R}^{n_{q_{[i]}}}$. For each $[i] \in \mathbb{Z}_{[1,n_c]}$, there exists a time-varying constant $\gamma_i(t) \in \mathbb{R}_+$ such that

$$\|\phi'_{[i]}(t, q_{[i]}) - \phi'_{[i]}(t, \bar{q}_{[i]})\|_2 \leq \gamma_i(t), \quad \forall q_{[i]}, \bar{q}_{[i]} \in \Omega, \quad \forall t \in [t_0, t_f]. \tag{3.19}$$

If $\phi_{[i]}$ is Lipschitz continuous with constant $\gamma_i/2$, then its Jacobian is bounded by $\gamma_i/2$, and hence the above bound holds with γ_i.

On the other hand, if ϕ is locally L-smooth, then its Jacobian is Lipschitz continuous. That is, for any compact set $\Omega \subset \mathbb{R}^{n_{q_{[i]}}}$, there exists a time-varying constant $\beta_i(t) \in \mathbb{R}_+$ such that for all $i \in \mathbb{Z}_{[1,n_c]}$,

$$\|\phi'_{[i]}(t, q_{[i]}) - \phi'_{[i]}(t, \bar{q}_{[i]})\|_2 \leq \beta_i(t)\|\delta q_{[i]}(t)\|_2, \quad \forall q_{[i]}, \bar{q}_{[i]} \in \Omega, \quad \forall t \in [t_0, t_f]. \tag{3.20}$$

Remark 3.8. *The representation with* (3.20) *presents a direct way to control the level of nonlinearity, whereas the representation with* (3.19) *does not exploit this flexibility. It is obvious that the representation with* (3.19) *has a constant upper bound of the term,* $\|\phi'_{[i]}(t, q_{[i]}) - \phi'_{[i]}(t, \bar{q}_{[i]})\|_2$*, for each* t *whereas the upper bound in* (3.20) *goes to zero as the region around the nominal trajectory shrinks to zero (*$\bar{q}_{[i]} \leftarrow q_{[i]}$*). This implies that*

we can reduce the region, in which we seek for a funnel, arbitrarily to reduce the level of nonlinearity to zero and hence increase the chance of finding a funnel. This is not the case with (3.19) *because the level of nonlinearity may not be reduced no matter how small the region is.*

With the transformation matrix chosen as the identity $T(t) = I$, the valid set $\mathcal{N}(t)$ corresponding to the Lipchitz nonlinearity in (3.19) is given by

$$\begin{aligned}\mathcal{N}(t) = \{(N_1, N_2) : N_1 = \text{blkdiag}(\lambda_1 I_{n_{q_{[1]}}}, \ldots, \lambda_{n_c} I_{n_{q_{[n_c]}}}),\\ N_2 = \text{diag}(\lambda_1 \gamma_1^2(t), \ldots, \lambda_{n_c} \gamma_{n_c}^2(t)),\ \lambda_i \in \mathbb{R}_+,\ i \in \mathbb{Z}_{[1,n_c]}\}.\end{aligned} \tag{3.21}$$

Similarly, for the L-smooth nonlinearity in (3.20), the valid set $\mathcal{N}(t)$ is given by

$$\begin{aligned}\mathcal{N}(t) = \{(N_1, N_2) : N_1 = \text{blkdiag}(\lambda_1 I_{n_{q_{[1]}}}, \ldots, \lambda_{n_c} I_{n_{q_{[n_c]}}}),\\ N_2 = \text{diag}(\lambda_1 s_1(t)\beta_1^2(t), \ldots, \lambda_{n_c} s_{n_c}(t)\beta_{n_c}^2(t)),\\ \|\delta q_{[i]}(t)\|_2^2 \le s_i(t),\ \lambda_i \in \mathbb{R}_+,\ i \in \mathbb{Z}_{[1,n_c]}\},\end{aligned} \tag{3.22}$$

where $s : [t_0, t_f] \to \mathbb{R}_+^{n_c}$ serves as one of the decision variables. To ensure the validity of the multiplier matrix for the L-smooth nonlinearity, each inequality $\|\delta q_{[i]}\|_2^2 \le s_i$ must be satisfied. Assuming $G_i = 0$, it follow that $\|\delta q_{[i]}\|_2 = \|C_i\eta(t) + D_i\xi(t)\|_2$. Therefore, the condition $\|\delta q_{[i]}\|_2^2 \le s_i$ can be equivalently enforced by the following LMI:

$$\begin{bmatrix} s_i(t) I_{n_x} & * \\ (C_i Q(t) + D_i Y(t))^\top & Q(t) \end{bmatrix} \succeq 0. \tag{3.23}$$

Example 3.9. *Recall the Jacobian ϕ' of the nonlinearity ϕ of the unicycle model (see Example 3.4) given in* (3.14)*. Suppose the control input for the velocity is bounded such that $|u_1| \le v_{max}$ for some $v_{max} \in \mathbb{R}_+$. Then, the row-wise norms of ϕ' can be deduced by*

$$\begin{aligned}\|\phi'_{[1]}\|_2 &= \sqrt{u_1^2 \sin^2 x_3^2 + \cos^2 x_3^2} \le \max\{1, v_{max}\},\\ \|\phi'_{[2]}\|_2 &= \sqrt{u_1^2 \cos^2 x_3^2 + \sin^2 x_3^2} \le \max\{1, v_{max}\}.\end{aligned}$$

Since the Jacobian is bounded, each $\phi_{[i]}$ is Lipschitz continuous with constant at most $\max\{1, v_{max}\}$, and thus the corresponding constant γ_i in (3.19) *is $\gamma_i = 2\max\{1, v_{max}\}$*

for $i = 1, 2$. $\frac{\partial}{\partial q_{[1]}} \phi'_{[1]}(t, q_{[1]})$ *In addition, since* ϕ'_i *is differentiable, we can compute:*

$$\frac{\partial \phi'_{[1]}(t, q_{[1]})}{\partial q_{[1]}} = \begin{bmatrix} -u_1 \cos x_3 & -\sin x_3 \\ -\sin x_3 & 0 \end{bmatrix}, \frac{\partial \phi'_{[2]}(t, q_{[2]})}{\partial q_{[2]}} = \begin{bmatrix} -u_1 \sin x_3 & \cos x_3 \\ \cos x_3 & 0 \end{bmatrix}.$$

Taking norms, we find

$$\left|\left|\frac{\partial \phi'_{[1]}}{\partial q_{[1]}}\right|\right|_2 \leq \max\{\sqrt{2}, v_{max}\}, \left|\left|\frac{\partial \phi'_{[2]}}{\partial q_{[2]}}\right|\right|_2 \leq \max\{\sqrt{2}, v_{max}\}.$$

Hence, the valid L-smooth constants β_i *in* (3.20) *are* $\beta_i = \max\{\sqrt{2}, v_{max}\}$ *for* $i = 1, 2$.

3.2.4 Invariance condition by DLMI

We establish funnel invariance using a dissipation inequality based on a Lyapunov function in the following Lemma.

Lemma 3.10. *Let* $Q : [t_0, t_f] \to \mathbb{S}^{n_x}_{++}$, $K : [t_0, t_f] \to \mathbb{R}^{n_u \times n_x}$, $\lambda_w : [t_0, t_f] \to \mathbb{R}_+$ *be piecewise continuous functions. Define the Lyapunov function* $V : [t_0, t_f] \times \mathbb{R}^{n_x} \to \mathbb{R}_+$ *as in* (3.5). *Suppose that all trajectories of the incremental system* (3.12) *under the feedback control* $\xi(t) = K(t)\eta(t)$ *satisfy, for almost all* $t \in [t_0, t_f]$, *the following inequalities:*

$$\dot{V}(t, \eta) + \alpha V(t, \eta) - \lambda_w(t) w(t)^\top w(t) \leq 0, \tag{3.24a}$$

$$0 \leq \lambda_w(t) \leq \alpha, \tag{3.24b}$$

for some decay rate $\alpha \in \mathbb{R}_{++}$. *Then, the state funnel* $\mathcal{E}_\eta(t)$, *defined in* (3.6), *with the support value* $c_Q = w^2_{max}$ *is invariant. That is, if* $\eta(\cdot)$ *is a solution of* (3.12) *with* $\eta(t_0) \in \mathcal{E}_\eta(t_0)$, *then* $\eta(t) \in \mathcal{E}_\eta(t)$ *for all* $t \in [t_0, t_f]$.

Proof. Start with the dissipation inequality (3.24a) multiply both sides by $e^{\alpha(t-t_0)}$:

$$e^{\alpha(t-t_0)} \dot{V}(t, \eta) + \alpha e^{\alpha(t-t_0)} V(t, \eta) \leq \lambda_w(t) e^{\alpha(t-t_0)} w(t)^\top w(t).$$

Noting that the left-hand side is the derivative of $e^{\alpha(t-t_0)} V(t, \eta)$, we write

$$\frac{d}{dt}\left(e^{\alpha(t-t_0)} V(t, \eta)\right) \leq \lambda_w(t) e^{\alpha(t-t_0)} w(t)^\top w(t).$$

Integrate both sides from t_0 to $t \in [t_0, t_f]$:

$$e^{\alpha(t-t_0)}V(t,\eta) - V(t_0,\eta) \leq \int_{t_0}^{t} \lambda_w(\tau)(\tau)e^{\alpha(\tau-t_0)}w(\tau)^\top w(\tau)\mathrm{d}\tau.$$

Using the pointwise bound $\|w\|_2^2 \leq w_{max}^2$ from (3.2) and $\lambda_w(t) \leq \alpha$ from (3.24b), we further bound the integral:

$$\int_{t_0}^{t} \lambda_w(\tau)(\tau)e^{\alpha(\tau-t_0)}w(\tau)^\top w(\tau)\mathrm{d}\tau \leq \alpha w_{max}^2 \int_{t_0}^{t} e^{\alpha(\tau-t_0)}\mathrm{d}\tau = w_{max}^2(e^{\alpha(t-t_0)} - 1).$$

Therefore,

$$e^{\alpha(t-t_0)}V(t,\eta) \leq V(t_0,\eta) + w_{max}^2(e^{\alpha(t-t_0)} - 1).$$

Multiply both sides by $e^{-\alpha(t-t_0)}$:

$$V(t,\eta) \leq e^{-\alpha(t-t_0)}V(t_0,\eta) + (1 - e^{-\alpha(t-t_0)})w_{max}^2.$$

If $\eta(t_0) \in \mathcal{E}_\eta(t_0)$, so $V(t_0,\eta) \leq w_{max}^2$, and we have

$$V(t,\eta) \leq w_{max}^2.$$

Therefore, we can conclude that $\eta(t) \in \mathcal{E}_\eta(t)$ for all $t \in [t_0, t_f]$. □

The following lemma shows the invariance of the state funnel $\mathcal{E}_\eta$ implies the invariance of the full funnel $\mathcal{F}$.

Lemma 3.11. *If the state funnel $\mathcal{E}_\eta(t)$ is invariant for the incremental system* (3.12) *under the linear feedback gain $K(t)$, then the funnel $\mathcal{F}(t)$, defined in* (3.8)*, is invariant for the uncertain nonlinear system* (3.1) *in the sense described in Definition 3.1.*

Proof. Since $x(t_0) \in \{\bar{x}(t_0)\} \oplus \mathcal{E}_\eta(t_0)$, it follows that $\eta(t_0) = x(t_0) - \bar{x}(t_0) \in \mathcal{E}_\eta(t_0)$. By the assumed invariance of the state $\mathcal{E}_\eta(t)$, we have $\eta(t) \in \mathcal{E}_\eta(t)$ for all $t \in [t_0, t_f]$. Under the feedback law $u(t) = \bar{u}(t) + K(t)\eta(t)$, it follows that the input deviation is $\xi(t) = K(t)\eta(t)$, and hence $\xi(t) \in \mathcal{E}_\xi(t)$. Therefore, $(x(t), u(t)) \in \mathcal{F}(t)$ for all $t \in [t_0, t_f]$. □

Now, we derive a sufficient condition for funnel invariance by formulating the dissipation inequality 3.24 as a DLMI. This condition enables the computation of the funnel-shaping matrices $Q(t)$ and feedback gains $K(t)$ using convex optimization techniques.

Theorem 3.12. *Consider the incremental system given in 3.12. Suppose there exist functions* $Q : [t_0, t_f] \to \mathbb{S}^{n_x}_{++}$, $Y : [t_0, t_f] \to \mathbb{R}^{n_u \times n_x}$, $\lambda_w : [t_0, t_f] \to \mathbb{R}_+$, $N_1 : [t_0, t_f] \to \mathbb{S}^{n_q}$, *and* $N_2 : [t_0, t_f] \to \mathbb{S}^{n_p}$, *such that* $(N_1(t), N_2(t)) \in \mathcal{N}(t)$, $0 \leq \lambda_w(t) \leq \alpha$, *and the following DLMI holds for almost all* $t \in [t_0, t_f]$*:*

$$\begin{bmatrix} H_{11}(t) + H_{11}(t)^\top - \dot{Q}(t) & \star \\ H_{12}(t) & H_{22}(t) \end{bmatrix} \preceq 0, \tag{3.25}$$

where the matrix blocks are defined as:

$$H_{11}(t) = A(t)Q(t) + B(t)Y(t) - ET_{22}(t)^{-1}T_{21}(t)(CQ(t) + DY(t)) + \frac{\alpha}{2}Q(t), \tag{3.26a}$$

$$H_{12}(t) = \begin{bmatrix} N_2(t)T_{22}(t)^{-\top}E^\top \\ F(t)^\top - D^\top T_{21}(t)^\top T_{22}(t)^{-T}E^\top \\ \Sigma(t)(CQ(t) + DY(t)) \end{bmatrix} \tag{3.26b}$$

$$H_{22}(t) = \begin{bmatrix} -N_2(t) & \star & \star \\ 0 & -\lambda_w(t)I & \star \\ \Lambda N_2 & \Sigma(t)D & -N_1(t) \end{bmatrix} \tag{3.26c}$$

and the auxiliary matrices are:

$$\Lambda(t) \coloneqq T_{12}(t)T_{22}(t)^{-1}, \quad \Sigma(t) \coloneqq T_{11}(t) - T_{12}(t)T(t)_{22}^{-1}T_{21}(t).$$

Then, with the feedback gain defined as $K(t) = Y(t)Q(t)^{-1}$, *the dissipation inequality* (3.24) *holds almost all* $t \in [t_0, t_f]$ *and the funnel* $\mathcal{F}(t)$ *defined in* (3.8) *is invariant in the sense described in Definition 3.1.*

Proof. Throughout, the time argument t is often omitted whenever clear. Define the closed-loop matrices $A_{cl} = A + BK$ and $C_{cl} = C + DK$. Then the DLMI (3.25) becomes:

$$\begin{bmatrix} Q(A_{cl} - ET_{22}^{-1}T_{21}C_{cl})^\top + (A_{cl} - ET_{22}^{-1}T_{21}C_{cl})Q - \dot{Q} + \alpha Q & \star & \star & \star \\ N_2T_{22}^{-\top}E^\top & -N_2 & \star & \star \\ F^\top - D^\top T_{21}^\top T_{22}^{-T}E^\top & 0 & -\lambda_w I & \star \\ \Sigma C_{cl}Q & \Lambda N_2 & \Sigma D & -N_1 \end{bmatrix} \preceq 0.$$

Multiply on the left and right by $\mathrm{diag}(Q^{-1}, N_2^{-1}, I, I)$ and apply Schur complement:

$$\begin{bmatrix} (A_{cl} - ET_{22}^{-1}T_{21}C_{cl})^\top Q^{-1} + Q^{-1}(A_{cl} - ET_{22}^{-1}T_{21}C_{cl}) - \dot{Q} + \alpha Q & \star & \star \\ T_{22}^{-\top}E^\top & -N_2 & \star \\ F^\top Q^{-1} - D^\top T_{21}^\top T_{22}^{-T} E^\top Q^{-1} & 0 & -\lambda_w I \end{bmatrix} + (\star)^\top N_1^{-1} \begin{bmatrix} \Sigma C_{cl} & \Lambda & \Sigma D \end{bmatrix} \preceq 0.$$

More compactly, we can write:

$$\begin{bmatrix} A_{cl}^\top Q^{-1} + Q^{-1}A_{cl} - Q^{-1}\dot{Q}Q^{-1} + \alpha Q^{-1} - \Theta & \star & \star \\ T_{22}^{-\top}E^\top Q^{-1} & 0 & \star \\ F^\top Q^{-1} - D^\top T_{21}^\top T_{22}^{-T} E^\top Q^{-1} & 0 & -\lambda_w I \end{bmatrix} + \Psi \preceq 0,$$

where

$$\Theta \coloneqq (ET_{22}^{-1}T_{21}C_{cl})^\top Q^{-1} + Q^{-1}(ET_{22}^{-1}T_{21}C_{cl}),$$

$$\Psi \coloneqq \begin{bmatrix} C_{cl} & 0 & D \\ 0 & I & 0 \end{bmatrix}^\top \begin{bmatrix} \Sigma & \Lambda \\ 0 & I \end{bmatrix}^\top \begin{bmatrix} N_1^{-1} & 0 \\ 0 & -N_2^{-1} \end{bmatrix} \begin{bmatrix} \Sigma & \Lambda \\ 0 & I \end{bmatrix} \begin{bmatrix} C_{cl} & 0 & D \\ 0 & I & 0 \end{bmatrix}.$$

To eliminate the cross-coupling terms and simplify the structure, define:

$$T_{post} \coloneqq \begin{bmatrix} I & 0 & 0 \\ T_{21}C_{cl} & T_{22} & T_{21}D \\ 0 & 0 & I \end{bmatrix}, \quad T_{pre} \coloneqq T_{post}^\top.$$

Post- and pre-multiplying by the above by T_{pre} and T_{post}, respectively, yields:

$$\begin{bmatrix} A_{cl}^\top Q^{-1} + Q^{-1}A_{cl} - Q^{-1}\dot{Q}Q^{-1} + \alpha Q^{-1} & \star & \star \\ E^\top Q^{-1} & 0 & \star \\ F^\top Q^{-1} & 0 & -\lambda_w I \end{bmatrix} + \begin{bmatrix} C_{cl} & 0 & D \\ 0 & I & 0 \end{bmatrix}^\top T^\top \begin{bmatrix} N_1^{-1} & 0 \\ 0 & -N_2^{-1} \end{bmatrix} T \begin{bmatrix} C_{cl} & 0 & D \\ 0 & I & 0 \end{bmatrix} \preceq 0.$$

This matrix inequality implies that for all $\eta \in \mathbb{R}^{n_x}$, $\delta p \in \mathbb{R}^{n_p}$, and $w \in \mathbb{R}^{n_w}$, we have

$$\dot{V}(t) + \alpha V(t) + \begin{bmatrix} \delta q \\ \delta p \end{bmatrix}^\top T^\top \begin{bmatrix} N_1^{-1} & 0 \\ 0 & -N_2^{-1} \end{bmatrix} T \begin{bmatrix} \delta q \\ \delta p \end{bmatrix} - \lambda_w(t) w(t)^\top w(t) \leq 0.$$

It follows from the definition of the δQC (3.15) that the middle term is nonpositive. Thus, the dissipation inequality (3.24)

$$\dot{V}(t) + \alpha V(t) - \lambda_w(t) w(t)^\top w(t) \leq 0,$$

holds almost all $t \in [t_0, t_f]$. By Lemma 3.10, this implies the invariance of the state funnel $\mathcal{E}_\eta(t)$, and consequently, the invariance of the full funnel $\mathcal{F}(t)$ follows from Lemma 3.11. $\square$

3.2.5 Funnel constraints

We develop time-varying LMI conditions implying the constraint satisfaction as defined in Definition (3.2), requiring that all states and control inputs within the funnel respect the state and input constraints.

Let the feasible sets be defined as:

$$\mathcal{X} = \bigcap_{i=1}^{m_x} \mathcal{X}^i, \quad \mathcal{U} = \bigcap_{j=1}^{m_u} \mathcal{U}^j \tag{3.27a}$$

$$\mathcal{X}^i = \{x \mid h_i(x) \leq 0\}, \quad i = \mathbb{Z}_{[1,m_x]}, \tag{3.27b}$$

$$\mathcal{U}^j = \{u \mid g_j(u) \leq 0\}, \quad j = \mathbb{Z}_{[1,m_u]}, \tag{3.27c}$$

where each $h_i : \mathbb{R}^{n_x} \to \mathbb{R}$ and $g_j : \mathbb{R}^{n_u} \to \mathbb{R}$ are assumed to be continuously differentiable. Our goal is to ensure that the state and input funnels remain inside the feasible sets:

$$(\bar{x}(t) \oplus \mathcal{E}_\eta(t)) \subseteq \mathcal{X}, \tag{3.28a}$$

$$(\bar{u}(t) \oplus \mathcal{E}_\xi(t)) \subseteq \mathcal{U}, \quad \forall t \in [t_0, t_f], \tag{3.28b}$$

which can be compactly written as $\mathcal{F}(t) \subseteq \mathcal{X} \times \mathcal{U}$. However, when it comes to general nonlinear functions h_i and g_j, it is not tractable to reformulate the conditions (3.28) into equivalent LMIs. We therefore linearize h_i and g_j around the nominal trajectory $(\bar{x}, \bar{u})$, to obtain the following time-varying convex polytope approximations:

$$\mathcal{P}_x = \bigcap_{i=1}^{m_x} \mathcal{P}_x^i, \quad \mathcal{P}_u = \bigcap_{j=1}^{m_u} \mathcal{P}_u^i \tag{3.29a}$$

$$\mathcal{P}_x^i(t) = \{x \mid (a_i^h(t))^\top x \leq b_i^h(t)\}, \quad i = \mathbb{Z}_{[1,m_x]}, \tag{3.29b}$$

$$\mathcal{P}_u^u(t) = \{u \mid (a_j^g(t))^\top u \leq b_j^g(t)\}, \quad j = \mathbb{Z}_{[1,m_u]}, \tag{3.29c}$$

where

$$a_i^h(t) = \left.\frac{\partial h_i}{\partial x}\right|_{x=\bar{x}(t)}, \quad b_i^h(t) = a_i^h(t)^\top \bar{x}(t) - h_i(\bar{x}),$$
$$a_j^g(t) = \left.\frac{\partial g_j}{\partial u}\right|_{u=\bar{u}(t)}, \quad b_j^g(t) = a_i^g(t)^\top \bar{u}(t) - g_j(\bar{u}).$$

We now provide LMI conditions that ensures the funnel remains within the linearized state and input constraint sets (3.29).

Lemma 3.13. *Suppose $\bar{x}(t) \in \mathcal{P}_x(t)$ and $\bar{u}(t) \in \mathcal{P}_u(t)$ for all $t \in [t_0, t_f]$. Then, the inclusion $\mathcal{F}(t) \subseteq \mathcal{P}_x(t) \times \mathcal{P}_u(t)$ is equivalent to satisfying the following LMIs for all $t \in [t_0, t_f]$, $i = \mathbb{Z}_{[1,m_x]}$, and $j = \mathbb{Z}_{[1,m_u]}$:*

$$0 \preceq \begin{bmatrix} (b_i^h(t) - a_i^h(t)^\top \bar{x}(t))^2 & c_Q a_i^h(t)^\top Q(t) \\ c_Q Q(t) a_i^h(t) & c_Q Q(t) \end{bmatrix}, \tag{3.30a}$$
$$0 \preceq \begin{bmatrix} (b_j^g(t) - a_j^g(t)^\top \bar{u}(t))^2 & c_Q a_i^g(t)^\top Y(t) \\ c_Q Y(t)^\top a_i^g(t) & c_Q Q(t) \end{bmatrix}. \tag{3.30b}$$

Proof. We derive the LMI (3.30b) for the input constraint; the state constraint follows analogously. The inclusion $\bar{u} \oplus \mathcal{E}_\xi \subset \mathcal{P}_u$ means that for all $\xi \in \mathcal{E}_\epsilon$, we must have $(a_j^g)^\top(\bar{u}+\xi) \leq b_j^g$. This is equivalent to

$$\max_{\xi \in \mathcal{E}_\xi} (a_j^g)^\top (\bar{u} + \xi) \leq b_j^g.$$

Using the definition of the input funnel $\mathcal{E}_\xi$ in (3.7), we can write

$$\max_{\|y\|_2 \leq 1} (a_j^g)^\top (c_Q K Q K^\top)^{\frac{1}{2}} y \leq b_j^g - (a_j^g)^\top \bar{u},$$

where the term related to $\bar{u}$ is moved to the right-hand side since $\bar{u}$ is fixed. The solution of the maximization problem is equal to the norm $\|(c_Q K Q K^\top)^{\frac{1}{2}} a_j^g\|_2$. Hence, we obtain:

$$\|(c_Q K Q K^\top)^{\frac{1}{2}} a_j^g\|_2 \leq b_j^g - (a_j^g)^\top \bar{u}.$$

Since the left-hand side is always nonnegative, squaring both sides preserves the inequality:

$$c_Q (a_j^g)^\top K Q K^\top a_j^g \leq (b_j^g - (a_j^g)^\top \bar{u})^2.$$

Applying Schur complement generates:

$$\begin{bmatrix} (b-(a_j^g)^\top \bar{u})^2 & \sqrt{c_Q}a^\top K \\ \sqrt{c_Q}K^\top a & Q^{-1} \end{bmatrix} \succeq 0$$

Multiplying both sides by $\mathrm{diag}(1, \sqrt{c_Q}Q)$ and using $Y = KQ$ completes the proof. □

The inclusion $\mathcal{P}_x^i(t) \subseteq \mathcal{X}^i$ (or $\mathcal{P}_u^j(t) \subset \mathcal{U}^j$) is not guaranteed for general nonlinear functions h_i (or g_j). However, this inclusion holds under certain structural assumptions. Specifically, when the nonlinear functions are concave, the linearized sets are inner assumptions.

Lemma 3.14. *Let $h_i : \mathbb{R}^{n_x} \to \mathbb{R}$ be a concave and twice differentiable function, and suppose $\bar{x}(t) \in \mathcal{X}^i$ where $\mathcal{X}^i$ is defined in* (3.27) *for some $i \in \mathbb{Z}_{[1,m_x]}$. Then, the polytope set $\mathcal{P}_x^i(t)$, defined in* (3.29)*, satisfies $\mathcal{P}_x^i(t) \subseteq \mathcal{X}^i$.*

Proof. Pick any $x \in \mathcal{P}_x^i$, and then we want to show that $x \in \mathcal{X}^i$, that is, $h_i(x) \leq 0$. By Taylor's remainder theorem applied at $\bar{x}$, we have:

$$h_i(x) = h_i(\bar{x}) + (a_i^h)^\top (x-\bar{x}) + \frac{1}{2}(x-\bar{x})^\top \nabla^2 h(\tilde{x})(x-\bar{x}),$$

where $\tilde{x}$ lies on the line segment between $\bar{x}$ and x. Since h_i is concave and twice differentiable, we have $\nabla^2 h_i(\tilde{x}) \preceq 0$, and thus the quadratic term is nonpositive. Also, it follows from $x \in \mathcal{P}_x^i$ that

$$(a_i^h)^\top x \leq b_i^h = (a_i^h)^\top \bar{x}(t) - h_i(\bar{x}),$$

which implies $h_i(\bar{x}) + a_i^h(x-\bar{x}) \leq 0$. Combining the two, we get $h_i(x) \leq 0$, i.e., $x \in \mathcal{X}^i$. □

Remark 3.15. *If h_i is convex instead of concave, the linearized set $\mathcal{P}_x^i$ forms an outer approximation of $\mathcal{X}^i$. In this case, instead of using $\mathcal{P}_x^i$ derived in* (3.29)*, techniques from polyhedral inner approximation, such as those discussed in [45], can be used to construct a conservative polytope such that $\mathcal{P} \subset \mathcal{X}^i$. Then, one can enforce the constraint using LMIs similar to those in* (3.30)*.*

3.2.6 Funnel cost functions

A typical objective in funnel design is to maximize the size of the funnel entry to enlarge the set of initial states from which the system can be safely controlled. The proposed method does not restrict the choice of the cost function, as long as it remains convex with respect to the decision variables. In this subsection, we outline several convex cost functions that can be used to promote larger funnel entries while maintaining computational tractability.

We consider cost functions $J : \mathbb{S}^{n_x}_{++} \to \mathbb{R}$ that depend on the ellipsoidal shape of the funnel entry $Q(t_0)$. To reflect directional preference or scale differences in the state space, we define the weighted matrix $Q_{w0} \coloneqq W_0 Q(t_0) W_0$, where $W_0 \in \mathbb{S}^{n_x}_{++}$ is a diagonal weighting matrix. The examples of convex cost functions include:

$$J(Q(t_0)) \in \{-\mathrm{trace}(Q_{w0}), -\log\det(Q_{w0}), \mathrm{trace}(Q_{w0}^{-1})\}. \tag{3.31}$$

The first cost function promotes larger funnel entries by maximizing the sum of lengths of weighted principal axes. The second one is proportional to maximizing the volume of the weighted funnel entry [20]. The third one is to minimize the sum of the eigenvalues of Q_{w0}^{-1}, which encourages increasing the eigenvalues of Q_{w0}, thereby enlarging the funnel entry.

3.2.7 Funnel synthesis problem

We are now ready to formulate the funnel synthesis problem as a continuous-time optimization problem.

Problem 3.16. *Continuous-time funnel synthesis.*

$$\min_{Q(\cdot),Y(\cdot),N_1(\cdot),N_2(\cdot),\lambda_d(\cdot)} \quad J(Q(t_0)) \tag{3.32a}$$

$$s.t.\ Q(t) \succ 0, \tag{3.32b}$$

$$(3.25), \tag{3.32c}$$

$$(3.30) \tag{3.32d}$$

$$(N_1(t), N_2(t)) \in \mathcal{N}(t), \tag{3.32e}$$

$$Q(t_f) \preceq Q_f \tag{3.32f}$$

$$\forall t \in [t_0, t_f].$$

The constraint (3.32b) enforces positive definiteness of the matrix $Q(t)$. The dissipation inequality implied by the DLMI (3.32c) imposes the funnel invariance condition. The pointwise LMIs (3.32d) ensure the funnel feasibility. The inclusion (3.32e) enforces valid multiplier matrices for the nonlinearity. Finally, the terminal condition (3.32f) with $Q_f \in \mathbb{S}^{n_x}_{++}$ limits the size of the funnel at the final time.

Remark 3.17. *The resulting optimization problem is convex in continuous time, except for the decay rate α. Since α is a scalar, it can be efficiently selected via line search. In the special case where the disturbance w is absent, setting $\alpha = 0$ is sufficient for the funnel's invariance and leads to a fully convex formulation.*

In the rest of this subsection, we specify how the inclusion (3.32e) can be converted into pointwise (i.e., at each time t) linear constraints or LMIs. Consider the sector bounded nonlinearity illustrated in (3.18). Then, the inclusion $(N_1, N_2) \in \mathcal{N}$ is equivalent to the following linear equality constraints:

$$N_1(t) = \lambda_{sec}(t)S, \quad N_2(t) = \lambda_{sec}(t)S, \tag{3.33}$$

for any positive scalar-valued function $\lambda_{sec} : [t_0, t_f] \to \mathbb{R}_+$. In this case, the function $\lambda_{sec}(t)$ can be treated as an our decision variable with the constraint on $\lambda_{sec}(t) \geq 0$, while preserving the convexity of our funnel synthesis problem (3.32).

For the Lipschitz nonlinearity (3.19) and its associated multiplier matrix (3.21), the inclusion (3.32e) is equivalent to the following linear equality constraints:

$$N_1(t) = \text{blkdiag}(\lambda^\gamma_1(t) I_{n_{q_{[1]}}}, \ldots, \lambda^\gamma_{n_c}(t) I_{n_{q_{[n_c]}}}), \tag{3.34a}$$

$$N_2(t) = \text{diag}(\lambda^\gamma_1(t)\gamma^2_1(t), \ldots, \lambda^\gamma_{n_c}(t)\gamma^2_{n_c}(t)), \tag{3.34b}$$

where $\lambda^\gamma_i : [t_0, t_f] \to \mathbb{R}_+$ are positive-valued functions that can be optimized as decision variables with the constraint $\lambda^\gamma_i(t) \geq 0$.

On the other hand, for the L-smooth nonlinearity (3.20) and its multiplier matrix (3.22), the inclusion (3.32e) can be converted into the following two linear equality and one LMI

constraint:

$$N_1(t) = \text{blkdiag}(\lambda_1^\beta I_{n_{q_{[1]}}}, \ldots, \lambda_{n_c}^\beta I_{n_{q_{[n_c]}}}), \tag{3.35a}$$

$$N_2(t) = \text{diag}(\lambda_1^\beta s_1(t)\gamma_1^2(t), \ldots, \lambda_{n_c}^\beta s_{n_c}(t)\gamma_{n_c}^2(t)), \tag{3.35b}$$

$$\begin{bmatrix} s_i(t)I_{n_x} & * \\ (C_iQ(t) + D_iY(t))^\top & Q(t) \end{bmatrix} \succeq 0, \quad \forall i \in \mathbb{Z}_{[1,n_c]}. \tag{3.35c}$$

where $\lambda_i^\beta \in \mathbb{R}_+$ are positive constants and $s_i : [t_0, t_f] \to \mathbb{R}_+$ are scalar-valued functions that can be jointly optimized as decision variables.

3.2.8 Recursive feasibility for nominal MPC

This subsection demonstrates that the funnel computed by solving (3.32) can be used to guarantee the recursive feasibility of a nominal MPC in which the bounded disturbance in (3.1) is not neglected. Consider the following finite-horizon optimal control problem (FHC):

$$\min_{u_o(\cdot)} \int_{t_s^i}^{t_s^f} J_{\text{MPC}}(x(\tau), u_o(\tau))\mathrm{d}\tau \tag{3.36a}$$

$$\text{s.t. } \dot{x} = f(x, u_o, 0), \tag{3.36b}$$

$$x(t) \in \mathcal{X},\ u_o(t) \in \mathcal{U}, \tag{3.36c}$$

$$x(t_s^i) = x_s, \tag{3.36d}$$

$$x(t_s^f) \in \bar{x}(t_s^f) \oplus \mathcal{E}_\eta(t_s^f), \tag{3.36e}$$

where $x_s \in \mathbb{R}^{n_x}$ is the given initial condition and $J_{\text{MPC}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is the stage cost. The problem differs from a standard MPC formulation only in the terminal constraint, which enforces that the terminal state lies inside the funnel at $t = t_s^f$, thereby enabling recursive feasibility.

Lemma 3.18. *Suppose the FHC is feasible at t_0 with horizon length T_0^{MPC} such that $t_0 + T_0^{MPC} \leq t_f$. Given the funnel $\mathcal{F}(t)$ computed by* (3.32) *and the terminal constraint* (3.36e)*, the FHC remains feasible at t_j for step lengths $\delta_j = t_{j+1} - t_j$ satisfying $0 \leq \delta_j \leq T_j$ and $T_{j-1}^{MPC} - \delta_{j-1} \leq T_j^{MPC}$ for all $t_j + T_j^{MPC} \leq t_f$.*

Proof. We proceed by induction. The base case where $t = t_0$ holds because of assumption. For the induction step, assume that the FHC is feasible at $t = t_j$, and let $u_j(t)$ be its optimal solution over $[t_j, t_j + T_j^{\mathrm{MPC}}]$. The terminal condition (3.36e) implies $x(t_j^f) \in \bar{x}(t_j^f) \oplus \mathcal{E}_\eta(t_j^f)$ with $t_j^f = t_j + T_j^{\mathrm{MPC}}$. For the FHC problem at $t = t_{j+1}$, define the candidate control input $u_{j+1}(t)$ constructed by

$$u_{j+1}(t) = \begin{cases} u_j(t) & t \in [t_{j+1}, t_j + T_j^{\mathrm{MPC}}] \\ \bar{u}(t) + K(t)(x(t) - \bar{x}(t)) & t \in [t_j + T_j^{MPC}, t_{j+1} + T_{j+1}^{\mathrm{MPC}}] \end{cases},$$

where $K(t)$ is the feedback gain obtained from the funnel synthesis in (3.32). The first segment is feasible because it is part of the previous optimal solution. At $t = t_j + T_j^{\mathrm{MPC}}$, the state lies inside of the funnel, and by the funnel construction, the feedback law in the second segment keeps the state and input feasible. Thus, $u_{j+1}(t)$ is a feasible solution at t_{j+1}, completing induction. □

Remark 3.19. *If the terminal time $t_j + T_j^{MPC}$ would exceed t_f, we simply replace it by t_f. Because the funnel invariance holds up to t_f, this modification ensures that the state remains inside the admissible set until the end of the horizon, thereby preserving recursive feasibility.*

The constructed MPC algorithm can be summarized as follows:

1. Initialization: Set $j = 0$.

2. Iteration: At each computation time t_j, for $j \in \mathbb{Z}_{[0,\infty)}$ and $t_j \leq t_f$ perform:

   (a) Measure the state $x(t_j)$ of the nominal system and set $x_s = x(t_j)$.

   (b) If $t_j + T_j^{\mathrm{MPC}}$, solve the FHC problem with $t_s^j = t_j$ and $t_s^f = \min(t_f, t_j + T_j^{\mathrm{MPC}})$ to obtain $u(t)$ over $t \in [t_j, t_s^f]$.

   (c) Apply $u(t)$ to the nominal system over $[t_j, \min(t_{j+1}, t_f)]$.

   (d) If $t = t_f$, terminate the iteration.

3.2.9 Extension to Uncertain Nonlinear Systems

This subsection describes how to derive the same form of the incremental system (3.12) from uncertain nonlinear systems. The derivation extends the approach for nonlinear systems (3.1) under the assumption that the introduced uncertain term satisfies δQC (3.15). We consider the following uncertain nonlinear systems:

$$\begin{aligned}\dot{x}(t) &= f(t, x(t), u(t), w(t)) + E_p\psi(t), \quad t \in [t_0, t_f],\\ q_\psi(t) &= C_\psi x(t) + D_\psi u(t) + G_\psi w(t),\\ \psi(t) &= \Delta_\psi(t, q_\psi(t)).\end{aligned}$$

Here, the pair (q_ψ, ψ) represents the uncertainty where $\Delta_\psi : [t_0, t_f] \times \mathbb{R}^{n_{q_\psi}} \to \mathbb{R}^{n_\psi}$ is an unknown function, and $q_\psi(t) \in \mathbb{R}^{n_{q_\psi}}$ is its argument. The matrices C_ψ, D_ψ, D_ψ, and E_ψ are selector matrices with entries of 0s and 1s, chosen to organize q_ψ and ψ. This uncertain function Δ_ψ is assumed to satisfy δQC (3.15) with the pair (q_ψ, ψ).

Assumption 3.20. *It is assumed that the uncertain nonlinear system* (3.37) *is well-posed, that is, for any* $x(t_0) \in \mathbb{R}^{n_x}$, $u(\cdot) \in \mathcal{L}_2^{n_u}[t_0, t_f]$, $w(\cdot) \in \mathcal{L}_2^{n_w}[t_0, t_f]$, *there exist unique solutions* $x(\cdot) \in \mathcal{L}_2^{n_x}[t_0, t_f]$ *and* $\psi(\cdot) \in \mathcal{L}_2^{n_\psi}[t_0, t_f]$.

Next, we describe the assumption on the nominal trajectory $(\bar{x}, \bar{u})$ of the uncertain system (3.37). For the nominal (uncertain-free) nonlinear systems (3.1), it is assumed that the nominal trajectory is dynamically feasible (3.4a). However, this assumption is not appropriate for the uncertain systems, since the uncertainty is unknown and cannot be specified during trajectory generation.

Instead, we assume that the nominal trajectory is dynamically feasible with the following dynamics:

$$\dot{\bar{x}}(t) = f(t, \bar{x}(t), \bar{u}(t), 0) + E_\psi\hat{\psi}(t), \tag{3.38a}$$

$$\bar{q}_\psi(t) = C_\psi\bar{x}(t) + D_\psi\bar{u}(t), \tag{3.38b}$$

$$\hat{\psi}(t) = \hat{\Delta}_\psi(t, \bar{q}_\psi(t)), \tag{3.38c}$$

where $\hat{\Delta}_\psi : [t_0, t_f] \times \mathbb{R}^{n_{q_\psi}} \to \mathbb{R}^{n_\psi}$ is a known function serving as an approximation of the uncertain function Δ_ψ in (3.37).

The incremental dynamical system for the uncertain nonlinear system is given by

$$\dot{\eta}(t) = A_o(t)\eta(t) + B_o(t)\xi(t) + F_o(t)w(t) + E\delta\phi(t) + E_\psi(\psi(t) - \hat{\psi}(t)).$$

This can be equivalently rewritten as

$$\dot{\eta}(t) = A_o(t)\eta(t) + B_o(t)\xi(t) + F_o(t)w(t) + E\delta\phi(t) + E_\psi(\psi(t) - \bar{\psi}(t)) + E_\psi(\bar{\psi}(t) - \hat{\psi}(t)),$$

where $\bar{\psi}(t) = \Delta_\psi(t, \bar{q}_\psi)$. The term $\bar{\psi} - \hat{\psi}$ represents the dynamic infeasibility of the nominal trajectory. We assume that this error is bounded by

$$\|\bar{\psi}(t) - \hat{\psi}(t)\|_2 \leq e_\psi,$$

for some $e_\psi \in \mathbb{R}_+$ and all $t \in [t_0, t_f]$. This can be viewed as a bounded disturbance for the incremental system. Meanwhile, the incremental uncertain term $\psi - \bar{\psi}$ represents the state- and input-dependent uncertainty, which can be modeled by δQC (3.15). The same machinery for nonlinear systems, including incremental analysis and funnel computation, can now be applied to uncertain nonlinear systems.

Chapter 4

COMPUTATION OF FUNNEL SYNTHESIS VIA NUMERICAL OPTIMAL CONTROL

Continued from Chapter 3, this chapter focuses on solving the continuous-time funnel synthesis problem (3.32) using an optimal control approach. The notion of funnel dynamics, represented by a differential linear matrix equality (DLME), is introduced to reformulate the problem as an optimal control problem. Using numerical optimal control techniques, the continuous-time problem is approximated by a discrete-time optimal control problem. To achieve continuous-time constraint satisfaction (CTCS), two methods are presented: i) introducing intermediate constraint-checking points, and ii) applying successive convexification (SCvx) with subgradients. The method is demonstrated on two numerical examples: a unicycle and a 6-degree-of-freedom quadrotor performing obstacle avoidance.

Chapter-specific notation. This chapter uses the same notation as that of Chapter 3.

4.1 Solution methods

What distinguishes the funnel synthesis problem 3.32 from conventional LMI-based robust controller synthesis [4] is that it is formulated as a continuous-time optimization problem, thereby involving DLMI and pointwise LMIs rather than time-invariant LMIs. As a result, it cannot be directly solved using off-the-shelf SDP solvers. To enable tractable computation, a dedicated solution method is required to reformulate the problem into a finite dimensional SDP, and this reformulation must approximate the original problem as closely as possible, ideally without introducing conservatism.

Our solution methods are based on numerical optimal control techniques. To reformulate the funnel synthesis problem (3.32) as an equivalent optimal control problem, we first introduce the notion of funnel dynamics that plays a role analogous to the system dynamics in a standard optimal control problem.

4.1.1 Funnel Dynamics

To derive funnel dynamics, we rewrite the DLMI condition (3.26) as an equivalent differential matrix equation (DME) that serves as the funnel dynamics along with a pointwise LMI constraint. We consider two types of funnel dynamics: Lyapunov-type and Direct-type, allowing us the flexibility to choose between them.

The DLMI condition in (3.26) can be equivalently written as

$$\dot{Q}(t) = H_{11}(t) + H_{11}(t)^\top + Z_1(t), \tag{4.1a}$$

$$\begin{bmatrix} -Z_1(t) & \star \\ H_{12}(t) & H_{22}(t) \end{bmatrix} \preceq 0, \tag{4.1b}$$

where a slack symmetric matrix-valued function $Z_1 : [t_0, t_f] \to \mathbb{S}^{n_x}$ is introduced for this equivalent conversion. We refer to (4.1a) as the Lyapunov-type funnel dynamics as it structurally resembles Lyapunov differential equations [15].

Alternatively, the DLMI condition (3.26) can be rewritten as

$$\dot{Q}(t) = Z_2(t), \tag{4.2a}$$

$$\begin{bmatrix} H_{11}(t) + H_{11}(t)^\top - Z_2(t) & \star \\ H_{12}(t) & H_{22}(t) \end{bmatrix} \preceq 0, \tag{4.2b}$$

where $Z_2 : [t_0, t_f] \to \mathbb{S}^{n_x}$ is a symmetric slack matrix-valued function. We refer to this formulation (4.2) as the Direct-type funnel dynamics as the slack variable Z_2 directly defines the evolution of Q.

Remark 4.1. *It is important to note that both formulations are mathematically equivalent. However, once control parameterization is applied after discretization, the resulting solution space differs depending on the chosen funnel dynamics. For instance, piecewise constant parameterization of $Z_2(t)$ yields a piecewise linear $Q(t)$, whereas the same applied to $Z_1(t)$ does not, as $Q(t)$ is implicitly defined as a solution of differential equation* (4.1a).

Both formulations, (4.1a) and (4.2a), define LTV systems where Q serves as the state and all other decision variables , including Y and Z (which corresponds to Z_1 in (4.1) or

Z_2 in (4.2)), act as control inputs. Consequently, the block-form LMIs (4.1b) and (4.2b) naturally represent state-input constraints.

To simplify subsequent analysis, we introduce vectorized variables as follows:

$$q_v \coloneqq \mathrm{vec}(Q), \quad y_v = \mathrm{vec}(Y), \quad z_v = \mathrm{vec}(Z).$$

Using these variables, both forms of the funnel dynamics, (4.1a) and (4.2a), can be expressed in the vectorized form:

$$\dot{q}_v(t) = \underbrace{A_q(t)q_v(t) + B_{qy}(t)y_v(t) + B_{qz}(t)z_v(t)}_{\coloneqq F(t,q_v,y_v,z_v)}, \tag{4.3}$$

where $A_q(t) \in \mathbb{R}^{n_q \times n_q}$, $B_{qy}(t) \in \mathbb{R}^{n_q \times n_y}$, and $B_{qz}(t) \in \mathbb{R}^{n_q \times n_q}$ are time-varying matrices that can be constructed using Kronecker products [21, 59] with dimensions $n_q = n_x^2$, $n_y = n_x n_u$, and $n_z = n_q$. We also define the vectorized form of the multiplier matrices N_1 and N_2 as:

$$m_{1,v} \coloneqq \mathrm{vec}(N_1), \quad m_{2,v} = \mathrm{vec}(N_2).$$

4.1.2 Discretization

We consider a uniform time grid defined as

$$t_k = t_0 + \frac{k}{N}(t_f - t_0), \quad k \in \mathbb{Z}_{[0,N]}, \tag{4.4a}$$

$$\triangle_k = \triangle(t_k), \tag{4.4b}$$

where $N \in \mathbb{Z}_{++}$. The symbol $\triangle$ serves as a placeholder for any time-varying variable, and we denote its value at $t = t_k$ by $\triangle_k \coloneqq \triangle(t_k)$. Each $\triangle_k$ is referred to as a node point. With the uniform time grid (4.4), we use the first-order hold (FOH) interpolation for the control inputs in the funnel dynamics, defined by

$$\square(t) = \lambda_k^m(t)\square_k^m + \lambda_k^p(t)\square_{k+1}^p, \quad \forall t \in [t_k, t_{k+1}), \tag{4.5a}$$

$$\lambda_k^m(t) = \frac{t_{k+1} - t}{t_{k+1} - t_k}, \quad \lambda_k^p(t) = \frac{t - t_k}{t_{k+1} - t_k}, \tag{4.5b}$$

for all $k \in \mathbb{Z}_{[0,N)}$. The symbol $\square$ is a placeholder for any funnel input variables (e.g. y_v, z_v, m_v) for the funnel dynamics. To employ the zeroth-order hold (ZOH), it suffices to set

$\square_k^m = \square_{k+1}^p$ for all $k \in \mathbb{Z}_{[0,N-1]}$. Also, continuity at each node point $t = t_k$ can be enforced by $\square_k^p = \square_k^m$ for all $k \in \mathbb{Z}_{[1,N-1]}$.

We adopt a multiple-shooting scheme [16] to enforce the funnel dynamics. Over each subinterval $[t_k, t_{k+1})$, the funnel state evolves according to

$$q_v(t) = q_v(t_k) + \int_{t_k}^{t} F(\tau, q_v(\tau), y_v(\tau), z_v(\tau))\mathrm{d}\tau,$$

where F is defined in (4.3). Enforcing continuity across subintervals yields the funnel dynamics constraint:

$$q_{v,k+1} = q_v(t_k) + \int_{t_k}^{t_{k+1}} F(\tau, q_v(\tau), y_v(\tau), z_v(\tau))\mathrm{d}\tau. \tag{4.6}$$

Since F is linear in its arguments except t, and FOH interpolation is used, the right-hand side of (4.6) is affine in the decision variables $q_{v,k}, y_{v,k}^m, y_{v,k+1}^p, z_{v,k}^m$, and $z_{v,k+1}^p$. Thus, we can exactly rewrite the constraint (4.6) as

$$q_{v,k+1} = A_k^q q_{v,k} + B_{yk}^m y_{v,k}^m + B_{yk}^p y_{v,k+1}^p + B_{zk}^m z_{v,k}^m + B_{zk}^p z_{v,k+1}^p, \tag{4.7}$$

where A_k^q,$B_{yk}^{m/p}$, and $B_{zk}^{m/p}$ are appropriately sized matrices that can be obtained via variational discretization [72].

4.1.3 Positive Definiteness of Q

To ensure the validity of the Lyapunov function (3.5) and the associated funnel (3.6), it is essential to maintain $Q(t) \succ 0$ for all $t \in [t_0, t_f]$. Since $q_{v,k}$ is our decision variable, we can enforce $Q(t) \succ 0$ for each $t = t_k$ for all $k \in \mathbb{Z}_{[0,N]}$. However, the positive definiteness of $Q(t)$ within the subinterval (t_k, t_{k+1}) depends on the funnel dynamics and the corresponding control inputs, which may lead to inter-node constraint violations. Here, we present cases where the funnel state Q preserves its positive definiteness for all t by only imposting $Q(t) \succ 0$ at each node point.

Lemma 4.2. *Consider the Direct-type funnel dynamics* (4.2a)*, and let the variable* $Z(t)$ *(or its vectorized form* $z_v(t)$*) be parameterized by the FOH scheme* (4.5) *such that*

$$Z_k^m \succeq Z_{k+1}^p \tag{4.8}$$

for all $k \in \mathbb{Z}_{[0,N-1]}$. *Then, imposing* $Q(t) \succ 0$ *for each* $t = t_k$ *for all* $k \in \mathbb{Z}_{[0,N]}$ *ensures the solution of* (4.2a) *preserves the positive definiteness, that is,* $Q(t) \succ 0$ *for all* $t \in [t_0, t_f]$.

Proof. Let the step size be $d_k = t_{k+1} - t_k > 0$ and introduce the local time $\theta_t = t - t_k$. With this variable, the FOH expression becomes

$$Z(\theta_t) = Z_k^m + Z_k^r \theta_t,$$

where

$$Z_k^r = \frac{Z_{k+1}^p - Z_k^m}{dt_k}.$$

Note that by assumption $Z_k^m \succeq Z_{k+1}^p$, so $Z_k^r \preceq 0$. Integrating the Direct-funnel dynamics $\dot{Q}(t) = Z(t)$ from t_k to t gives

$$Q(t_k + \theta_t) = Q(t_k) + \theta_t Z_k^m + \frac{1}{2}\theta_t^2 Z_k^r. \tag{4.9}$$

Setting $t = t_{k+1}$ (so, $\theta_t = d_k$) reproduces the node value

$$Q(t_{k+1}) = Q(t_k) + d_k Z_k^m + \frac{1}{2} d_k^2 Z_k^2 \succ 0. \tag{4.10}$$

Define a convex combination $\tilde{Q}(t)$ of $Q(t_k)$ and $Q(t_{k+1})$ for $t \in [t_k, t_{k+1}]$:

$$\tilde{Q}(t) \coloneqq (1 - \frac{\theta_t}{d_k})Q(t_k) + \frac{\theta_t}{d_k} Q(t_{k+1}). \tag{4.11}$$

Since $\tilde{Q}(t)$ is the convex combination of two PD matrices $Q(t_k)$ and $Q(t_{k+1})$, $\tilde{Q}(t) \succ 0$ for all $t \in [t_k, t_{k+1}]$. Substituting (4.10) into $\tilde{Q}$ generates

$$\tilde{Q}(t) = Q(t_k) + \theta_t Z_k^m + \frac{1}{2}\theta_t d_k Z_k^r.$$

Comparing this with (4.9) and using $0 \leq \theta_t \leq d_k$ together with $Z_k^2 \preceq 0$ gives $\tilde{Q}(t) \preceq Q(t)$. Hence, $Q(t) \succ 0$ for all $t \in [t_k, t_{k+1}]$. We can apply this argument for all $k \in \mathbb{Z}_{[0,N-1]}$, so $Q(t) \succ 0$ for all $t \in [t_0, t_f]$. □

Remark 4.3. *It is known that the solution of a differential Riccati equation preserves the positive definiteness of* Q *[35]. However, the Lyapunov-type funnel dynamics* (4.1a) *are not in exact Riccati form, and therefore do not guarantee* $Q(t) \succ 0$ *over the entire interval. One*

possible way to leverage the positivity-preserving property of Riccati-type dynamics is to treat the feedback gain K as a decision variable instead of Y. This, however, introduces bilinear terms between K and Q, leading to a nonconvex formulation. To maintain convexity, we retain the use of Lyapunov-type dynamics with Y as the decision variable, and instead address potential inter-sample constraint violations directly in the next section.

4.1.4 Nodal constraint satisfaction

In addition to the funnel dynamics, we must satisfy the invariance condition (4.1b) or (4.2b), along with the state and input constraints (3.30) for all $t \in [t_0, t_f]$. For notational simplicity, we collectively express these conditions in the unified form:

$$L_l(t, q_v(t), \zeta_v(t)) \preceq 0, \quad i \in \mathbb{Z}_{[1,m_L]}, \tag{4.12}$$

where $\zeta_v := \{y_v, z_v, n_v\}$ collects decision variables other than q, and the indices l are partitioned as follows:

- $l \in \mathbb{Z}_{[1,m_{inv}]}$: invariance condition, (4.1b)or (4.2b) where $m_{inv} = 1$,
- $l \in \mathbb{Z}_{[m_{inv}+1,m_l]}$: state constraints (3.30a) and input constraints (3.30b) where $m_l = m_{inv} + m_x + m_u$.

Enforcing (4.12) continuously over the entire horizon $[t_0, t_f]$ is nontrivial, as decision variables are only defined at discrete node points t_k. A practical approximation is to enforce constraints only at the nodes. Specifically, the nodal constraint satisfaction requires:

$$L_l(t_k, q_{v,k}, \zeta^m_{v,k}) \preceq 0, \tag{4.13a}$$

$$L_l(t_{k+1}, q_{v,k+1}, \zeta^p_{v,k+1}) \preceq 0, \tag{4.13b}$$

$$\forall k \in \mathbb{Z}_{[0,N-1]}, \quad \forall l = \mathbb{Z}_{[1,m_l]}.$$

With the first-order interpolation (4.5), we allow discontinuities of $\zeta_v(t)$ at node points, so the constraints are enforced at both left and right evaluations of each node, given by (4.13a) and (4.13b), respectively.

4.1.5 Discrete funnel synthesis problem

We now derive a finite-dimensional optimization problem based on the discretization of the continuous-time funnel synthesis problem (3.32) over the uniform time grid (4.4). This discretized formulation is referred to as the discrete funnel synthesis problem.

Problem 4.4. *(Discrete funnel synthesis).*

$$\min_{q_{v,k},\zeta^m_{v,k},\zeta^p_{v,k}} J(Q_0) \tag{4.14a}$$

$$(4.7), \tag{4.14b}$$

$$(4.13), \tag{4.14c}$$

$$(\mathrm{mat}((m_{1,v})_k), \mathrm{mat}((m_{2,v})_k)) \in \mathcal{N}(t_k), \forall k \in \mathbb{Z}_{[0,N]} \tag{4.14d}$$

$$\mathrm{mat}(q_N) \preceq Q_f. \tag{4.14e}$$

The continuous-time set multiplier set inclusion (3.32e) is imposed at each node point t_k, leading to the pointwise inclusion condition in (4.14d). This nodewise enforcement is sufficient under the following conditions. For sector bounded nonlinearity (3.18), if the inclusion constraint (3.33) hold at each node, it holds for all $t \in [t_0, t_f]$ by convexity and FOH interpolation. For Lipschitz nonlinearity (3.21), if we define the conservative Lipschitz constant on each subinterval as

$$\gamma_k \coloneqq \max_{t\in[t_k,t_{k+1}]} \gamma(t),$$

then enforcing the corresponding multiplier condition (3.34) only at the node points ensures constraint satisfaction throughout the entire interval. For L-smooth nonlinearity (3.22), a similar bounding can be applied by defining conservative value

$$\beta_k \coloneqq \max_{t\in[t_k,t_{k+1}]} \beta(t).$$

However, unlike the sector-bounded and Lipschitz cases, the time-varying LMI (3.35c) associated with L-smooth nonlinearity must be enforced over the entire interval, not just at the nodes. In our formulation, we address this by including an additional LMI condition in L_l as described in (4.12), ensuring that the L-smooth constraints are enforced properly.

The discrete funnel synthesis problem (4.14) is a finite-dimensional SDP, which can be efficiently solved using standard solvers such as Mosek [7] or Clarabel [43].

4.2 Continuous-time constraint satisfaction

In this section, we present two techniques to enhance the continuous-time satisfaction of time-varying LMI constraints (4.12) beyond enforcement only at discrete node points.

4.2.1 Constraint reformulation

In numerical optimal control, continuous-time inequality constraints (e.g., $g(x(t)) \leq 0$) are often difficult to enforce directly at all times. One remedy is constraint reformulation [72, 40] , where violations over a time interval are quantified and either penalized or constrained via an integrated form.

We adopt a similar idea to our continuous-time LMI constraints. Specifically, noting that the matrix inequality $L_l(t, q_v(t), \zeta_v(t)) \preceq 0$ is equivalent to the scalar inequality $\lambda_{max}(L_l(t, q_v(t), \zeta_v(t)) \leq 0$, the satisfaction of (4.12) over the entire subinterval $t \in [t_k, t_{k+1}]$ is equivalent to the integral condition:

$$\int_{t_k}^{t_{k+1}} \max(0, \lambda_{max}(L_l(t, q_v(t), \zeta_v(t)))^{p_l} dt \leq 0, \tag{4.15}$$

where the exponent satisfies $p_l \geq 1$. This equivalence holds because $\lambda_{max}(\cdot)$ is continuous [53, Chapter 2, Section 5.1], and $q(\cdot)$, and $\zeta(\cdot)$ are continuous on each subinterval $[t_k, t_{k+1}]$; the detailed proof of this equivalence is given in [40, Lemma 2]. Since $q(t)$ and $\zeta(t)$ are linear functions of the decision variables $q_{v,k}$, $\zeta^m_{v,k}$, and $\zeta^p_{v,k+1}$ within the subinterval, we define the left-hand side of (4.15) as:

$$h^l_k(\chi_k) \leq 0, \quad k \in \mathbb{Z}_{[0,N-1]}, \tag{4.16}$$

where $\chi_k \coloneqq (q_{v,k}, \zeta^m_{v,k}, \zeta^p_{v,k+1})$ collects the relevant decision variables on $[t_k, t_{k+1}]$, and h^l_k is an integral functional.

Lemma 4.5. *The function $h^l_k(\cdot)$ is convex for each subinterval $[t_k, t_{k+1}]$.*

Proof. For each $t \in (t_k, t_{k+1})$, the matrix-valued function L_l is affine in χ_k. Since $\lambda_{max}(\cdot)$ is convex, and $\max(0, \cdot)$ is convex and nondecreasing, their composition $\max(0, \lambda_{max}(L_l(t, q(t), \zeta(t))$ is convex in χ_k [20, Chapter 3.2.4]. Finally, integration over $t \in [t_k, t_{k+1}]$ preserves the convexity, so $h^l_k(\chi_k)$ is convex. □

Remark 4.6. *Although $h_k^l(\cdot)$ is convex, it does not take the form of a standard conic constraint, such as second-order cone or a PSD cone. Hence, it cannot be directly handled by conventional conic optimization solvers.*

4.2.2 Intermediate constraint-checking points

One way to make h_k^l tractable is to approximate the integral by a finite sum using standard integration techniques, such as Simpson's method with the midpoint rule [23]. This yields the approximation:

$$\int_{t_k}^{t_{k+1}} \max(0, \lambda_{max}(L_l(t, q_v(t), \zeta_v(t)))^{p_l} dt \approx \sum_{s=0}^{N_s+1} c_s \max(0, \lambda_{max}(L_l(t_{k,s}, q_v(t_{k,s}), \zeta_v(t_{k,s})))^{p_l},$$

where N_s is the number of intermediate points, $c_s \in \mathbb{R}_{++}$ are integration weights, $t_{k,s}$ for $s \in \mathbb{Z}_{[1,N_s]}$ are the evaluation points with $t_{k,0} = t_k$ and $t_{k,N_s+1} = t_{k+1}$. Note that the following equivalence holds:

$$\sum_{s=0}^{N_s+1} c_s \max(0, \lambda_{max}(L_l(t_{k,s}, q_v(t_{k,s}), \zeta_v(t_{k,s})))^{p_l} \leq 0$$
$$\Leftrightarrow L_l(t_{k,s}, q_v(t_{k,s}), \zeta_v(t_{k,s})) \preceq 0, \quad \forall s \in \mathbb{Z}_{[0,N_s+1]}.$$

This equivalence holds because each c_s is nonnegative, and the integrand $\max(0, \lambda_{\max}(\cdot))^p$ is nonnegative as well. Therefore, the entire sum is nonpositive if and only if each term in the sum is zero, which implies that all matrix inequalities are individually satisfied.

As a result, approximating the integral by a finite sum leads to the introduction of intermediate constraint-checking points $t_{k,s} \in (t_k, t_{k+1})$, at which the LMI constraints (4.12) are enforced. Under the FOH interpolation (4.5) and the linear funnel dynamics (4.3), $q(t_{k,s})$ and $\zeta(t_{k,s})$ at any $t_{k,s} \in [t_k, t_{k+1}]$ become linear functions of the nodal variables $(q_k, \zeta_k, \zeta_{k+1})$. Therefore, the conditions

$$L_l(t_{k,s}, q(t_{k,s}), \zeta(t_{k,s})) \preceq 0, \tag{4.17}$$

remain LMIs in the decision variables and can be directly incorporated into the discrete problem formulation (4.14).

The intermediate points $t_{k,s}$ can be uniformly distributed over each subinterval using

$$t_{k,s} = t_k + \frac{s}{N_s + 1}(t_{k+1} - t_k), \quad s \in \mathbb{Z}_{[1,N_s]},$$

where $N_s \in \mathbb{R}_{++}$ is the number of intermediate points per subintervals.

Remark 4.7. *Adding the intermediate constraint-checking points* (4.17) *is more efficient compared to merely increasing the number k of node points. While both strategies improve constraint satisfaction over* $[t_0, t_f]$*, increasing node points leads to more decision variables and thus higher computational cost. In contrast, adding checking points only increases the number of LMI constraints without expanding the optimization variables, offering a more computationally efficient approach. Another benefit of this approach is that it preserves the number of control updates, thereby avoiding unnecessary increases in control frequency, which is often a desirable property in practical applications.*

4.2.3 Derivation of subgradient

Instead of approximating the integral, the second approach is to obtain the (sub) gradient of $h_k^l(\cdot)$ and apply the successive convexification method illustrated in [40]. In this and next subsections, we establish some related theoretical results.

We define the function g_k^l as

$$g_k^l(\chi_k) \coloneqq \int_{t_k}^{t_{k+1}} \tilde{g}_k^l(t, \chi_k)\mathrm{d}t, \tag{4.18}$$

where $\tilde{g}_k^l \in \partial_{\chi_k}\{\max(0, \lambda_{max}(L_l(t, q_v, \zeta_v)))^{p_l}\}$ denotes a subgradient of the integrand with respect to χ_k at time t.

Lemma 4.8. *If* $\tilde{g}_k^l(t, \chi_k)$ *is Lebesgue integrable over* $[t_k, t_{k+1}]$*, then* $g_k^l(\chi_k)$ *is a subgradient of* h_k^l *at* χ_k*.*

Proof. For notational brevity, we abuse the notation L_l as a matrix-valued function defined by:

$$L_l(t, \chi_k) \coloneqq L_l(t, q_v(t), \zeta_v(t)), \quad t \in [t_k, t_{k+1}].$$

Since q_v and ζ_v are affine in χ_k, L_l is also affine in χ_k for each fixed t. Let $\bar{\chi}_k$ be any point in the domain. For each fixed $t \in [t_k, t_{k+1}]$, by the definition of the subgradient, we have

$$\max(0, \lambda_{max}(L_l(t, \bar{\chi}_k)))^{p_l} \geq \max(0, \lambda_{max}(L_l(t, \chi_k)))^{p_l} + \tilde{g}_k^l(t, \chi_k)^\top (\bar{\chi}^k - \chi_k).$$

This inequality holds pointwise in t, and both sides are Lebesgue integrable since $\tilde{g}_k^l(t, \chi_k)$ is assumed to be Lebesgue integrable. We can therefore integrate both sides over the interval $[t_k, t_{k+1}]$, yielding:

$$\begin{aligned} h_k^l(\bar{\chi}_k) &\geq h_k^l(\chi_k) + \int_{t_k}^{t_{k+1}} \tilde{g}_k^l(t, \chi_k)^\top (\bar{\chi}^k - \chi_k) \mathrm{d}t, \\ &= h_k^l(\chi_k) + g_k^l(\chi_k)^\top (\bar{\chi}^k - \chi_k). \end{aligned}$$

Therefore, $g_k^l(\chi_k)$ is a subgradient of h_k^l at χ_k. □

About the integrability of $\tilde{g}_k^l$, we can establish the following result.

Lemma 4.9. *For a subinterval $[t_k, t_{k+1}]$ and a power $p_l \geq 1$, there exists a Lebesgue measurable selection $\tilde{g}_k^l(t, \chi_k) \in \partial_{\chi_k}\{\max(0, \lambda_{max}(L_l(t, \chi_k)))^{p_l}\}$ for almost every $t \in [t_k, t_{k+1}]$ and this selection is bounded on the subinterval. Hence, $\tilde{g}_k^l(t, \chi_k)$ is Lebesgue integrable.*

Proof. We apply results from [103]. The function $\lambda_{max}(L_l(t, \chi_k))$ is measurable in t for any fixed χ_k, and continuous in χ_k for each $t \in (t_k, t_{k+1})$. Therefore, it qualifies as a normal integrand [103, Example 14.29]. According to Proposition 14.44, the composition $\max(0, \lambda_{max}(L_l(t, \chi_k)))^{p_l}$ is also a normal integrand. Then, by Theorem 14.56, the subgradient mappings $\partial_{\chi_k}\{\max(0, \lambda_{max}(L_l(t, \chi_k)))^{p_l}\}$ are closed-valued and measurable. Finally, Corollary 14.6 ensures that this measurable set-valued map admits a measurable selection. Because $L_l(t, \chi_k)$ is affine in χ_k (hence Lipschitz), λ_{max} is 1-Lipschitz on symmetric matrices, and the scalar maps $s \to \max(0, s)$ and $s \to s^{p_l}$ are Lipschitz on every bounded interval, their composition is Lipschitz on the compact set $[t_k, t_{k+1}] \times \mathrm{dom}(\chi)$ where $\mathrm{dom}(\chi) \subset \mathbb{R}^{n_{q_v} + n_{y_v} + n_{z_v}}$ denotes the bounded domain of χ_k. Consequently, its subgradient is bounded. Hence, the measurable selection $\tilde{g}_k^l(t, \chi_k)$ is integrable. □

While Lemma 26 guarantees that a integrable selection $\tilde{g}_k^l(t, \chi_k)$ exists almost everywhere on $[t_k, t_{k+1}]$, the lemma is purely existential; it does not specify which subgradient is selected or how to compute it. For implementation, we choose a specific subgradient of the integrand $\max(0, \lambda_{max}(L_l(t, q_v, \zeta_v)))^{p_l}$ at $\bar{\chi}_k$, obtained from the chain rule. Let v_i be the eigenvectors associated with $\lambda_{max}(L_l(t, \chi_k))$ and $M(t, \chi_k)$ the algebraic multiplicity of $\lambda_{max}(L_l)$. Define the matrix

$$W(t, \chi_k) = \frac{1}{M(t, \chi_k)} \sum_{i:\lambda_i = \lambda_{max}(L_l)} v_i v_i^\top,$$

which averages the outer products of the eigenvectors associated with the largest eigenvalue. The subgradient we use at $\bar{\chi}_k$ is

$$\tilde{g}_k^l(t, \chi_k) = \left[\frac{\partial \max(0, \lambda_{max}(L_l(t)))^{p_l}}{\partial \lambda_{max}(L_l(t))} \cdot \frac{\partial \mathrm{vec}(L_l(t))}{\partial \chi_k} \cdot \mathrm{vec}(W(t, \chi_k)) \right]_{\chi_k = \bar{\chi}_k}, \tag{4.19}$$

where $L_l(t, \chi_k)$ is compactly written as $L_l(t)$. When $p_l > 1$ and the largest eigenvalue of L_l is simple ($M(t, \chi_k) = 1$), the integrand $\max(0, \lambda_{max}(L_l(t, q_v, \zeta_v)))^{p_l}$ is differentiable. In that case the expression above coincides with the exact gradient of the integrand.

4.2.4 Successive convexification with subgradients

We now apply the successive convexification (SCvx) framework proposed in [40] to solve the convex constraint $h_k^l(\chi_k) \leq 0$, using the subgradient $g_k^l(\chi_k)$ derived in the previous subsection. The SCvx method can be interpreted as a combination of exact penalization and prox-linear method [39] that is sequential convex programming for convex-composite minimization. The goal is to solve the discrete funnel synthesis problem with the ϵ-relaxed CTCS constraint:

$$\min_{q_{v,k}, \zeta_{v,k}^m, \zeta_{v,k}^p} \quad J(Q_0) \tag{4.20a}$$

$$(4.7), (4.13), (4.14\text{d}), (4.14\text{e}) \tag{4.20b}$$

$$h_k^l(\chi_k) \leq \epsilon, \quad \forall k \in \mathbb{Z}_{[0, N-1]}, \tag{4.20c}$$

where $\epsilon \in \mathbb{R}_{++}$ is a small constant introduced to ensure linear independence constraint qualification (LICQ), as discussed in [40, Lemma 10]. In practice, ϵ is typically chosen to

be sufficiently small, for example, less than 10^{-6}, so that it does not significantly affect the quality of continuous-time constraint satisfaction. Note that the constraints in (4.20b) include those already present in the discrete funnel synthesis problem (4.14).

At each iteration, the original constraint for the CTCS is replaced by its first-order approximation using a subgradient. Specifically, at the i-th iteration, the following convex SDP subproblem is solved:

Problem 4.10. *(SDP subproblem)*

$$\min_{q_{v,k},\zeta^m_{v,k},\zeta^p_{v,k}} J(Q_0) + w_h \sum_{k=0}^{N-1} \max(0, v^l_k) + w_{tr} \left(\sum_{k=0}^{N-1} \|\chi_k - \bar{\chi}_k\|_2^2 + \|q_{v,N} - \bar{q}_{v,N}\|_2^2 \right) \tag{4.21a}$$

$$(4.7), (4.13), (4.14\text{d}), (4.14\text{e}), \tag{4.21b}$$

$$h^l_k(\bar{\chi}_k) + g^l_k(\bar{\chi}_k)^\top (\chi_k - \bar{\chi}_k) - \epsilon = v^l_k, \quad \forall k \in \mathbb{Z}_{[0,N-1]}. \tag{4.21c}$$

The reference solutions $\bar{\chi}_k$ for all $k \in \mathbb{Z}_{[0,N-1]}$ and $\bar{q}_{v,N}$ come from an initial guess or the previous iteration's solution. The penalization terms in the objective reflect two key components of the SCvx method. The first term weighted by $w_h \in \mathbb{R}_{++}$ enforces the linearized constraint through exact penalization. The second term weighted by w_{tr} promotes proximity to the linearization point $\bar{\chi}_k$ by penalizing the trust-region. This term also aligns the subproblem structure with the prox-linear method framework [39].

Remark 4.11. *When the subgradient $g^l_k(\bar{\chi}k)$ coincides with the true gradient, that is, when the maximum eigenvalue of $L_l(t, \chi_k)$ is simple for almost every $t \in [t_0, t_f]$, then a convergence guarantee is available from [40, Theorem 30]. In such cases, the Scvx method provably converges to a solution of* (4.20)*, under mild assumptions. However, since we employ subgradients instead of exact gradients, a general convergence guarantee remains an open question. Nevertheless, this approach offers a practical and meaningful extension of existing gradient-based methods to settings where differentiability cannot be assumed in enforcing continuous-time constraint satisfaction.*

4.2.5 Summary of algorithm

We summarize the proposed method for solving the continuous-time funnel synthesis problem with CTCS. The method proceeds in two stages. First, we solve the discrete

funnel synthesis problem with additional constraint enforcement at a finite number of N_s intermediate constraint-checking points, as described in Section 4.2.2. The resulting solution is then used as an initial guess for the Scvx method with subgradients, presented in Section 4.2.4. The overall algorithm is as follows:

Algorithm 1 Funnel Synthesis with SCvx

1: Choose N, N_s, and initialize problem data
2: Solve the discrete funnel synthesis problem (4.14) with additional constraint enforcement (4.17) at N_s intermediate points per subinterval
3: Set an initial guess $\bar{\chi}_k$ for all $k \in \mathbb{Z}_{[0,N-1]}$, and $\bar{q}_{v,N}$
4: **for** $i = 1, \ldots, N_{iter}$ **do**
5: Compute the subgradient $g_k^l(\bar{\chi}_k)$ for each constraint h_k^l using (4.19)
6: Solve the SDP subproblem (4.21).
7: Update the solutions $\bar{\chi}_k \leftarrow \chi_k$ for all $k \in \mathbb{Z}_{[0,N-1]}$ and $\bar{q}_{v,N} \leftarrow q_{v,N}$.
8: **end for**
9: Return solution variables

4.3 Numerical examples

In this section, we validate the proposed method through two numerical examples involving obstacle avoidance: control of a unicycle and a 6-DoF quadrotor. In both cases, a Mosek solver is used to solve the SDP (4.14) and (4.21). All simulations are written in Julia and executed on a MacBook with an Apple M1 Pro processor.

4.3.1 Unicycle

We consider a unicycle model illustrated in Example 3.4. The simulation environment for all unicycle experiments is configured as follows. The time horizon is set to $t_0 = 0$ and $t_f = 5$ seconds, uniformly divided into $N = 9$ subintervals. Two circular obstacles, each with a radius of 0.5 m, define state constraints, and input constraints are given by $0 \leq u_1 \leq 2$ (m/s) and $|u_2| \leq 2$ (rad/s), as specified in (3.27). The cost function is chosen as $\text{trace}(Q_{w0}^{-1})$, with the weight matrix $W_0 = I_3$ as defined in (3.31). The nominal trajectory

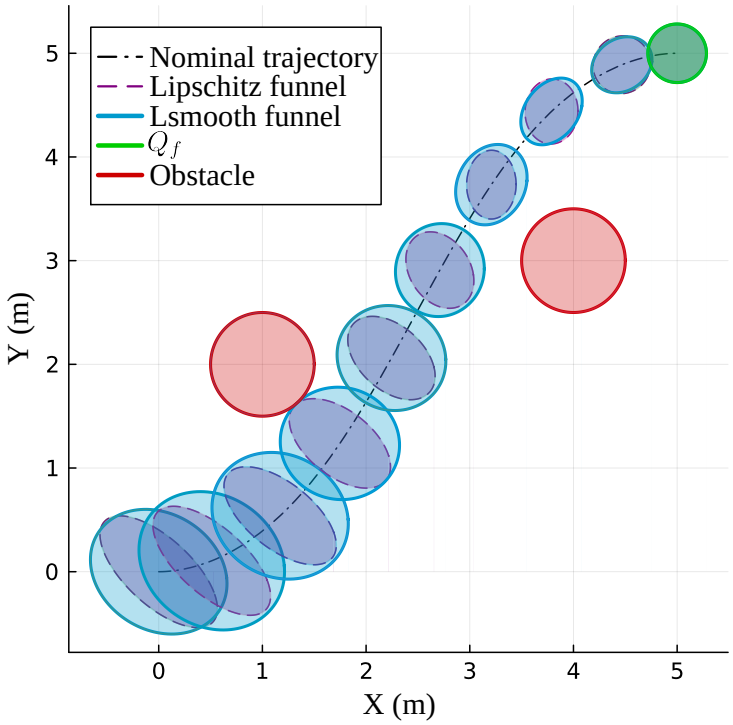

Figure 4.1: The synthesized state funnel projected on x (x_1) and y (x_2) position coordinates.

starts at $\bar{x}(t_0) = (0, 0, 0)$ and ends at $\bar{x}(t_f) = (5, 5, 0)$. The final funnel matrix Q_f, used as the terminal constraint in (4.14e), is set to diag(0.08, 0.08, 0.06).

Comparison of funnels computed under Lipschitz and L-smooth conditions

We first present results comparing two modeling approaches for the nonlinearity: the Lipschitz condition (3.19) and L-smooth condition (3.20). Specifically, we consider the use of global constants, where γ_i and β_i are valid over the entire domain $\Omega = \mathbb{R}^{n_{q_i}}$ in (3.19)and (3.20) for each $i = 1, 2$ and $t \in [t_0, t_f]$. As discussed in Example 3.9, the input constraint $|u_1| \leq 2$ implies that the global constants are $\gamma_i = 4$ and $\beta_i = 2$ for all $i = 1, 2$. For the L-smooth case, λ_i^β is set to 0.3 for all $i = 1, 2$, as discussed in (3.35). In this comparison, the bounded disturbance is set to zero $w(t) = 0$ for all $t \in [t_0, t_f]$, and Direct-type funnel

dynamics (4.2a) is used. The decay rate α is set to zero, as invariance can be guaranteed in the absence of bounded disturbances, as discussed in Remark 3.17. With zero disturbance, the level constant c_Q in (3.6) can be any positive value, and it is chosen to be 1 in this case.

The synthesized funnel using $\beta_i(t) = 2$ for all $t \in [t_0, t_f]$ is shown in Figure 4.1. It is found that the SDP problem (4.14) becomes infeasible when using the global Lipschitz constant, indicating that this formulation is overly conservative compared to the L-smooth case. To continue the comparison, we reduce the Lipschitz constant until the SDP becomes feasible, which occurs at $\gamma_i = 0.7$, which a value significantly smaller than the true global value. The resulting funnel is also shown in Figure 4.1. Notably, even though the L-smooth formulation uses the exact global value, it yields a larger funnel with a cost of 8.17. In contrast, the funnel computed with the Lipschitz condition has a significantly higher cost of 14.76, further demonstrating that the L-smooth condition is much less conservative.

To further compare the Lipschitz and L-smooth conditions, we consider the use of local constants where the domain Ω is strictly subset of Euclidean space. To this end, we first compute a funnel without considering the nonlinearity, which is equivalent to setting $\gamma_i = 0$ for all $i = 1, 2$. Since the nonlinearity is ignored, the funnel is larger, but the invariance no longer holds exactly for the original nonlinear system. This funnel is only used to collect state and input samples, from which the Lipschitz and L-smooth constants are estimated for each subinterval; the corresponding values γ_k and β_k are used in the funnel computation (4.14). As the estimation of these constants is not the focus of this paper, we refer the reader to [101, 97] for further details. For the L-smooth case, λ_i^β is set to 1.0 for all $i = 1, 2$. The resulting funnel are shown in Figure 4.2. It is shown that the L-smooth case achieves a larger funnel with a lower cost 2.96 compared to the Lipschitz case 5.11, again illustrating the reduced conservatism of the L-smooth condition.

Comparison of funnel dynamics

In the next simulations, we aim to compare the choice of funnel dynamics: Lyapunov-type (4.1a) and Direct-type (4.2a). For this comparison, we choose the use of global L-smooth constants $\beta_i = 2$ for all $i = 1, 2$. The bounded disturbance is ignored, and λ_i^β is set to

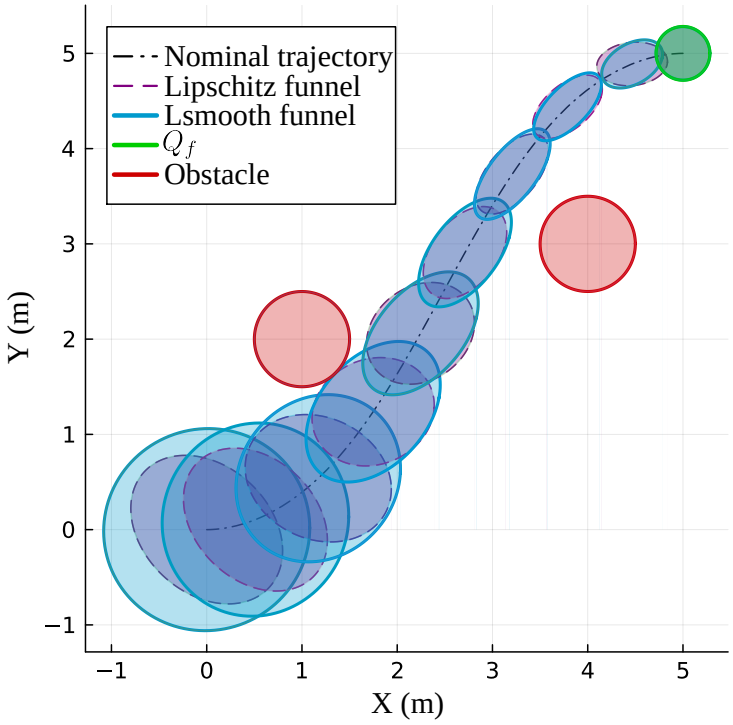

Figure 4.2: The synthesized state funnel with local Lipschitz and L-smooth constants.

0.3 for all $i = 1, 2$. The computed funnels are illustrated in Fig. 4.3. It is shown that the funnel computed with the Lyapunov-type dynamics achieves a larger entry size with a lower cost 7.17 compared to the funnel with the Direct-type dynamics with a cost 8.17. We conjecture that this is because the Lyapunov-type formulation effectively encodes the DLMI (3.25) through a Lyapunov differential equation (4.1a), allowing the funnel's evolution over time to be captured more naturally and smoothly. In contrast, the direct-type formulation constrains the funnel evolution to follow a second-order hold structure, which may limit its expressiveness. However, as discussed in Section 4.1.3, the solution of the Lyapunov-type dynamics does not preserve the positive definiteness of Q. Figure 4.4 shows the time evolution of the minimum eigenvalues of $Q(t)$ for both funnel dynamics formulations. The results reveal that, for the Lyapunov-type dynamics, the minimum eigenvalue occasionally falls below zero, indicating a loss of positive definiteness and hence invalidating the funnel. In contrast, as guaranteed by Lemma 4.2, the minimum eigenvalue remains strictly positive under the Direct-type dynamics, ensuring the funnel's validity throughout the time horizon.

CTCS by introducing intermediate constraint-checking points

Next, we demonstrate the effectiveness of introducing intermediate constraint-checking points as described in (4.17). In this simulation, we use the Lyapunov-type funnel dynamics, and all other settings are kept identical to the previous experiment. We add $N_s = 4$ checking points per subinterval. The time evolution of the maximum eigenvalues of each pointwise-in-time LMI constraint from (4.12) is shown in Figure 4.5. As illustrated, for every constraint L_l, the maximum eigenvalue remains below zero across the entire time horizon, indicating that CTCS is achieved.

Figure 4.6 compares the computational cost, measured by Mosek's solve time, of two strategies for improving CTCS: increasing the number of node points and introducing intermediate constraint-checking points. While both approaches enhance constraint coverage, increasing node points enlarges the decision variable space, which leads to higher computational complexity. In contrast, adding intermediate checking points preserves the number of decision variables and only increases the number of LMI constraints. Although

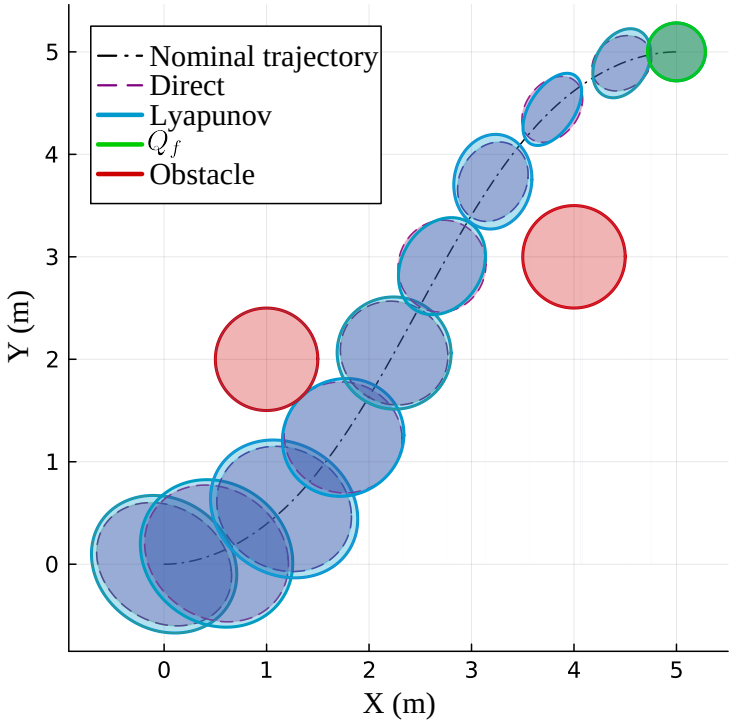

Figure 4.3: The synthesized state funnel without considering nonlinearity.

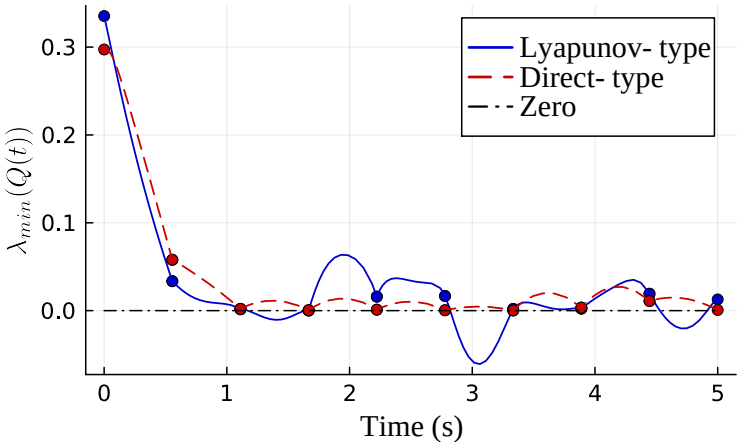

Figure 4.4: The time history of minimum eigenvalues of $Q(t)$.

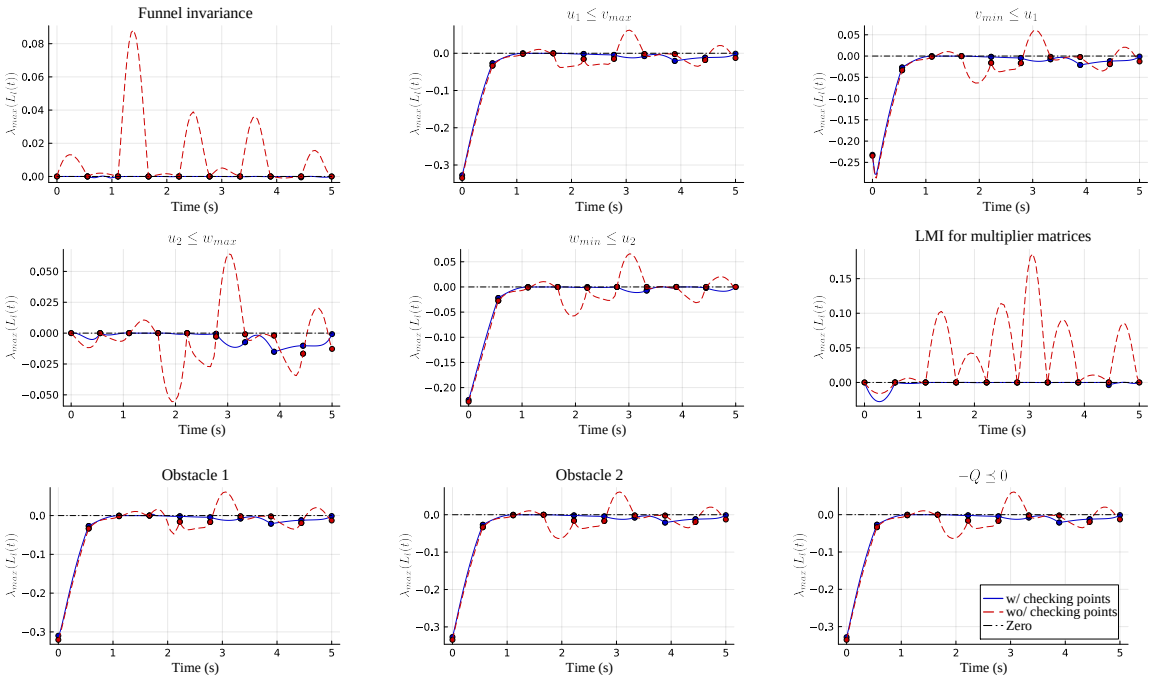

Figure 4.5: Time evolution of the maximum eigenvalue of each pointwise-in-time LMI constraint $L_l(t)$ associated with the corresponding constraint labeled in each subplot title.

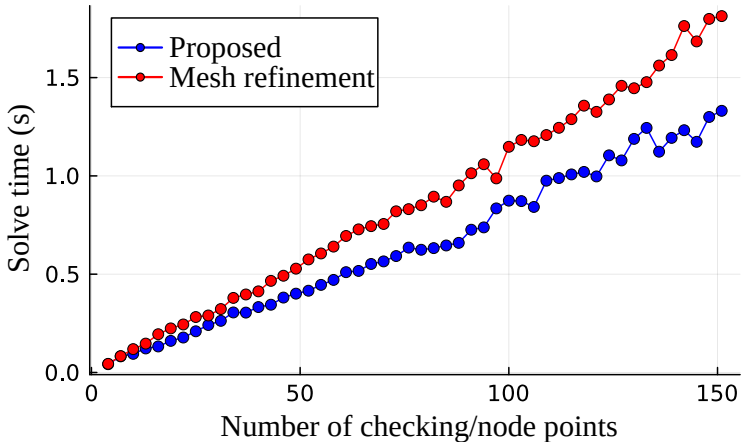

Figure 4.6: Mosek solve time versus number of constraints for two approaches: increasing node points and adding intermediate constraint-checking points (proposed).

the solve-time difference may appear modest, this is partly due to Mosek's internal handling, where it introduces auxiliary variables for the added constraints. Still, the intermediate-checking-point method consistently achieves lower solve times and avoids increasing the control update rate, which is often a desirable property in practice.

CTCS by SCvx with subgradients

In the following simulation, we validate the proposed SCvx-based approach with subgradients for CTCS, as described in Section 4.2.4. We employ the Direct-type funnel dynamics, while keeping all other settings identical to the previous experiment. As noted in Section 4.2.5, the solution to the problem (4.14) with one intermediate checking point ($N_s = 1$) for each subinterval is used as the initial guess for the SCvx iteration. The weights w_{vc} and w_{tr} in (4.21a) are set to 20 and 10^5, respectively. A total of 30 iterations are performed, with an average Mosek solve time of 0.2 seconds per iteration. The small epsilon in (4.21c) is set to $\epsilon = 10^{-5}$. The time evolution of the maximum eigenvalues of the pointwise-in-time LMI constraints associated with the invariance condition (4.2b) and the validity of the multiplier matrices (3.35c) is shown in 4.7. The plots indicate that as

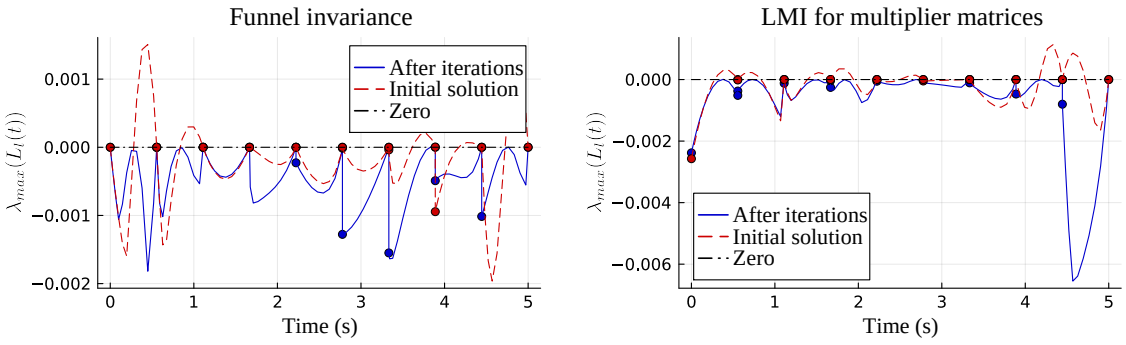

Figure 4.7: Time evolution of maximum eigenvalues for pointwise-in-time LMI constraints in the SCvx-based CTCS approach. Left: Funnel invariance constraint (4.2b), Right: valid multiplier constraint (3.35c).

the iterations proceed, the constraint violation between node points diminishes. The trust region penalization term in (4.21a) is plotted in Figure 4.8. The values are normalized such that the initial iteration starts at 1, for the purpose of clearly illustrating the convergence behavior.

Funnel under bounded disturbance

Next, we consider the case where the bounded disturbance exerts to the unicycle model:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} u_1 \cos x_3 \\ u_1 \sin x_3 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0.02 w_1 \\ 0.02 w_2 \\ 0 \end{bmatrix},$$

where the state and input variables are the same as those in Example (3.4), and $w = [w_1, w_2]^\top$ represents a bounded disturbance with $w_{max} = 1$ such that $\|w\|_2 \leq 1$. Following the discussion in Lemma 13.10, the level constant c_Q in the state funnel definition (3.6) is chosen to $c_Q = w_{\max}^2$. The decay rate is set to$\alpha = 0.1$. The Direct-type funnel dynamics 4.2a is employed and two intermediate checking points ($N_s = 2$) are used for each subinterval. All other settings including the cost function, constraint formulations, and simulation parameters are kept identical to those used in the previous undisturbed simulations.

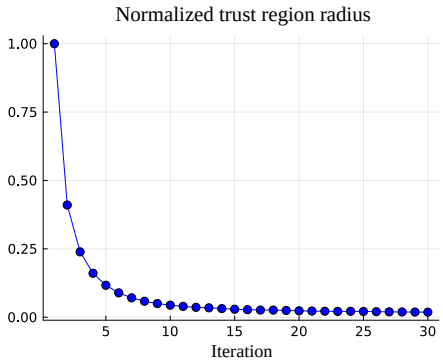

Figure 4.8: Normalized trust region penalization term from the SCvx cost function (4.21a) over 30 iterations. The values are normalized to 1 at the first iteration to clearly illustrate the convergence behavior.

The computed the state funnel centered around the nominal state is illustrated in Figure 4.9. It can be seen that the computed funnel without the disturbance is larger than the funnel computed under disturbance. This is expected, as the presence of external disturbances necessitates a more conservative funnel to ensure invariance is maintained despite the disturbance. The computed input funnel centered around the nominal input is illustrated in Figure 4.10. The result shows that the input funnel projected onto each input dimension satisfies the input limit constraints over entire horizon.

Finally, we check the invariance of the funnel by generating 100 sample trajectories. The initial states are randomly sampled from the boundary of the computed funnel at t_0, and the synthesized controller is applied to generate the trajectories. For each sample, a bounded disturbance signal $w(t)$ is randomly generated once at the beginning and held constant over the entire time horizon. This choice of fixed-in-time disturbance is intended to create a more persistent influence on the system, which is shown to be more disruptive than rapidly changing noise. The resulting trajectories are shown in Figure 4.9. To verify funnel invariance, the corresponding Lyapunov function values $V(t, \eta(t))$ are plotted in Figure 4.11. All values remain strictly below the funnel level constant c_Q, confirming that the invariance condition is satisfied across all sampled trajectories.

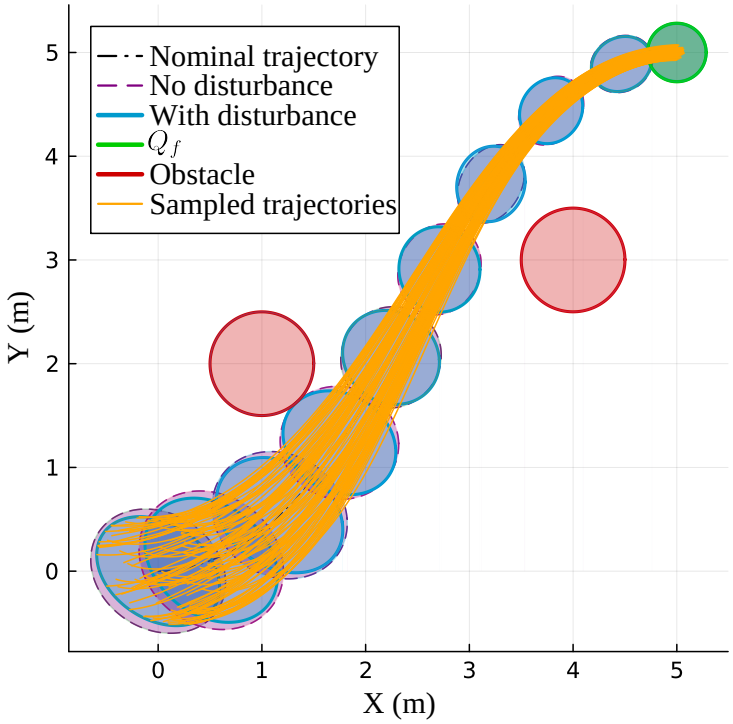

Figure 4.9: Comparison of synthesized state funnels with and without bounded disturbance; sampled trajectories are generated using the funnel synthesized under disturbance.

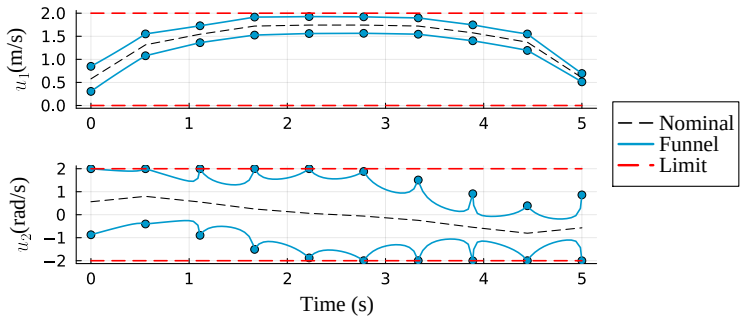

Figure 4.10: The synthesized input funnel projected onto each input dimension.

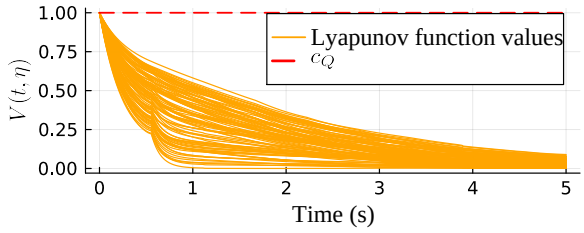

Figure 4.11: The Lyapunov function values of sampled trajectories.

4.3.2 6-DoF quadrator

We consider a 6-DoF quadrotor in the East-North-Up (ENU) inertial frame.. The state is $x = (r_p, v_p, \Phi, \Omega)$ where $r_p \in \mathbb{R}^3$ is the position vector, $v_p \in \mathbb{R}^3$ is the linear velocity, $\Phi \in \mathbb{R}^3$ is the Euler angles, and $\Omega \in \mathbb{R}^3$ is the body angular velocity. The control input is $u = (F_z, \tau_x, \tau_y, \tau_z)$ where F_z is the thrust along the body z-axis, and (τ_x, τ_y, τ_z) are the body-frame torques. The system model is given by

$$\begin{bmatrix} \dot{r_T} \\ \dot{v_T} \\ \dot{\Phi} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} v_T \\ \frac{1}{m} C_{I/B}(\Phi) F_B + g \\ R(\Phi)\Omega \\ J^{-1}\big(\tau - \Omega \times J\Omega\big) \end{bmatrix},$$

where $F_B = [0, 0, F_z]^\top$, $C_{I/B}(\Phi)$ is the rotation matrix from body to inertia frame, $R(\Phi)$ is the Euler-angle kinematics transformation. Here the following parameters, input constraint set $\mathcal{U}$, and the final funnel matrix Q_f are used

$$\begin{aligned} m &= 1.325 \text{ (kg)}, \ g = [0, 0, 9.81]^\top (\text{m/s}^2), \\ J &= \text{diag}\{0.03843, 0.02719, 0.060528\} \text{ (kgm}^2), \\ \mathcal{U} &= \{u \mid u_{\text{lb}} \le u \le u_{\text{ub}}\}, \\ u_{\text{ub}} &= (18, 0.1, 0.1, 0.1), \ u_{\text{lb}} = (0.0, -0.1, -0.1, -0.1) \\ Q_f &= \text{diag}\{0.2^2, 0.2^2, 0.2^2, 0.1^2, 0.1^2, 0.1^2, (5^\circ)^2, (5^\circ)^2, (5^\circ)^2, (2^\circ)^2, (2^\circ)^2, (2^\circ)^2\}. \end{aligned}$$

The state constraint corresponds to obstacle avoidance as depicted in Figure 4.12.

For the 6-DoF quadrotor dynamics, the first three state equations, corresponding to the position r_T, are linear in the states and inputs. Therefore, they do not contribute to the nonlinear term $\phi(\cdot)$, and the corresponding rows of the matrix E are zero. Specifically, we have $E = \begin{bmatrix} 0_{3\times 12} \\ I_9 \end{bmatrix}$. We partition the nonlinearities into three channels, each associated with a specific subset of states and inputs:

$$q_{[1]} = \begin{bmatrix} \Phi \\ F_z \end{bmatrix}, \quad q_{[2]} = \begin{bmatrix} \Phi_{1:2} \\ \Omega_{2:3} \end{bmatrix}, \quad q_{[3]} = \Omega,$$

where $\Phi_{1:2}$ denotes the first two components of Φ and $\Omega_{2:3}$ denotes the second and third components of Ω. The functions $\phi_{[1]}, \phi_{[2]}, \phi_{[3]} \in \mathbb{R}^3$ correspond to the nonlinear terms in the dynamics of v_T, Φ, and Ω, respectively. We use the L-smooth nonlinearity characterization, with the constants β_1, β_2, and β_3 set to 20, 5, and 1, respectively, estimated via sampling. The number of subintervals is set to $N = 15$. The nominal trajectory, illustrated in Figure 4.12, starts near $(-3, 4)$ follows a star-shaped path, and returns to its starting point. The total time of flight is around 15.78 seconds.

We compare two CTCS approaches: (i) introducing intermediate checking points, and (ii) applying SCvx with the subgradient, illustrated in Section 4.2.2 and 4.2.4, respectively. Both cases employ the Direct-type funnel dynamics and have a total 15 time subintervals $N = 15$. For the intermediate checking point method, we set $N_s = 2$. For SCvx, we use the initial guess as the solution of (4.14) without intermediate checking points. The weights w_{vc} and w_{tr} in (4.21a) are set to 2×10^5 and 2×10^3, respectively, with $\epsilon = 10^{-3}$. The state funnels projected onto the x–y coordinates are shown in Figure 4.12. The resulting costs for the two methods are 76.987 and 75.82, respectively. The total computation time for the former is 68.30, while for the SCvx method, the average computation time per iteration is 23.36 seconds over a total of 50 iterations. The time evolution of the maximum eigenvalues of the pointwise-in-time LMI constraints associated with the invariance condition (4.2b) and the validity of the multiplier matrices (3.35c) is shown in Figure 4.13.

In both methods, the constraint violation between node points is successfully reduced. For the quadrotor case, the problem dimension is significantly larger than in the unicycle example, both in terms of the system size and the number of discretization nodes N,

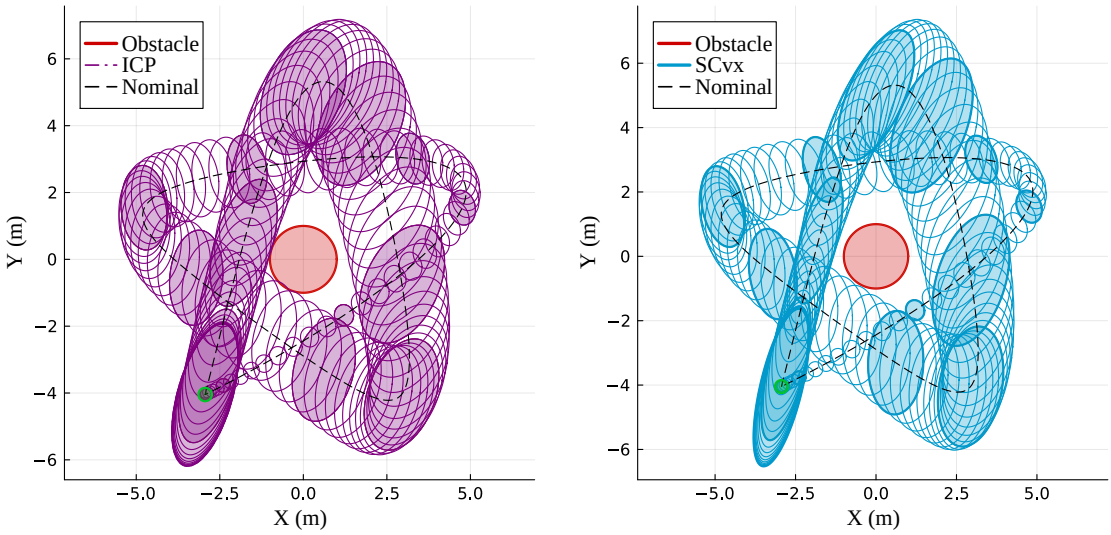

Figure 4.12: State funnels projected onto the x and y plane. Left: result obtained using intermediate checking points. Right: result obtained using the SCvx approach. Filled ellipsoids indicate funnels at discrete node points, while unfilled ellipsoids depict intermediate funnels between nodes.

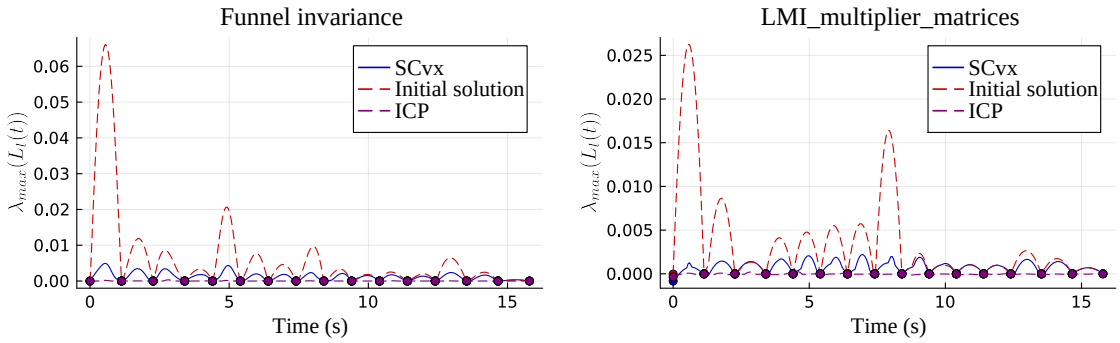

Figure 4.13: Time evolution of the maximum eigenvalue of each pointwise-in-time LMI constraint $L_l(t)$ associated with the corresponding constraint labeled in each subplot title.

leading to increased computational cost. In this setting, the iterative nature of SCvx can be computationally intensive, making the one-shot SDP approach with intermediate checking points a more efficient alternative.

Numerical simulations showed that funnel computation under the L-smooth condition yields less conservative results compared to the Lipschitz condition. Between two funnel dynamics formulations, the Lyapunov-type is not able to preserve the positive definiteness of Q as anticipated, while the direct-type preserved it as proven. Both CTCS methods, intermediate checking points and SCvx with subgradients, successfully reduced constraint violations between node points. For high-dimensional and long-horizon problems such as the quadrotor case, the repeated SDP solving required by SCvx was computationally heavy, making the intermediate-checking-point approach more efficient in practice.

Chapter 5

CONTINUOUS-TIME INVARIANCE BY LMI COPOSITIVITY CONDITIONS

This chapter proposes a computational funnel synthesis method that guarantees continuous-time invariance of the funnel. A finite set of linear matrix inequalities (LMIs) is derived to imply satisfaction of a differential linear matrix inequality (DLMI), ensuring funnel invariance over the entire time horizon. To achieve this, the uncertain LTV system is approximated as an uncertain linear parameter-varying (LPV) system, where the approximation error is modeled as a state- and input-dependent uncertainty. Based on this approximation, matrix copositivity conditions are employed to certify continuous-time invariance without requiring constraint sampling or reformulation. Compared to the two approaches in Chapter 4, this method offers the advantage of maintaining a fixed number of LMI constraints and requires solving a single semidefinite program (SDP), avoiding iterative procedures like SCvx. However, this benefit comes at the cost of increased conservatism in the resulting funnel. The method is validated through two case studies: a three-dimensional trajectory planning and control problem with obstacle avoidance, and a six-degree-of-freedom powered descent guidance problem.

Chapter-specific notation. The set notations $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{R}^n$ are the sets of real, nonnegative real, integer, and the n-dimensional Euclidean spaces, respectively. Intervals are written by $\mathbb{Z}_{[a,b)} = \{z \in \mathbb{Z} : a \leq z < b\}$. The symmetric matrix $Q = Q^\top (\succeq) \succ 0$ implies Q is a positive (semi) definite PD (PSD) matrix. The set of positive (semi) definite matrices whose size is $n \times n$ are denoted by $\mathbb{S}^n_{++}(\mathbb{S}^n_+)$. The identity matrix having $n \times n$ size is denoted by I_n. The subscript and the time argument will be omitted when it is clear from the context. The notation $\mathcal{L}_2[a,b]$ is a set of Lebesgue measurable functions $x(t)$ defined on an interval $[a,b] \subset \mathbb{R}$ such that $\left(\int_a^b x(t)^\top x(t) \mathrm{d}t\right)^{1/2} < \infty$. A property is said to hold almost everywhere, or for almost every $x \in X$ with some set X if the set on which it fails is a

Lebesgue measure zero set. We abbreviate the notations $A^\top PA$ and $\begin{bmatrix} a & b^\top \\ b & c \end{bmatrix}$ as $(\star)^\top PA$ and $\begin{bmatrix} a & \star \\ b & c \end{bmatrix}$, respectively. The Minkowski sum is denoted by $\oplus$. The stacking of vectors x and y is denoted by $(x, y) = [x^\top y^\top]^\top$. The operation $\mathrm{diag}(\cdot)$ outputs a diagonal matrix formed from a given vector.

5.1 Invariance of funnel

5.1.1 Nonlinear systems

Consider the finite-horizon continuous-time nonlinear systems in the form of

$$\dot{x}(t) = f(t, x(t), u(t), w(t)), \quad \forall t \in [t_0, t_f], \tag{5.1}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state, $u(t) \in \mathbb{R}^{n_u}$ is the input, $w(t) \in \mathbb{R}^{n_w}$ is the (exogenous) disturbance, and t_0, t_f are the initial and final time. The system dynamics $f : \mathbb{R}_+ \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ are assumed to be continuous in t and continuously differentiable in x, u and w. We assume that the input signal $u(\cdot) \in \mathcal{L}_2^{n_u}[t_0, t_f]$ and the disturbance signal $w(\cdot) \in \mathcal{L}_2^{n_w}[t_0, t_f]$ are piecewise continuous. We further assume that the disturbance signal $w(\cdot)$ is essentially bounded from above by one, that is,

$$\|w(\cdot)\|_\infty \leq 1, \tag{5.2}$$

where $\|w(\cdot)\|_\infty \coloneqq \operatorname{ess\,sup}_{t\in[t_0,t_f]} \|w(t)\|_2$. Notice that the choice of the upper bound (i.e., one) is not restrictive, since if $w(\cdot)$ is bounded above by some constant $w_{max} \in \mathbb{R}_+$, we can redefine $w \leftarrow w/w_{max}$ so that the bound becomes one. We refer to a trajectory as a collection of the state, the input, and the disturbance signals, denoted together by $(x(\cdot), u(\cdot), w(\cdot))$. The nominal trajectory having the zero disturbance $(\bar{x}(\cdot), \bar{u}(\cdot), 0)$ is assumed to be given and dynamically feasible, that is, $\dot{\bar{x}} = f(t, \bar{x}, \bar{u}, 0)$ for all $t \in [t_0, t_f]$. With the given nominal trajectory, we can rewrite the system (5.1) in the linear fractional form

$$f(t, x, u, w) = A(t)x + B(t)u + F(t)w + E_o\phi(t, q) \tag{5.3a}$$

$$q = C_o x + D_o u + G_o w, \tag{5.3b}$$

where the function $\phi : \mathbb{R}_+ \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_\phi}$ represents the nonlinearity of the system and $q(t) \in \mathbb{R}^{n_q}$ is its argument. The matrices $A(t)$, $B(t)$, and $F(t)$ are chosen as first-order approximations of (5.1) around the nominal trajectory. We assume that $A(t)$, $B(t)$, and $F(t)$ are bounded for all $t \in [t_0, t_f]$. The constant matrices $E_o \in \mathbb{R}^{n_x \times n_\phi}$, $C_o \in \mathbb{R}^{n_q \times n_x}$, $D_o \in \mathbb{R}^{n_q \times n_u}$, and $G_o \in \mathbb{R}^{n_q \times n_w}$ are selector matrices, composed of 0s and 1s, chosen to structure the nonlinearity of the system.

Example 5.1. *The unicycle model can be written as*

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} u_1 \cos(x_3 + c_1 w_1) \\ u_1 \sin(x_3 + c_1 w_1) \\ u_2 + c_2 w_2 \end{bmatrix}, \tag{5.4}$$

where x_1 and x_2 are x- and y-positions, x_3 are the yaw angle, u_1 is the velocity control, and u_2 is the angular velocity control. The disturbances w_1 and w_2 affect to the yaw angle estimation and the angular velocity control, and $c_1, c_2 \in \mathbb{R}_+$ are system parameters. The argument q of ϕ can be chosen as $[x_3, u_1, w_1]$ with

$$C_o = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, D_o = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}, G_o = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Since the only first two components in f involve with the nonlinearity, the matrix E_o is $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^\top$. *The time-varying matrices $A(t)$, $B(t)$, and $F(t)$ are given as follows:*

$$A(t) = \begin{bmatrix} 0 & 0 & -\bar{u}_1(t) \sin \bar{x}_3(t) \\ 0 & 0 & \bar{u}_1(t) \cos \bar{x}_3(t) \\ 0 & 0 & 0 \end{bmatrix}, B(t) = \begin{bmatrix} \cos \bar{x}_3(t) & 0 \\ \sin \bar{x}_3(t) & 0 \\ 0 & 1 \end{bmatrix}, \tag{5.5a}$$

$$F(t) = \begin{bmatrix} -c_1 \bar{u}_1(t) \sin \bar{x}_3(t) & 0 \\ c_1 \bar{u}_1(t) \cos \bar{x}_3(t) & 0 \\ 0 & c_2 \end{bmatrix} \tag{5.5b}$$

where $\bar{x} = [\bar{x}_1, \bar{x}_2, \bar{x}_3]^\top$ and $\bar{u} = [\bar{u}_1, \bar{u}_2]^\top$ are the nominal state and input, respectively. The

nonlinear function ϕ that is the remainder term in (5.3) *is given by*

$$\phi(t,q) = \begin{bmatrix} c_1\bar{u}_1 w_1 \sin\bar{x}_3 - u_1\cos\bar{x}_3 + u_1\cos(c_1 w_1 + x_3) + \bar{u}_1 x_3 \sin\bar{x}_3, \\ -c_1\bar{u}_1 w_1 \cos\bar{x}_3 - u_1\sin\bar{x}_3 + u_1\sin(c_1 w_1 + x_3) - \bar{u}_1 x_3 \cos\bar{x}_3 \end{bmatrix},$$

where the time argument is omitted.

5.1.2 Incremental dynamical system and its LPV approximation

The incremental form of dynamics illustrates the behavior of the system (5.1) relative to the nominal trajectory. To derive it, we first define difference variables as

$$\eta(t) \coloneqq x(t) - \bar{x}(t), \quad \xi(t) = u(t) - \bar{u}(t), \quad \delta q(t) = q(t) - \bar{q}(t),$$

where $\bar{q}(t) \coloneqq C_o\bar{x}(t) + D_o\bar{u}(t)$. Having η as the state, the incremental dynamics can be derived as the following uncertain LTV system:

$$\dot{\eta} = A(t)\eta + B(t)\xi + F(t)w + E_o\left(\phi(t,q) - \phi(t,\bar{q})\right), \tag{5.6a}$$

$$\delta q = C_o\eta + D_o\xi + G_o w. \tag{5.6b}$$

Since the original system f in (5.1) is continuously differentiable, f is locally Lipschitz, thereby resulting in ϕ being locally Lipschitz with its second argument. Then, for each $t \in [t_0, t_f]$, there exists a local Lipschitz constant $\gamma_i(t) \in \mathbb{R}_+$ such that

$$\|\phi_i(t,q) - \phi_i(t,\bar{q})\|_2 \le \gamma_i(t)\|q(t) - \bar{q}(t)\|_2, \quad \forall i \in \mathbb{Z}_{[1,n_\phi]}, \quad \forall q, \bar{q} \in \mathcal{Q}, \tag{5.7}$$

for any compact set $\mathcal{Q} \subseteq \mathbb{R}^{n_q}$. Here the term $\phi(t,q) - \phi(t,\bar{q})$, due to the nonlinearity of (5.3), is a state-, input-, and disturbance-dependent uncertainty characterized by the constraint (5.7). This type of LTV system in incremental form with Lipschitz nonlinearity has been studied in the context of funnel computation in several existing works. See, for example, [97, 59] for related formulations and applications.

Now we describe an LPV approximation of the LTV system (5.6). The underlying motivation of this approximation is that we can derive a finite number of LMIs that imply the invariance of the funnel for the LPV system, which is not tractable for the LTV system.

We start by partitioning the time horizon $[t_0, t_f]$ into $N \in \mathbb{R}_+$ uniform subintervals using time nodes defined as:

$$t_k = t_0 + \frac{k}{N}(t_f - t_0), \quad \forall k \in \mathbb{Z}_{[0,N]}.$$

On each subinterval $[t_k, t_{k+1}]$, we approximate the time-varying system matrices $A(t)$, $B(t)$, and $F(t)$ using a first-order hold (FOH) approach, which linearly interpolates between their values at the endpoints of the interval. This results in convex combinations of the matrices at t_k and t_{k+1}. We define:

$$\tilde{\square}(t) \coloneqq \sigma_1^k(t)\square_k + \sigma_2^k(t)\square_{k+1}, \quad t \in [t_k, t_{k+1}] \tag{5.8a}$$

$$\sigma_1^k(t) = \frac{t_{k+1} - t}{t_{k+1} - t_k}, \quad \sigma_2^k(t) = \frac{t - t_k}{t_{k+1} - t_k}, \tag{5.8b}$$

where $\square_k = \square(t_k)$ and the placeholder $\square$ corresponds to A, B, and F. By applying (5.8) across all k in $\mathbb{Z}_{[0,N-1]}$, we obtain continuous piecewise linear approximations $(\tilde{A}(t), \tilde{B}(t), \tilde{F}(t))$ of the system matrices $(A(t), B(t), F(t))$ over the entire horizon $[t_0, t_f]$.

It is worth noting that the system matrices $(A(t), B(t), F(t))$ are not necessarily piecewise linear, and therefore the approximation error is generally nonzero. However, it is common in the literature (e.g., [97, 75]) to assume piecewise linearity to simplify funnel computation. In contrast, this paper does not assume zero approximation error; instead, the error is modeled explicitly as a state-, input-, and disturbance-dependent uncertainty.

Using (5.8), we can rewrite (5.6) equivalently as follows:

$$\dot{\eta} = \tilde{A}(t)\eta + \tilde{B}(t)\xi + \tilde{F}(t)w + \underbrace{\Delta_e(t)\begin{bmatrix} \eta \\ \xi \\ w \end{bmatrix}}_{\coloneqq e(t)} + E_o\delta\phi, \tag{5.9a}$$

$$\Delta_e(t) \coloneqq \begin{bmatrix} A(t) - \tilde{A}(t) & B(t) - \tilde{B}(t) & F(t) - \tilde{F}(t) \end{bmatrix}, \tag{5.9b}$$

where $\delta\phi(t, \delta q) \coloneqq \phi(t, q) - \phi(t, \bar{q})$. Note that two representations (5.6) and (5.9) are equivalent by the definition of the error term $e(t)$ in (5.9a). Since f is assumed to be continuously differentiable, the matrices $A(t)$, $B(t)$, and $F(t)$ are continuous, so bounded

within the finite interval $[t_0, t_f]$. Hence, there exists a positive constant $\beta_e(t) \in \mathbb{R}_+$ for each t such that

$$\|\Delta_e(t)\|_2 \leq \beta_e(t), \tag{5.10}$$

where $\|\cdot\|_2$ denotes the matrix 2-norm.

Alternatively, we can express the error term $e(t)$ more compactly as

$$e(t) = E_\Delta \Delta(t) q_\Delta(t)$$

where the matrix $E_\Delta \in \mathbb{R}^{n_x \times n_\Delta}$, the block-diagonal matrix $\Delta \in \mathbb{R}^{n_\Delta \times n_{q_\Delta}}$ of uncertainties, and the vector $q_\Delta \in \mathbb{R}^{n_{q_\Delta}}$ are given by

$$E_\Delta = \begin{bmatrix} E_\eta & E_\xi & E_w \end{bmatrix}, \quad \Delta = \begin{bmatrix} \Delta_\eta & 0 & 0 \\ 0 & \Delta_\xi & 0 \\ 0 & 0 & \Delta_w \end{bmatrix}, \quad q_\Delta = \begin{bmatrix} q_\eta \\ q_\xi \\ q_w \end{bmatrix},$$

with $n_\Delta \in \mathbb{R}_+$ and $n_{q_\Delta} = n_{q_\eta} + n_{q_\xi} + n_{q_w}$. These components are chosen such that

$$\begin{aligned} (A(t) - \tilde{A}(t))\eta &= E_\eta \Delta_\eta q_\eta, \\ (B(t) - \tilde{B}(t))\xi &= E_\xi \Delta_\xi q_\xi, \\ (F(t) - \tilde{F}(t))w &= E_w \Delta_w q_w, \end{aligned}$$

where the vectors q_η, q_ξ, and q_w are linear functions of η, ξ, and w, respectively, as follows:

$$q_\eta = C_\eta \eta, \quad q_\xi = D_\xi \xi, \quad q_w = G_w w.$$

In this formulation, the uncertainties are captured using the following inequalities:

$$\|\Delta_\square(t)\|_2 \leq \beta_\square(t), \tag{5.11}$$

where $\beta_\square(t) > 0$ bounds the corresponding uncertainty $\Delta_\square(t)$, with the placeholder $\square$ representing the subscripts η, ξ, and w. Compared to the single norm-bounded inequality in (5.10), these bounds in (5.11) can represent the uncertainly more compactly by exploiting its structure. More details on the structured uncertainty representation can be found in [99, 6.2.1], [37], [19].

Now we can summarize the incremental dynamics, written in the form of an uncertain LPV system, as follows:

$$\dot{\eta} = \tilde{A}(t)\eta + \tilde{B}(t)\xi + \tilde{F}(t)w + Ep(t,r), \tag{5.12a}$$

$$r \coloneqq C\eta + D\xi + Gw = [q_\Delta^\top, \delta q^\top]^\top \tag{5.12b}$$

$$p \coloneqq \begin{bmatrix} \Delta(t)q_\Delta(t) \\ \delta\phi(t,\delta q) \end{bmatrix}, \quad E \coloneqq \begin{bmatrix} E_\Delta & E_o \end{bmatrix}, \tag{5.12c}$$

$$C \coloneqq \begin{bmatrix} C_\eta \\ 0 \\ 0 \\ C_o \end{bmatrix}, D \coloneqq \begin{bmatrix} 0 \\ D_\xi \\ 0 \\ D_o \end{bmatrix}, G \coloneqq \begin{bmatrix} 0 \\ 0 \\ G_w \\ G_o \end{bmatrix}. \tag{5.12d}$$

All uncertainties in the system (5.12) are lumped into the vector $p(t,r) \in \mathbb{R}^{n_p}$ where $n_p = n_\Delta + n_\phi$ and $r(t) \in \mathbb{R}^{n_r}$. We characterize the uncertain term p using the following quadratic inequality (QI) [3, 4] for each t:

$$\begin{bmatrix} r(t) \\ p(t,r) \end{bmatrix}^\top M(t) \begin{bmatrix} r(t) \\ p(t,r) \end{bmatrix} \geq 0, \tag{5.13}$$

where the matrix $M(t) \in \mathbb{R}^{(n_r+n_p)\times(n_r+n_p)}$ is called a multiplier matrix. It follows from (5.7) and (5.10) that the valid multiplier matrix has the form of

$$M(t) = \begin{bmatrix} N_1(t)^{-1} & 0 \\ 0 & -N_2(t)^{-1} \end{bmatrix}, N_1(t) \coloneqq \begin{bmatrix} N_1^\beta(t) & 0 \\ 0 & \frac{1}{n_\phi}\lambda_\gamma(t) I_{n_q} \end{bmatrix}, \tag{5.14a}$$

$$N_2(t) \coloneqq \begin{bmatrix} N_2^\beta(t) & 0 \\ 0 & \lambda_\gamma(t)\Gamma(t)^2 \end{bmatrix}, N_1^\beta(t) \coloneqq \mathrm{diag}(\lambda_{\beta_\eta}(t)I, \lambda_{\beta_\xi}(t)I, \lambda_{\beta_w}(t)I), \tag{5.14b}$$

$$N_2^\beta(t) \coloneqq \mathrm{diag}(\lambda_{\beta_\eta}(t)\beta_\eta^2(t)I, \lambda_{\beta_\xi}(t)\beta_\xi^2(t)I, \lambda_{\beta_w}(t)\beta_w^2(t)I), \tag{5.14c}$$

where $\Gamma(t) = \mathrm{diag}(\gamma_1(t), \ldots, \gamma_{n_\phi}(t))$ and any positive real-valued functions $\lambda_{\beta_\square}(t) > 0$ with the placeholder $\square$ representing η, ξ, and w, and $\lambda_\gamma(t) > 0$. This can be verified by deriving

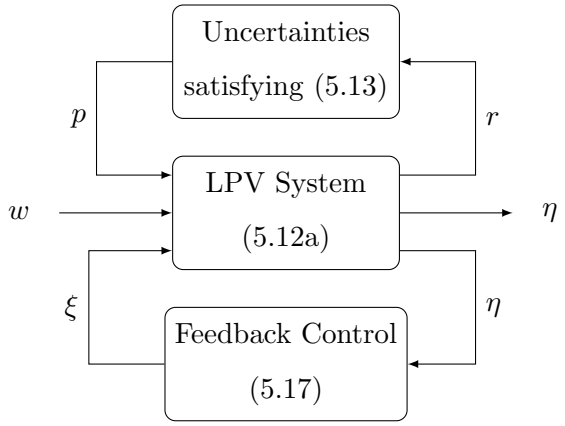

Figure 5.1: Uncertain LPV system interconnection with feedback control.

the following:

$$\begin{bmatrix} r \\ p \end{bmatrix}^\top M \begin{bmatrix} r \\ p \end{bmatrix} = \sum_{\square=\{\eta,\xi,w\}} \frac{1}{\lambda_{\beta_\square}} (q_\square^\top q_\square - \frac{1}{\beta_\square^2} q_\square^\top \Delta_\square^\top \Delta_\square q_\square) + \frac{1}{\lambda_\gamma} \left(n_\phi \delta q^\top \delta q - \delta\phi^\top \Gamma(t)^{-2} \delta\phi \right) \geq 0,$$

where the last inequality comes from (5.7) and (5.10). The resulting interconnection in the uncertain LPV system is illustrated in Figure 5.1.

5.1.3 Invariance condition

We consider a scalar-valued continuous Lyapunov function $V : \mathbb{R} \times \mathbb{R}^{n_x} \to \mathbb{R}$ defined by

$$V(t, \eta(t)) = \eta(t)^\top Q(t)^{-1} \eta(t), \tag{5.15}$$

where $Q(t) \in \mathbb{S}_{++}^{n_x}$ is a continuous-time PD matrix-valued continuous function. A *state funnel* is defined as a 1-sublevel set of V, that is,

$$\mathcal{E}_Q(t) \coloneqq \{\eta \mid \eta^\top Q(t)^{-1} \eta \leq 1\}. \tag{5.16}$$

We employ the following linear time-varying feedback controller:

$$\xi(t) = K(t)\eta(t), \tag{5.17}$$

for the system (5.12) with the feedback gain $K(t) \in \mathbb{R}^{n_u \times n_x}$. With this linear feedback, ξ becomes linear in η. It follows that $\eta \in \mathcal{E}_Q$ implies $\xi \in \mathcal{E}_{KQK^\top}$ where

$$\mathcal{E}_{KQK^\top}(t) \coloneqq \{(K(t)Q(t)K(t)^\top)^{\frac{1}{2}} y \mid \|y\|_2 \leq 1, y \in \mathbb{R}^{n_u}\}. \tag{5.18}$$

We refer $\mathcal{E}_{KQK^\top}$ to as an *input funnel*. We use the subscript $KQK^\top$ because, when $KQK^\top$ is PD, the set $\mathcal{E}_{KQK^\top}$ is equivalent to $\{\xi \mid \xi^\top (KQK^\top)^{-1}\xi \leq 1\}$. When $KQK^\top$ is only PSD (not PD), $\mathcal{E}_{KQK^\top}$ becomes a degenerated ellipsoid, which remains well-defined and compatible with (5.18). Using the state and input funnels, a *funnel* $\mathcal{F}(t)$ is defined as an Cartesian product of state and input funnel centered around the nominal state $\bar{x}(t)$ and input $\bar{u}(t)$, respectively, that can be written as

$$\mathcal{F}(t) \coloneqq (\{\bar{x}(t)\} \oplus \mathcal{E}_Q(t)) \times \left(\{\bar{u}(t)\} \oplus \mathcal{E}_{KQK^\top}(t)\right). \tag{5.19}$$

Lemma 5.2. *Suppose there exists a PD matrix-valued continuous function $Q(t) \in \mathbb{S}^{n_x}_{++}$ and a continuous matrix-valued function $K(t) \in \mathbb{R}^{n_u \times n_x}$ such that all trajectories of the system* (5.12) *satisfy*

$$\textit{If } V \geq w^\top w, \textit{ then } \dot{V}(t, \eta(t)) \leq 0, \tag{5.20}$$

for almost all $t \in [t_0, t_f]$. Then, with every almost everywhere continuous signal $w(\cdot)$ such that (5.2)*, $\mathcal{E}_Q(t)$ defined in* (5.16) *is an invariant time-varying ellipsoid for the system* (5.12) *on $[t_0, t_f]$, that is, if $\eta(\cdot)$ is a solution of* (5.12) *with $\eta(t_0) \in \mathcal{E}_Q(t_0)$, then $\eta(t) \in \mathcal{E}_Q(t)$ for all $t \in [t_0, t_f]$.*

Proof. We prove by contradiction. Suppose that the invariance condition does not hold. Then, there exists a solution $\eta(\cdot)$ of (5.12) such that $\eta(t_0) \in \mathcal{E}_Q(t_0)$ and $\eta(t_2) \notin \mathcal{E}_Q(t_2)$ for some $t_2 > t_0$. Since $V(t, \eta)$ is continuous in t by continuity of η and Q, there exists $t_1 \in (t_0, t_2)$ such that $V(t_1, \eta(t_1)) = 1$ and $V(t, \eta(t)) > 1$ for all $t \in (t_1, t_2]$ by the intermediate value theorem. It follows from (5.20) that $\dot{V}(t, \eta(t)) \leq 0$ for almost all $t \in [t_1, t_2]$ as (5.2). Observe that

$$\begin{aligned} V(t_2, \eta(t_2)) &= V(t_1, \eta(t_1)) + \int_{t_1}^{t_2} \dot{V}(t, \eta(t)) \mathrm{d}t, \\ &\leq V(t_1, \eta(t_1)) = 1. \end{aligned}$$

This implies $\eta(t_2) \in \mathcal{E}_Q(t_2)$ that contradicts our hypothesis that $\eta(t_2) \notin \mathcal{E}_Q(t_2)$. Thus, $\mathcal{E}_Q(t)$ is invariant. □

For a detailed discussion of similar Lyapunov conditions in the context of input-to-state stability, we refer the reader to [115]. The similar Lyapunov condition with the time-invariant Lyapunov function is also studied in [2, Lemma 1]. While Lemma 5.2 follows a similar proof, we extend the result to consider more general types of disturbance signals $w(\cdot)$ that are continuous almost everywhere. In the next lemma, we define the invariance of the funnel $\mathcal{F}$ and its relevance to the invariance of $\mathcal{E}_Q$.

Lemma 5.3. *If $\mathcal{E}_Q(t)$ is invariant for the uncertain LPV system* (5.12) *with the linear feedback control $\xi = K\eta$, then the funnel $\mathcal{F}(t)$ defined in* (5.19) *is invariant for the original system* (5.1) *in a sense that if $x(\cdot)$ is a solution of* (5.1) *with $x(t_0) \in \{\bar{x}(t_0)\} \oplus \mathcal{E}_Q(t_0)$ and the control law,*

$$u(t) = \bar{u}(t) + K(t)(x(t) - \bar{x}(t)), \tag{5.21}$$

then $(x(t), u(t)) \in \mathcal{F}(t)$ for all $t \in [t_0, t_f]$.

Proof. Consider a solution $x(\cdot)$ of (5.1) with $x(t_0) \in \{\bar{x}(t_0)\} \oplus \mathcal{E}_Q(t_0)$. This implies that $\eta(t_0) \in \mathcal{E}_Q(t_0)$, and then by the invariance of $\mathcal{E}_Q$, we have $\eta(t) = x(t) - \bar{x}(t) \in \mathcal{E}_Q(t)$ with $\xi(t) = u(t) - \bar{u}(t) = K(t)\eta(t)$. Thus, $(x(t), u(t)) \in \mathcal{F}(t)$ for all $t \in [t_0, t_f]$ with (5.21). □

5.1.4 Differential linear matrix inequality

This subsection describes the derivation of the DLMI that implies the invariance condition (5.20).

Lemma 5.4. *Consider the uncertain LPV system in* (5.12) *with the multiplier matrix $M(t)$ satisfying* (5.14). *Suppose there exist $Q(t) \in \mathbb{S}^{n_x}_{++}$, $Y(t) \in \mathbb{R}^{n_u \times n_x}$, $\lambda_\beta(t) \in \mathbb{R}_+$, $\lambda_\gamma(t) \in \mathbb{R}_+$,*

and $\lambda_w \in \mathbb{R}_+$ such that the following differential matrix inequality holds for some $t \in [t_0, t_f]$:

$$D_{LMI}(t, \dot{Q}, Q, Y, \lambda_\beta, \lambda_\gamma) \coloneqq$$
$$\begin{bmatrix} W - \dot{Q} & \star & \star & \star \\ N_2 E^\top & -N_2 & \star & \star \\ \tilde{F}^\top & 0 & -\lambda_w I & \star \\ CQ + DY & 0 & G & -N_1 \end{bmatrix} \preceq 0, \tag{5.22}$$

where $W = \tilde{A}Q + \tilde{B}Y + Q\tilde{A}^\top + Y^\top \tilde{B}^\top + \lambda_w Q$ and N_1, N_2 are defined in (5.14)*. Let $K(t) = Y(t)Q(t)^{-1}$, and then the condition* (5.20) *holds for the given t.*

Proof. Pre- and post-multiplying (5.22) by $\mathrm{diag}(Q^{-1}, N_2^{-1}, I, I)$ generates

$$\begin{bmatrix} Q^{-1}(W - \dot{Q})Q^{-1} & \star & \star & \star \\ E^\top Q^{-1} & -N_2^{-1} & \star & \star \\ \tilde{F}^\top Q^{-1} & 0 & -\lambda_w I & \star \\ C_{cl} & 0 & G & -N_1 \end{bmatrix} \preceq 0,$$

where $C_{cl} = C + DK$. By applying Schur complement, we can derive

$$\begin{bmatrix} Q^{-1}(W - \dot{Q})Q^{-1} & \star & \star \\ E^\top Q^{-1} & 0 & \star \\ \tilde{F}^\top Q^{-1} & 0 & -\lambda_w I \end{bmatrix} + (\star)^\top \begin{bmatrix} N_1^{-1} & 0 \\ 0 & N_2^{-1} \end{bmatrix} \begin{bmatrix} C_{cl} & 0 & G \\ 0 & I & 0 \end{bmatrix} \preceq 0.$$

By post- and pre-multiplying by $\mathrm{diag}(\eta, p, w)$ and its transpose, respectively, and applying the identity $\dot{Q}^{-1} = -Q^{-1}\dot{Q}Q^{-1}$, we obtain

$$\dot{V} + (\star)^\top \begin{bmatrix} N_1^{-1} & 0 \\ 0 & N_2^{-1} \end{bmatrix} \begin{bmatrix} r \\ p \end{bmatrix} + \lambda_w (V - w^\top w) \leq 0,$$

for all $\eta \in \mathbb{R}^{n_x}$, $p \in \mathbb{R}^{n_p}$, and $w \in \mathbb{R}^{n_w}$. By using (5.13), S-procedure [19, 132], and $\lambda_w > 0$, we can conclude that the above inequality implies (5.20). □

We omit the time arguments in Lemma 5.4 and its proof for the notational brevity. The matrices C, D, E, G, and the constant λ_w are time-invariant, while the terms $\tilde{A}$, $\tilde{B}$, $\tilde{F}$, Q, Y, K, W, N_1, and N_2 are time-varying. Notice that with the fixed positive constant λ_w, $D_{LMI}(\dot{Q}, Q, Y, \lambda_\beta, \lambda_\gamma)$ in (5.22) is linear in its arguments.

5.2 Funnel synthesis problem

5.2.1 Transformation of DLMI into a finite number of LMIs

To transform the DLMI (5.22) into a finite number of LMIs, we first parameterize $Q(t)$, $Y(t)$, $\lambda_\beta(t)$, and $\lambda_\gamma(t)$ by employing the same FOH interpolation used in (5.8), that is,

$$\tilde{\square}(t) \coloneqq \sigma_1^k(t)\square_k + \sigma_2^k(t)\square_{k+1}, \quad t \in [t_k, t_{k+1}],$$

where a placeholder $\square$ corresponds to Q, Y, λ_β, and λ_γ, and the time-varying parameters $\sigma_1^k(t)$ and $\sigma_2^k(t)$ are given in (5.8b). Here $Q_k \in \mathbb{S}_{++}^{n_x}, Y_k \in \mathbb{R}^{n_u \times n_x}$, $(\lambda_\beta)_k \in \mathbb{R}_+$, and $(\lambda_\gamma)_k \in \mathbb{R}_+$ for all $k \in \mathbb{Z}_{[0,N]}$ are our decision variables to be optimized. With this interpolation, the derivative of $\dot{Q}$ for each open subinterval (t_k, t_{k+1}), denoted by $\dot{Q}^k$, is constant and given by

$$\dot{Q}^k \coloneqq \dot{Q} = \frac{Q_{k+1} - Q_k}{t_{k+1} - t_k}, \tag{5.23}$$

where the superscript k indicates that $\dot{Q}$ is specific to each subinterval.

To derive a finite number of LMIs that ensure the invariance, we also need to represent the time-varying constants $\gamma(t)$ and $\beta(t)$ with a finite set of values. To this end, we define

$$(\gamma_i)_k \coloneqq \sup_{t \in [t_k, t_{k+1}]} \gamma_i(t), \quad \beta_k \coloneqq \sup_{t \in [t_k, t_{k+1}]} \beta(t), \tag{5.24}$$

for all k in $\mathbb{Z}_{[0,N-1]}$. These constants $(\gamma_i)_k$ and β_k serve as upper bounds for $\gamma_i(t)$ and $\beta(t)$, respectively, on each subinterval $[t_k, t_{k+1}]$. Hence, they become valid constants for (5.7) and (5.10), respectively on each subinterval $[t_k, t_{k+1}]$. This allows us to incorporate $(\gamma_i)_k$ and β_k as constants in the LMIs associated with each subinterval, facilitating the derivation of a finite set of LMIs for the entire horizon $[t_0, t_f]$

With the FOH interpolation (5.8) for the system matrices and our decision variables, the DLMI condition (5.22) for open subinterval (t_k, t_{k+1}) can be equivalently rewritten as

$$\sigma_1^k(t)\sigma_1^k(t)H_{k,k}^k + \sigma_2^k(t)\sigma_2^k(t)H_{k+1,k+1}^k + \sigma_1^k(t)\sigma_2^k(t)(H_{k,k+1}^k + H_{k+1,k}^k) \succeq 0, \tag{5.25}$$

where $H_{i,j}^k \in \mathbb{R}^{n_H \times n_H}$ with $n_H = n_x + n_p + n_w + n_r$ for $i, j \in \{k, k+1\}$ is given by

$$H_{i,j} \coloneqq -\begin{bmatrix} W_{i,j} - \dot{Q}^k & \star & \star & \star \\ N_{2,j}^k E^\top & -N_{2,j}^k & \star & \star \\ F_j^\top & 0 & -\lambda_w I & \star \\ L_j & 0 & G & -N_{1,j} \end{bmatrix}, \tag{5.26}$$

$$\begin{aligned} W_{i,j} &\coloneqq A_i Q_j + B_i Y_j + Q_j A_i^\top + Y_j^\top B_i^\top + \lambda_w Q_j, \\ L_j &\coloneqq C Q_j + D Y_j, N_{1,j} \coloneqq \operatorname{diag}((\lambda_\beta)_j I, (\lambda_\gamma)_j I), \\ N_{2,j}^k &\coloneqq \operatorname{diag}(\beta_j^2 (\lambda_\beta)_j I, (\lambda_\gamma)_j \Gamma_k^2), \end{aligned}$$

where $\Gamma_k = \operatorname{diag}((\gamma_1)_k, \ldots, (\gamma_{n_\phi})_k)$ and the superscript k is introduced for the piecewise constant terms $\dot{Q}^k$, Γ_k, and β_k. Then, observe that (5.25) can be expressed as

$$(\star)^\top \underbrace{\begin{bmatrix} H_{k,k}^k & \star \\ \frac{1}{2}\left(H_{k,k+1}^k + H_{k+1,k}^k\right) & H_{k+1,k+1}^k \end{bmatrix}}_{\coloneqq P} \underbrace{\begin{bmatrix} \sigma_1^k(t) I \\ \sigma_2^k(t) I \end{bmatrix}}_{\coloneqq \Sigma} \succeq 0. \tag{5.27}$$

The condition (5.27) can be achieved by satisfying $P \in \mathcal{C}$ where

$$\mathcal{C} \coloneqq \left\{ P \middle| \forall \sigma_1, \forall \sigma_2 \in \mathbb{R}_+, \Sigma = [\sigma_1 I, \sigma_2 I]^\top, \Sigma^\top P \Sigma \succeq 0 \right\}.$$

It is worth noting that if $n_H = 1$, the set $\mathcal{C}$ is equal to the set of 2×2 copositive matrices [20]. With this observation, we are ready to derive sufficient LMI conditions ensuring $P \in \mathcal{C}$ by adapting techniques from the literature on matrix copositivity [93, 10]. It is worth noting that we formulate these conditions specifically for our problem so that we can obtain a finite set of LMIs that guarantee the invariance of the funnel.

Lemma 5.5. *Suppose that the following holds:*

$$H_{k,k}^k \succeq 0, \quad H_{k+1,k+1}^k \succeq 0, \tag{5.28a}$$

$$H_{k,k+1}^k + H_{k+1,k}^k \succeq 0. \tag{5.28b}$$

Then, (5.27) *holds for all* t *in* (t_k, t_{k+1}).

Proof. Each block of P is PD because of the hypothesis in the Lemma 5.5. Since each $\sigma_i^k(t)$ for $i \in \{1, 2\}$ is nonnegative for all t in (t_k, t_{k+1}), we have (5.25). □

Similar results were discussed in [91, 97]. Here, we can further derive a less conservative LMI condition implying $P \in \mathcal{C}$ compared to (5.28).

Lemma 5.6. *Suppose that there exist $H^k_{i,j}$ and $X^k_{i,j}$ for all $i, j \in \{k, k+1\}$ such that*

$$\begin{bmatrix} H^k_{k,k} - X^k_{k,k} & \star \\ \frac{1}{2}\left(H^k_{k,k+1} + H^k_{k+1,k}\right) - X^k_{k+1,k} & H^k_{k+1,k+1} - X^k_{k+1,k+1} \end{bmatrix} \succeq 0, \tag{5.29a}$$

$$X^k_{i,j} = (X^k_{j,i})^\top \succeq 0, \quad \forall i, j \in \{k, k+1\}. \tag{5.29b}$$

Then, (5.27) *holds for all t in (t_k, t_{k+1}).*

Proof. Notice that the matrix P defined in (5.27) can be written as

$$P = P_1 + P_2,$$

where P_1 and P_2 are given by

$$P_1 = \begin{bmatrix} H^k_{k,k} - X^k_{k,k} & \star \\ \frac{1}{2}\left(H^k_{k,k+1} + H^k_{k+1,k}\right) - X^k_{k+1,k} & H^k_{k+1,k+1} - X^k_{k+1,k+1} \end{bmatrix},$$

$$P_2 = \begin{bmatrix} X^k_{k,k} & \star \\ X^k_{k+1,k} & X^k_{k+1,k+1} \end{bmatrix}.$$

We have $\Sigma^\top P_1 \Sigma \succeq 0$ because of the hypothesis in Lemma 5.6. Also, since every block $X^k_{i,j}$ for all $i, j \in \{k, k+1\}$ in P_2 is assumed to be PD, we have $\Sigma^\top P_2 \Sigma \succeq 0$. Hence, we can conclude $\Sigma^\top P \Sigma \succeq 0$. □

By solving either (5.28) or (5.29) for all k in $\mathbb{Z}_{[0,N-1]}$, we can guarantee the satisfaction of the DLMI (5.22) for all $t \in [t_0, t_f]$ except at each temporal node points $t = t_k$ for all k in $\mathbb{Z}_{[0,N]}$.

Remark 5.7. *The matrix-valued function $Q(t)$ with the FOH* (5.8) *is not differentiable at each temporal node points $t = t_k$ for all k in $\mathbb{Z}_{[0,N]}$. This does not compromise our objective of continuous-time invariance since it suffices for the Lyapunov condition* (5.20) *to hold for almost all t in $[t_0, t_f]$ according to Lemma 5.2.*

It is worth noting that (5.29) is less conservative than (5.28), as every solution of (5.28) is a special case of (5.29). This follows from the fact that (5.29) with the following constraints

$$\begin{aligned} H^k_{i,i} &= X^k_{i,i}, \quad \forall i \in \{k, k+1\}, \\ \frac{1}{2}\left(H^k_{k,k+1} + H^k_{k+1,k}\right) &= X^k_{k,k+1}, \end{aligned}$$

is equivalent to (5.28). However, solving (5.29) requires to introduce additional variables $X \in \mathbb{R}^{n_H \times n_H}$ that are larger in dimension than the variables Q and Y, making the solution of (5.29) more computationally expensive than that of (5.28).

5.2.2 Constraints

The proposed funnel synthesis algorithm aims to satisfy not only the invariance of the funnel but also the state and input constraints. We consider linear state and input constraints written as

$$\mathcal{P}_x = \{x \mid (a^h_i)^\top x \leq b^h_i, \quad i = 1, \ldots, m_x\}, \tag{5.30a}$$

$$\mathcal{P}_u = \{x \mid (a^g_j)^\top u \leq b^g_j, \quad j = 1, \ldots, m_u\}. \tag{5.30b}$$

We aim to make the state and input funnels at each node point centered around the nominal trajectory remain inside the feasible region. This can be stated using the set inclusions as $\{\bar{x}_k\} \oplus \mathcal{E}_Q(t_k) \subseteq \mathcal{P}_x$ and $\{\bar{u}_k\} \oplus \mathcal{E}_{KQK^\top}(t_k) \subseteq \mathcal{P}_u$ for all $k \in \mathbb{Z}_{[0,N]}$ where $\bar{x}_k = \bar{x}(t_k)$ and $\bar{u}_k = \bar{u}(t_k)$. These conditions can be equivalently written by the following LMIs [59]:

$$0 \preceq \begin{bmatrix} \left(b^h_i - (a^h_i)^\top \bar{x}_k\right)^2 & (a^h_i)^\top Q_k \\ Q_k a^h_i & Q_k \end{bmatrix}, i = \mathbb{Z}_{[1,m_x]}, \tag{5.31a}$$

$$0 \preceq \begin{bmatrix} \left(b^g_j - (a^g_j)^\top \bar{u}_k\right)^2 & (a^g_j)^\top Y_k \\ Y_k^\top a^g_j & Q_k \end{bmatrix}, j = \mathbb{Z}_{[1,m_u]}. \tag{5.31b}$$

5.2.3 Problem formulation

The proposed funnel synthesis algorithm solves the following SDP problem:

$$\underset{Q_k, Y_k, (\lambda_\beta)_k, (\lambda_\gamma)_k, X^k_{k,k}, X^k_{k,k+1}, X^k_{k+1,k+1}}{\text{minimize}} \quad J(Q_0, \ldots, Q_N) \tag{5.32a}$$

$$\text{subject to} \quad (5.28) \text{ or } (5.29), \forall k \in \mathbb{Z}_{[0,N-1]}, \tag{5.32b}$$

$$(5.31), \forall k \in \mathbb{Z}_{[0,N]}, \tag{5.32c}$$

$$0 \prec Q_k \preceq Q_{max}, \forall k \in \mathbb{Z}_{[0,N]}. \tag{5.32d}$$

The cost function J is assumed to be convex in Q_k. The matrix $Q_{max} \in \mathbb{S}^{n_x}_{++}$ is introduced to prohibit the funnel being arbitrarily large. By imposing $0 \prec Q_k$ for each $k \in \mathbb{Z}_{[0,N]}$, the FOH interpolation (5.8) ensures $0 \prec Q(t)$ for all $t \in [t_0, t_f]$.

Theorem 5.8. *Let $Q(t)$ and $Y(t)$ be obtained from the solution of* (5.32) *with the FOH interpolation* (5.8)*. Define the feedback gain $K(t) = Y(t)Q(t)^{-1}$. Then, the funnel $\mathcal{F}(t)$ defined in* (5.19) *is invariant for all $t \in [t_0, t_f]$.*

Proof. By Lemma 5.5 and 5.6, satisfying (5.32b) implies that the DLMI (5.22) holds for all $t \in [t_0, t_f]$ except at each temporal node point $t = t_k$ for all $k \in \mathbb{Z}_{[0,N]}$. It follows from Lemma 5.4 that the Lyapunov condition (5.20) holds almost everywhere. This implies the invariance of the state funnel $\mathcal{E}_Q$ followed from Lemma 5.2. Last, it follows from Lemma 5.3 that the funnel $\mathcal{F}(t)$ is invariant. □

Remark 5.9. *The feasibility condition in* (5.31) *enforces the constraints only at the temporal node points, which could result in violations between these node points. A possible remedy is to increase the number of node points. It is important to note that while the feasibility is only enforced at each node point, the invariance holds across the entire horizon $[t_0, t_f]$ as established in Theorem 5.8.*

Remark 5.10. *The feasibility of the problem* (5.32) *depends on many factors such as the magnitude of γ and β, the density of the node points, the configuration and size of obstacles, and the bounds on control inputs. Due to this complexity, it is generally difficult to guarantee feasibility a priori. However, a key advantage of the proposed approach is*

that it is formulated as a convex optimization problem, certifying the problem's infeasibility. In particular, if the problem is infeasible, the solver can provide a formal certificate of infeasibility.

5.2.4 Estimation of the constants via sampling

To solve the problem (5.32), it is necessary to obtain the local Lipschitz constant $(\gamma_i)_k$ and the bound β_k described in (5.10) for each $k \in \mathbb{Z}_{[0,N-1]}$. If the system is globally Lipschitz, one can directly use the global Lipschitz constant for all k [136]. One effective way for estimating local γ_k is to use sampling approaches [97], collecting samples of the triple (η_s, ξ_s, w_s) around the nominal trajectory, with s denoting each sample. Specifically, this involves sampling the state deviation η_s within the maximum state funnel Q_{max}, the input deviation ξ_s within the constraint (5.30b), and the disturbance within its bound (5.2) for each subinterval $[t_k, t_{k+1}]$. To obtain a less conservative constant, one might use the iterative procedure provided in [97] where the estimation of γ_k and funnel computation are alternately repeated until the convergence. This paper does not consider this iterative process for simplicity as our proposed method can be incorporated with any way to estimate the Lipschitz constant. On the other hand, the value of $\Delta_e(t)$ defined in (5.6) is known and can thus be evaluated at each t. Hence, for each subinterval $[t_k, t_{k+1}]$, we pick N_s sample temporal points t_s uniformly, and then $(\beta_e)_k$ can be determined through

$$(\beta_e)_k = \max_{s=1,\dots,N_s} \|\Delta_e(t_s)\|_2.$$

5.2.5 Algorithm summary

The overall procedure for synthesizing the funnel and controller is summarized below.

Algorithm 2 Funnel Synthesis

1: **Input:** Nominal trajectory, system dynamics, state/input constraints
2: Estimate γ and β as described in Section 3.4
3: **for** each λ_w in the search grid **do**
4: Solve the SDP problem in (5.32)
5: Store the cost and feasibility status
6: **end for**
7: Select λ_w with the lowest feasible cost
8: Return the corresponding funnel and controller

5.3 Numerical simulation

In this section, the proposed method is validated through two robotic motion planning and control applications: one for a unicycle model and the other for six-degree-of-freedom (6-DoF) powered descent guidance. For both examples, the Mosek solver is used to solve the SDP (5.32).

5.3.1 Unicycle model

We consider the unicycle model given in (5.4) with $c_1 = 0.03$ and $c_2 = 0.05$. We divide a 15 second time horizon into $N = 20$ subintervals, starting at $t_0 = 0$ and ending at $t_f = 15$. The cost function J in (5.32) is set as $-\text{trace}(Q_0) + \text{trace}(Q_N)$ to maximize the funnel entry and minimize the funnel exit. The nominal trajectory illustrated in Figure 5.2 starts at $(0, 0)$, passes through $(4, 8)$, and ends at $(8, 0)$. A state constraint for avoiding an obstacle is taken into account, resulting in a nonconvex state constraint as shown in Figure 5.2. This nonconvex constraint is linearized around the nominal trajectory to generate the polytopic constraints in the form of (5.30). The constraints for inputs are defined as follows: $0 \leq u_1 \leq 2$ (m/s) and $|u_2| \leq 1.5$ (rad/s). The parameter Q_{max} for the maximum funnel size is set to $\text{diag}(2^2, 2^2, (20\pi/180)^2)$. To estimate $(\gamma_i)k$ and β_k, we sample 100 triples of (η_s, ξ_s, w_s) and 20 samples of $\Delta(t_s)$ for each subinterval $[t_k, t_{k+1}]$.

To determine a value for λ_w, we solved the SDP problem (5.32) across a range of

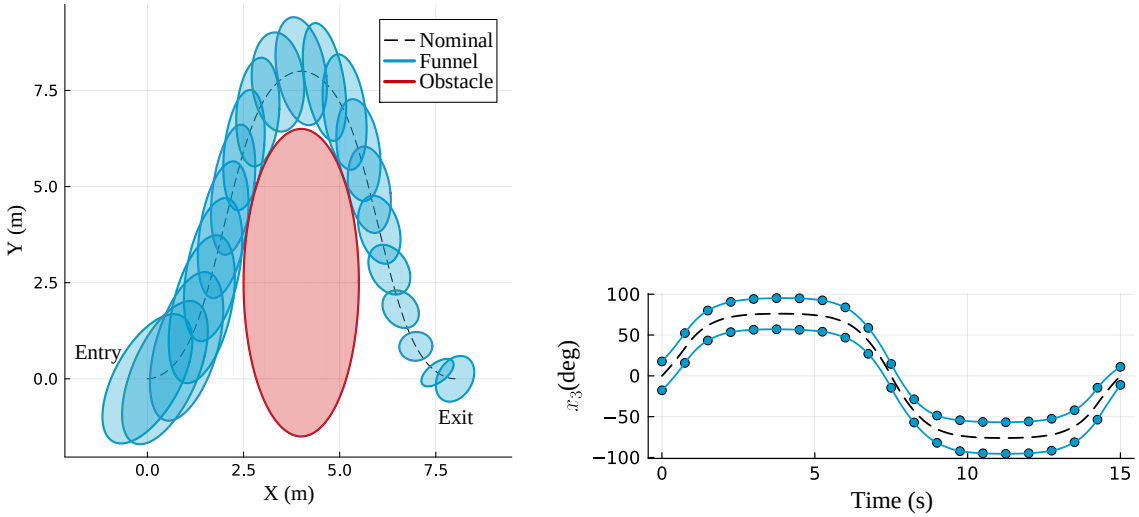

Figure 5.2: (Top) The synthesized state funnel projected on x (x_1) and y (x_2) position coordinates. (Bottom) Time history of the state funnel projected on yaw angle (x_3) coordinate.

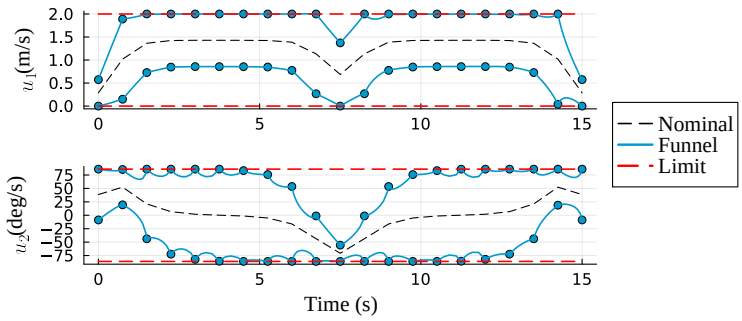

Figure 5.3: Time history of the synthesized input funnel projected on velocity command (u_1) and angular velocity command (u_2) coordinates, shown in the top and bottom figures, respectively.

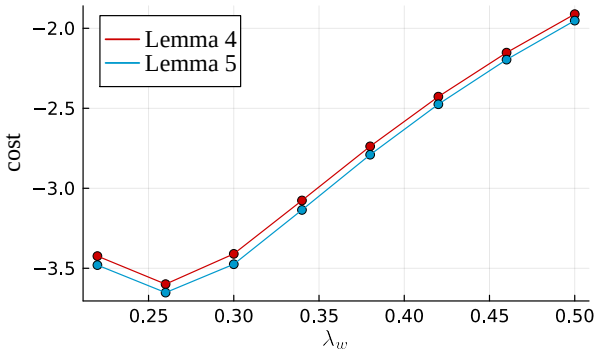

Figure 5.4: The cost results of (5.32) with different values of λ_w are presented for both cases: using (5.28) in Lemma 5.5 and (5.29) in Lemma 5.6.

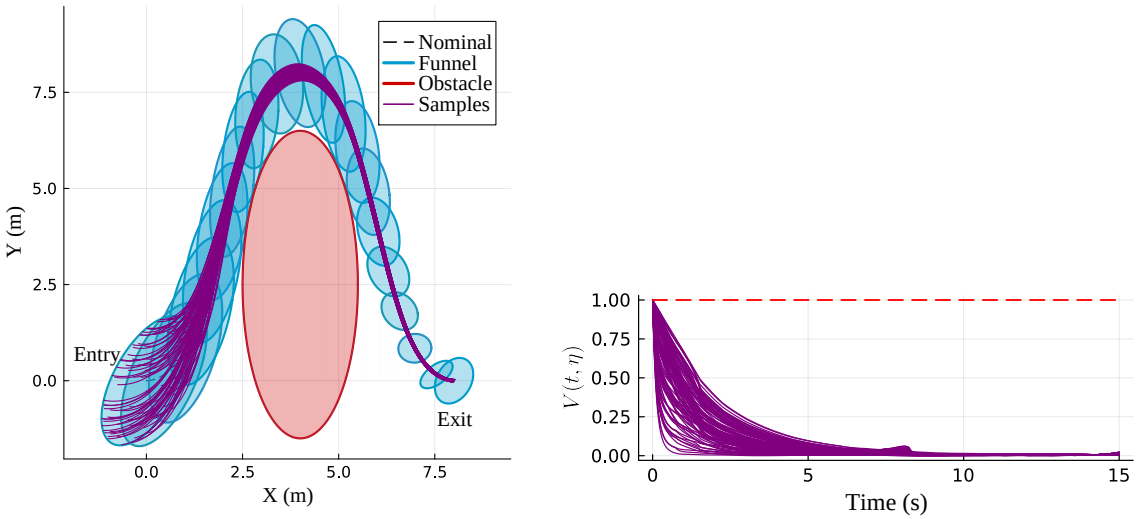

Figure 5.5: (Top) The results of trajectories propagated from randomly selected samples within the funnel entry. (Bottom) Time history of Lyapunov function V, as defined in (5.15), for each trajectory sample.

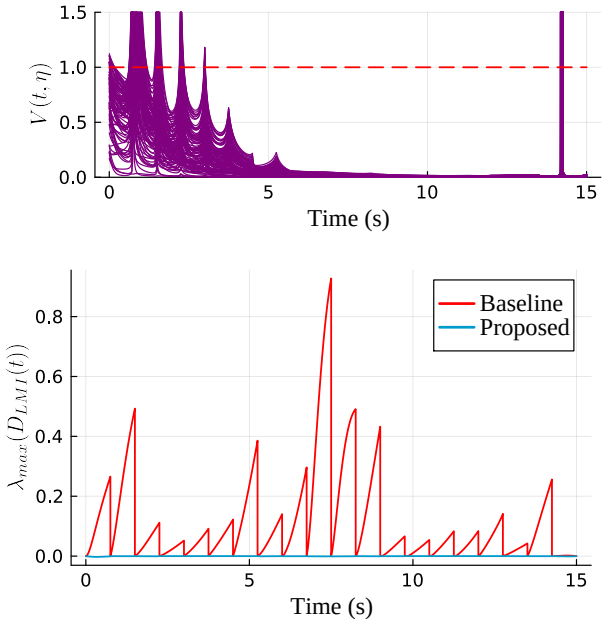

Figure 5.6: (Top) Evolution of the Lyapunov function for the funnel computed using the baseline approach. (Bottom) Time history of the maximum eigenvalue of D_{LMI}, as defined in (5.22), for the funnel obtained by both the baseline and proposed methods.

candidate λ_w values. The resulting cost values are summarized in Figure 5.4, which shows that the minimum cost is attained when $\lambda_w = 0.26$. The results of the state funnel and input funnel are provided in Figure 5.2 and Figure 5.3, respectively. It is clear that state and input constraints are satisfied at each node point. We conduct a comparison between two different conditions, (5.28) and (5.29), in terms of the resulting cost and computation time for synthesizing the funnel. The funnel computed using (5.29) has smaller cost, at -3.652, compared to -3.600 for the funnel using (5.28). The computational time for MOSEK to solve the SDP (5.32) with (5.28) is longer, at 11.63s, compared to 1.57s with (5.29), which is aligned with our expectation. Furthermore, Figure 5.4 shows that for every case, the cost associated with (5.29) is consistently lower than that obtained by (5.28).

To test the invariance condition of the funnel, we take a total of 500 samples from the surface of the funnel entry Q_0. Then, these samples are propagated through the control law (5.21). During the propagation, we randomly choose the disturbance w such that $\|w\|_2 = 1$. In the bottom figure of Figure 5.5, the value of the Lyapunov function V for each sample over the time horizon is illustrated. We can see that the values maintain below one, which means all trajectories remain in the funnel by the invariance property.

We compare the proposed method with a baseline approach that directly discretizes the DLMI (5.22) using the forward Euler method. This type of discretization has been commonly used in related funnel computation literature [74, Section 4.2]. To validate the baseline approach, we follow the same procedure as with the proposed method: 500 samples are taken from the surface of the funnel entry computed using the baseline, and each sample is propagated under the control law. The evolution of the Lyapunov function for each sample is shown in the top of Figure 5.6. Some trajectories exhibit Lyapunov function values greater than 1, indicating that the funnel computed by the baseline method violates the invariance condition. To explicitly assess satisfaction of (5.22) over time, we plot the time history of the maximum eigenvalue of D_{LMI}, which is defined in (5.22), using the decision variables from both the baseline and proposed methods in Figure 5.6. The results clearly show that the baseline approach violates the DLMI condition between the node points, whereas the proposed method satisfies it throughout the entire time horizon.

5.3.2 6-DoF Powered Descent Guidance

We consider a 6-DoF rigid-body rocket model in an East-North-Up inertia coordinate described by

$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{\Phi} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{m} C_{I/B}(\Phi) F + g \\ R(\Phi)\Omega \\ J^{-1}(T + r \times F - \Omega \times J\Omega) \end{bmatrix},$$

where the state vector $x = (p, v, \Phi, \Omega)$ consists of the position $p \in \mathbb{R}^3$, velocity $v \in \mathbb{R}^3$, Euler angles $\Phi \in \mathbb{R}^3$, and angular velocity $\Omega \in \mathbb{R}^3$, and the control input $u = (F, T)$ includes the thrust force $F \in \mathbb{R}^3$ and torque $T \in \mathbb{R}^3$. The constant $m \in \mathbb{R}$ is the vehicle mass, $J \in \mathbb{R}^{3\times3}$ is the inertia matrix, and $g \in \mathbb{R}^3$ is the gravitational constant. The vector $r \in \mathbb{R}^3$ is the position vector from the center of the mass to the location of the thrust. The matrix $C_{I/B}(\Phi)$ is the rotation matrix from the body frame to the inertia frame and the matrix $R(\Phi)$ represents the transformation between Euler angle rates and body-frame angular velocity. More details in the derivation of the model could be found in [117].

The nominal trajectory has a plane maneuver in the plane defined by $r_x = r_y$, with r_x and r_y denoting the position components along the East and North axes, respectively. We use nondimensionalized units, with U_T, U_L, and U_M denoting the units of time, length, and mass units, respectively. The mass m is set to 2 (U_M) and the gravity is set to $g = [0, 0, -1.625]^\top$ (U_M/U_T^2). We divide an entire horizon into $N = 5$ subintervals, starting at $t_0 = 0$ (U_T) and ending at $t_f = 7.457$ (U_T). The cost function is set to the same as in the unicycle example: $-\text{trace}(Q_0) + \text{trace}(Q_N)$. The constraint sets $\mathcal{X}$ and $\mathcal{U}$ are defined as

$$\begin{aligned} \mathcal{X} &= \{x \mid x_{lb} \le x \le x_{ub}\}, \\ \mathcal{U} &= \{u \mid u_{lb} \le u \le u_{ub}\}, \\ x_{ub} &= \text{diag}(8, 8, 8, 2, 2, 2, 30^\circ, 30^\circ, 30^\circ, 45^\circ, 45^\circ, 45^\circ), \\ x_{lb} &= -x_{lb}, \\ u_{ub} &= \text{diag}(1.5, 1.5, 5.0, 0.1, 0.1, 0.1), \\ u_{lb} &= -\text{diag}(1.5, 1.5, 0.0, 0.1, 0.1, 0.1). \end{aligned}$$

Since the bounded disturbance w is not considered in this example, it is not necessary to perform the line search for λ_w; thus λ_w is set to zero and the matrix $F(t)$ is set to the zero matrix for all time.

The computed funnels projected onto the two dimensional East-North plane and the three-dimensional East-North-Up space are illustrated in Figure 5.7 and in Figure 5.8, respectively. The funnel computed using (5.29) has a cost of -31.982, which is lower than the cost of the funnel computed using (5.28), -31.890. The MOSEK solver takes 28.10 seconds to solve the SDP with (5.29) and 17.41 seconds with (5.28).

Similar to the unicycle example, 500 samples are taken from the funnel entry, and each sample is propagated under the synthesized control law. Each sample's trajectory in the position coordinates is illustrated in Figure 5.8. From the results, we observe that although the nominal trajectory has an in-plane maneuver, the resulting funnel and sampled trajectories exhibit out-of-plane maneuver. The funnel invariance is validated by examining the evolution of the Lyapunov function, as shown in Figure 5.10, confirming that all trajectories remain inside the funnel. The resulting input funnel, projected onto each input dimension, is given in Figure 5.9, demonstrating that the input constraints are satisfied. The state funnel projected onto the Euler angle dimensions is given in Figure 5.10. While the funnel satisfies the state constraints at each node point, slight constraint violations occur between nodes.

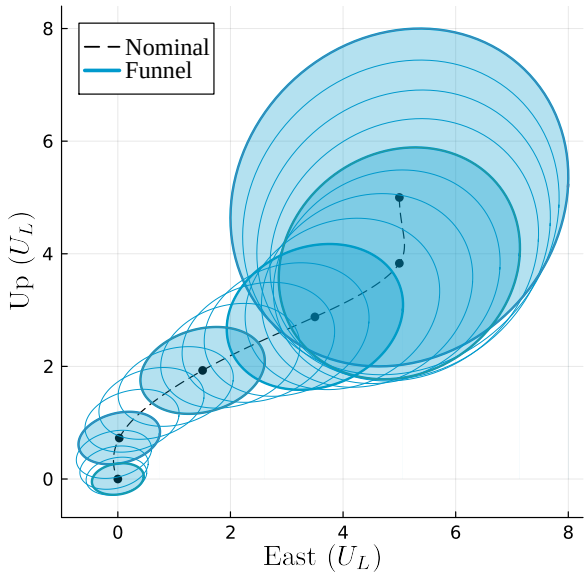

Figure 5.7: The synthesized state funnel projected onto the position coordinates. Filled ellipsoids represent the funnels at discrete node points, while the intermediate funnels between nodes are shown as unfilled ellipsoids.

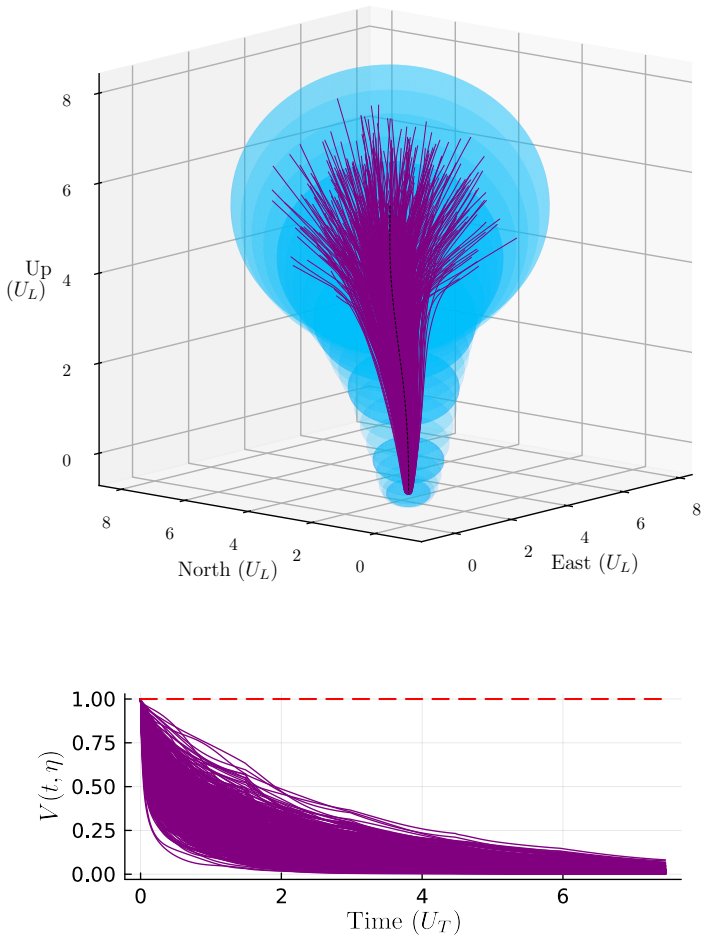

Figure 5.8: (Top) The computed funnel projected onto the three-dimensional position space. The funnel is shown in blue, and the propagated sample trajectories are shown in purple. (Bottom) Evolution of the Lyapunov function of the sample trajectories.

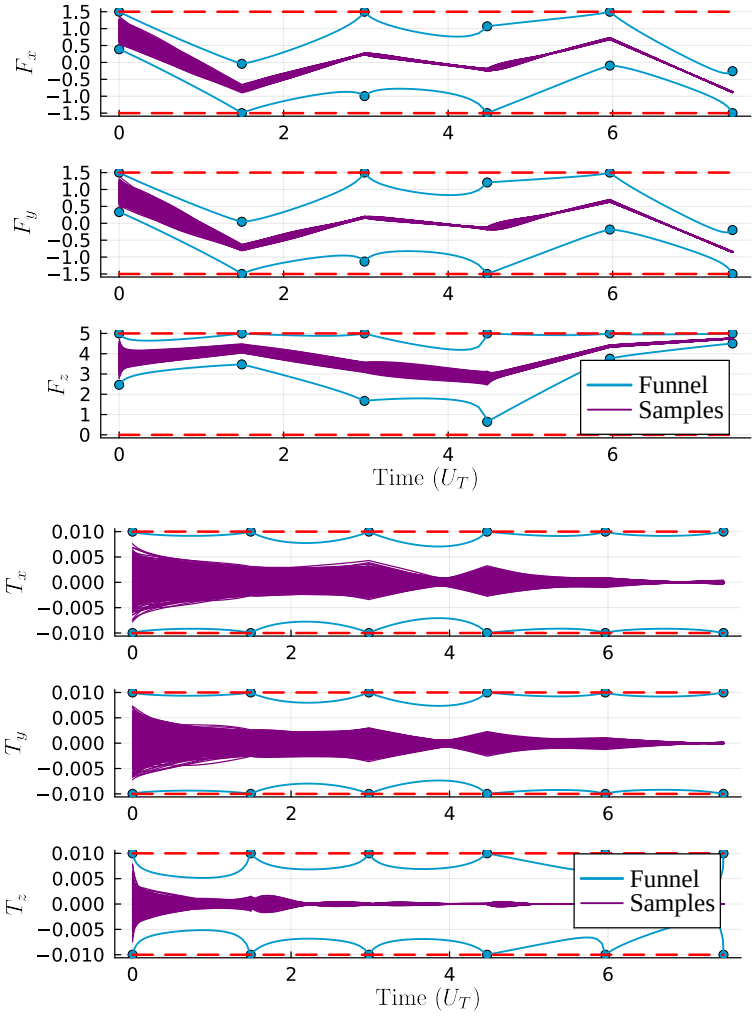

Figure 5.9: The synthesized input funnel onto each input dimension.

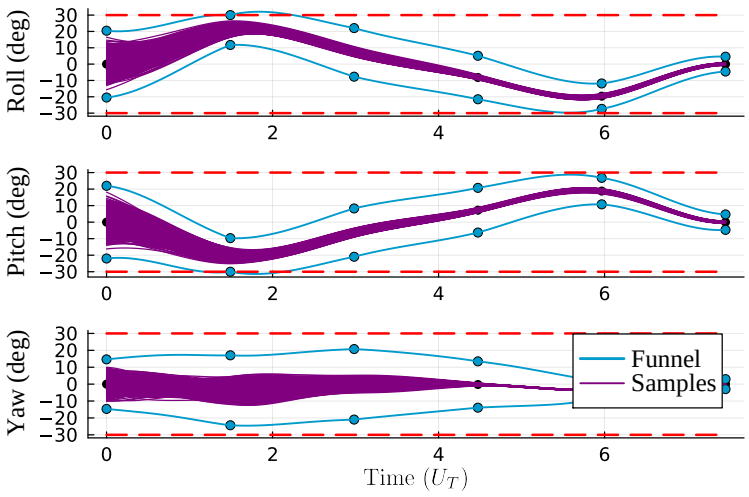

Figure 5.10: The synthesized state funnel projected onto each Euler angle dimension.

5.4 Conclusion

This paper presents a funnel synthesis algorithm for nonlinear systems under bounded disturbances. The proposed method can satisfy the invariance condition exactly over the finite time horizon. This is achieved by approximating the incremental system as an uncertain LPV system and deriving a finite number of LMIs that imply the DLMI resulting from the invariance condition. Simulation results show that the synthesized funnel satisfies both invariant and feasibility conditions.

Chapter 6

DISCRETE-TIME JOINT SYNTHESIS OF TRAJECTORY, CONTROLLED INVARIANT FUNNEL, AND FEEDBACK CONTROL

This chapter introduces a joint synthesis algorithm that computes both the nominal trajectory and the associated funnel within a unified optimization framework. While Chapters 3–5 focused on funnel synthesis with a fixed nominal trajectory, such separation often leads to conservative solutions. By allowing the trajectory and funnel to be co-designed, the proposed method reduces conservatism, improving feasibility and overall system performance. The chapter also presents how nominal dynamics and cost are integrated into the formulation and demonstrates the effectiveness of the approach through several simulation examples.

Chapter-specific notation. Let $\mathbb{R}$ be the field of real numbers, $\mathbb{R}^n$ be the n-dimensional Euclidean space, and $\mathbb{N}$ be the set of natural numbers. A finite set of consecutive non-negative integers is represented by $\mathcal{N}_q^r \coloneqq \{q, q+1, \ldots, r\}$. The symmetric matrix $Q = Q^\top (\succeq) \succ 0$ implies Q is positive-(semi-)definite matrix, and $(\mathbb{S}_+^n)\, \mathbb{S}_{++}^n$ denotes the set of all positive-(semi-)definite matrices whose size is $n \times n$. The symbol $\oplus$ denotes the Minkowski sum. The vector (x, y) represents the concatenation of two vectors x and u into a longer vector. The notation * represents the symmetric part of a matrix, i.e, $\begin{bmatrix} a & b^\top \\ b & c \end{bmatrix} = \begin{bmatrix} a & * \\ b & c \end{bmatrix}$, and $\{\bar{x}_k, \bar{u}_k, \bar{w}_k\}_{k=0}^K$ illustrates $\{\bar{x}_0, \bar{u}_0, \bar{w}_0, \ldots, \bar{x}_K, \bar{u}_K, \bar{w}_K\}$. The symmetric squared root of a symmetric matrix A is defined as $A^{\frac{1}{2}}$ by eigenvalue decomposition [20]. The operation diag$(\cdot)$ is a diagonal matrix formed from its vector argument.

6.1 Problem formulation

Consider a discrete-time uncertain nonlinear system of the following form:

$$x_{k+1} = f(t_k, x_k, u_k, w_k), \quad \forall\, k \in \mathcal{N}_0^{N-1}, \tag{6.1}$$

where $N \in \mathbb{N}$ is the length of the time horizon and $t_k \in \mathbb{R}$ is the time at the k. The function $f : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ is assumed to be a locally Lipschitz and at least once differentiable. The vector $x_k \in \mathbb{R}^{n_x}$ is the state, $u_k \in \mathbb{R}^{n_u}$ is the control input, and the signal $w_k \in \mathbb{R}^{n_w}$ is the exogenous disturbance or model mismatch that is assumed to be unknown but norm bounded: $\|w_k\|_2 \leq 1$ for all $k \in \mathcal{N}_0^{N-1}$.

Let $\{\bar{x}_k\}_{k=0}^{N}, \{\bar{u}_k, \bar{w}_k\}_{k=0}^{N-1}$ be a nominal trajectory that the CIF is centered around, and is feasible for the nonlinear dynamics (6.1). In this paper, the nominal trajectory is assumed to have zero disturbances, i.e., $\bar{w}_k = 0$ for all $k \in \mathcal{N}_0^{N-1}$. We define difference state $\eta_k \coloneqq x_k - \bar{x}_k$ and difference input $\xi_k \coloneqq u_k - \bar{u}_k$, and assume a linear feedback $\xi_k = K_k \eta_k$ for all $k \in \mathcal{N}_0^{N-1}$, which leads to a closed-loop system and a control law given by

$$\eta_{k+1} = f(t_k, x_k, u_k, w_k) - f(t_k, \bar{x}_k, \bar{u}_k, 0), \tag{6.2}$$

$$u_k = \bar{u}_k + K_k \eta_k, \quad \forall\, k \in \mathcal{N}_0^{N-1}, \tag{6.3}$$

where $K_k \in \mathbb{R}^{n_x \times n_u}$ is a feedback gain. In this paper, we consider a specific class of funnels that consists of ellipsoids of state and input. The ellipsoid for the difference state is represented as

$$\mathcal{E}_{Q_k} \coloneqq \{\eta \in \mathbb{R}^{n_x} \mid \eta^\top Q_k^{-1} \eta \leq 1\}, \quad \forall\, k \in \mathcal{N}_0^{N}, \tag{6.4}$$

where $Q_k \in \mathbb{S}_{++}^{n_x \times n_x}$ is a positive definite matrix. With the linear feedback gain K_k, it follows from Schur complement that $\eta_k \in \mathcal{E}_{Q_k}$ implies $\xi_k \in \mathcal{E}_{K_k Q_k K_k^\top}$ [97]. Now we are ready to formally define the quadratic CIF.

Definition 6.1. *A quadratic controlled positively invariant funnel, $\mathcal{F}_k$, associated with a closed loop system* (6.2) *is a time-varying set in state and control space that is parameterized by a time-varying positive definite matrix $Q_k \in \mathbb{S}_{++}^{n_x}$ and a time-varying matrix $K_k \in \mathbb{R}^{n_x \times n_u}$ such that $\mathcal{F}_k = \mathcal{E}_{Q_k} \times \mathcal{E}_{K_k Q_k K_k^\top}$, and the funnel $\mathcal{F}_k$ is invariant and feasible for all $k \in \mathcal{N}_0^N$.*

The invariance property of the CIF with the closed-loop system (6.2) and the control law (6.3) can be mathematically stated as follows:

$$(\eta_0, \xi_0) \in \mathcal{F}_0 \Rightarrow (\eta_k, \xi_k) \in \mathcal{F}_k, \; \forall\, k \in \mathcal{N}_1^N. \tag{6.5}$$

This condition implies that if a particular initial condition is inside the funnel, then a trajectory propagated with the closed-loop model (6.2) remains within the funnel as well. The feasibility property for the funnel $\mathcal{F}_k$ can be mathematically expressed as:

$$\{\bar{x}_k\} \oplus \mathcal{E}_{Q_k} \subseteq \mathcal{X}, \tag{6.6a}$$

$$\{\bar{u}_k\} \oplus \mathcal{E}_{K_k Q_k K_k^\top} \subseteq \mathcal{U}, \quad \forall\, k \in \mathcal{N}_0^{N-1}. \tag{6.6b}$$

The feasibility conditions require that every state and input in the funnel around the nominal trajectory should be feasible for the given state and input constraint sets $\mathcal{X}$ and $\mathcal{U}$, respectively.

Now we are ready to derive the problem formulation. The goal of the joint synthesis of trajectory and CIF is to solve a discrete-time nonconvex optimization problem of the following form:

$$\underset{\substack{\bar{x}_k, Q_k, \mu_k^Q \,\forall\, k \in \mathcal{N}_0^N, \\ \bar{u}_k, K_k, \mu_k^K, \forall\, k \in \mathcal{N}_0^{N-1}}}{\text{minimize}} \quad \sum_{k=0}^{N-1} J_t(\bar{x}_k, \bar{u}_k) + w_Q \sum_{k=0}^{N} \mu_k^Q + w_K \sum_{k=0}^{N-1} \mu_k^K \tag{6.7a}$$

$$\text{subject to} \quad \bar{x}_{k+1} = f(t_k, \bar{x}_k, \bar{u}_k, 0), \forall\, k \in \mathcal{N}_0^{N-1} \tag{6.7b}$$

$$Q_k \preceq \mu_k^Q I, \forall\, k \in \mathcal{N}_0^N \tag{6.7c}$$

$$K_k Q_k K_k^\top \preceq \mu_k^K I, \forall\, k \in \mathcal{N}_0^{N-1} \tag{6.7d}$$

$$\text{conditions } (6.5) - (6.6),$$

$$\bar{x}_0 \oplus \mathcal{E}_{Q_0} \supseteq \mathcal{X}_0, \tag{6.7e}$$

$$\bar{x}_N \oplus \mathcal{E}_{Q_N} \subseteq \mathcal{X}_N, \tag{6.7f}$$

where the summands in the objective function consist of the trajectory cost and the funnel cost, and $0 < w_Q \in \mathbb{R}$ and $0 < w_K \in \mathbb{R}$ are user-defined weights. The function J_t is a cost for the trajectory and is assumed to be convex in $\bar{x}_k$ and $\bar{u}_k$. The slack variables $\mu_k^Q \in \mathbb{R}$ and

$\mu_k^K \in \mathbb{R}$ are introduced to minimize the diameter of the ellipsoidal sets $\mathcal{E}_{Q_k}$ and $\mathcal{E}_{K_k Q_k K_k^\top}$ in the funnel by imposing the constraints (6.7c)-(6.7d). Minimizing the size of the funnel leads the effect of the propagated disturbances starting from the initial set to be minimized [74]. While minimizing the cost, the formulation guarantees the invariance property in (6.5) and ensures the feasibility of the ellipsoids encapsulating the nominal states and inputs in (6.6). For boundary conditions, the initial and final ellipsoids, $\mathcal{X}_0$ and $\mathcal{X}_N$, are given as

$$\mathcal{X}_0 = \left\{ x \,\middle|\, (x - x_i)^\top Q_i^{-1} (x - x_i) \leq 1 \right\}, \tag{6.8a}$$

$$\mathcal{X}_N = \left\{ x \,\middle|\, (x - x_f)^\top Q_f^{-1} (x - x_f) \leq 1 \right\}, \tag{6.8b}$$

where $x_i \in \mathbb{R}^{n_x}$ is a nominal initial state, $Q_i \in \mathbb{S}_{++}^{n_x}$ is a constant matrix defining the initial ellipsoidal set, $x_f \in \mathbb{R}^{n_x}$ is the nominal final state, and $Q_f \in \mathbb{S}_{++}^{n_x}$ is a constant matrix defining the final ellipsoidal set. The computed funnel at $k = 0$ should include the initial set $\mathcal{X}_0$ to generate the trajectory from any state in the initial set. Also, the ellipsoid corresponding to the state in the funnel at $k = N$ should be a subset of $\mathcal{X}_N$ so that the resulting trajectory is guaranteed to terminate in $\mathcal{X}_N$.

It is worth mentioning that the system dynamics (6.7b) for the nominal trajectory has no disturbances ($\bar{w}_k = 0$), but the invariance property is achieved with the closed-loop dynamics (6.2)-(6.3) in which the disturbances exist. Hence, any trajectory propagated for the uncertain nonlinear dynamics (6.1) with the control law (6.3) from any initial state in $\mathcal{X}_0$ remains within the feasible region under the presence of norm bounded uncertainties. The block diagram of the resulting control signal is illustrated in Figure 6.2.

6.2 Iterative Robust Trajectory Optimization

In this section, we discuss the details of the proposed method to solve the robust trajectory optimization problem given in (6.7a)-(6.7f). The method tackles the problem by iteratively updating the nominal trajectory $\{\bar{x}_k\}_{k=0}^N$,$\{\bar{u}_k\}_{k=0}^{N-1}$, the parameters of the set $\{Q_k\}_{k=0}^N$ and the feedback controller $\{K_k\}_{k=0}^N$ in the CIF. In each iteration, the method consists of 3 steps: the nominal trajectory update, the estimation of the locally Lipschitz constant, and the funnel update. In this section, we denote an initial guess or solution variables of the previous

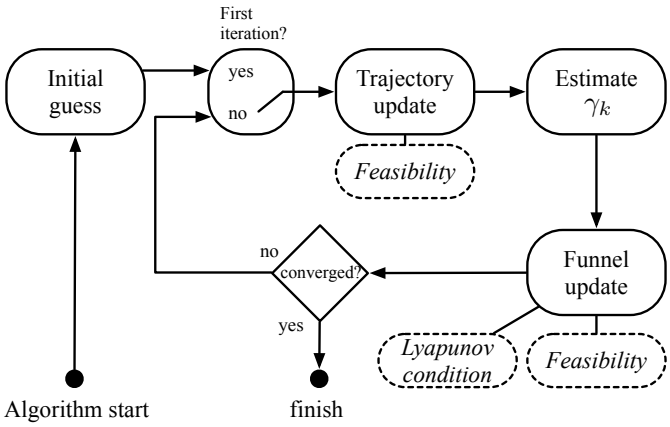

Figure 6.1: A block diagram of the proposed method. Starting from the initial guess, the method optimizes the trajectory while considering the feasibility of the funnel. The local Lipschitz constant γ_k of the nonlinearity around the obtained trajectory is then estimated. The next step is to optimize the funnel with the funnel constraints and the Lyapunov condition that ensures the invariance property. The entire process is repeated until both the trajectory and the funnel converge.

iteration (i.e., reference trajectory and funnel parameters) by $\{\hat{x}_k, \hat{Q}_k\}_{k=0}^{N}, \{\hat{u}_k, \hat{K}_k\}_{k=0}^{N-1}$. The block diagram of the proposed algorithms is given in Figure 6.1.

6.2.1 Nominal trajectory update

We require the nominal trajectory to satisfy the (possibly nonconvex) constraints (6.7b) and (6.6) while minimizing the trajectory cost J_t by approximating the original problem with a convex sub-problem. This is a typical process in many SCP methods to solve nonconvex trajectory optimization problems [77]. In contrast to the typical SCP methods, the feasibility problem in (6.6) involves the funnel parameters that are fixed as the reference funnel variables $\{\hat{Q}_k, \hat{K}_k\}_{k=0}^{N-1}$ in this trajectory update step.

In each sub-problem, the intermediate trajectory solution should satisfy the following

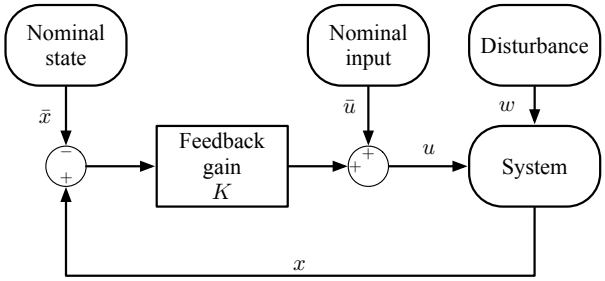

Figure 6.2: A block diagram of the control procedure.

affine system:

$$x_{k+1} = \bar{A}_k x_k + \bar{B}_k u_k + \bar{z}_k + v_k, \quad \forall\, k \in \mathcal{N}_0^{N-1} \tag{6.9}$$

where $\bar{A}_k, \bar{B}_k, \bar{z}_k$ define the linearized model of the nonlinear dynamics given in (6.1) evaluated around the reference trajectory $\{\hat{x}_k, \hat{u}_k\}_{k=0}^{N-1}$ with zero disturbance $\bar{w}_k = 0$. The term v_k is a virtual control variable that serves to prevent the sub-problem from being artificially infeasible [77] due to linearization of dynamics and constraints.

The feasible sets $\mathcal{X}$ and $\mathcal{U}$ are expressed as

$$\mathcal{X} = \{x \mid h_i(x) \leq 0, \quad i = 1, \ldots, m_x\},$$
$$\mathcal{U} = \{u \mid g_j(u) \leq 0, \quad j = 1, \ldots, m_u\},$$

where h_i and g_j are at least once differentiable functions. While we assume here that $\mathcal{X}$ and $\mathcal{U}$ are time-invariant for brevity, the proposed framework, however, can easily incorporate time-varying sets. The nonlinear constraints need to be linearized to ensure convexity of the sub-problem. Thus, we approximate the feasible set $\mathcal{X}$ and $\mathcal{U}$ as polytopes, which are obtained via linearization around $\{\hat{x}_k, \hat{u}_k\}_{k=0}^{N-1}$ as follows:

$$\mathcal{P}_k^x = \{x \mid (a_i^x)_k^\top x_k \leq (b_i^x)_k, \quad i = 1, \ldots, m_x\},$$
$$\mathcal{P}_k^u = \{u \mid (a_j^u)_k^\top x_k \leq (b_j^u)_k, \quad j = 1, \ldots, m_u\},$$

where (a_i^x, b_i^x) and (a_j^u, b_j^u) are first-order approximations of h_i and g_j, respectively. Notice that while $\mathcal{X}$ and $\mathcal{U}$ are assumed to be time-invariant, their polytopic approximations $\mathcal{P}_k^x$

and $\mathcal{P}_k^u$ would be time-varying due to the time variation in the reference trajectory $\hat{x}_k, \hat{y}_k$. Then, the feasibility conditions with the fixed funnel parameters $\{\hat{Q}_k, \hat{K}_k\}_{k=0}^{N-1}$ in (6.6) can be approximated as linear constraints as follows [97]:

$$\|(\hat{Q}_k^\top)^{\frac{1}{2}}(a_i^x)_k\|_2 + (a_i^x)_k^\top x_k \leq (b_i^x)_k, i = 1, \ldots, m_x, \tag{6.10a}$$

$$\|(\hat{K}_k\hat{Q}_k\hat{K}_k^\top)^{\frac{1}{2}}(a_j^u)_k\|_2 + (a_j^u)_k^\top u_k \leq (b_j^u)_k, j = 1, \ldots, m_u, \quad \forall\, k \in \mathcal{N}_0^{N-1}. \tag{6.10b}$$

The trajectory update step for the nominal trajectory has the following form of a second-order cone program (SOCP):

$$\underset{\substack{\bar{x}_k, \bar{u}_k, \bar{v}_k, \bar{x}_N, \\ \forall\, k \in \mathcal{N}_0^{N-1}}}{\text{minimize}} \quad \sum_{k=0}^{N-1} J_t(\bar{x}_k, \bar{u}_k) + J_{vc}(\bar{v}_k) + J_{tr}(\bar{x}_k, \bar{u}_k) \tag{6.11a}$$

$$\text{subject to conditions } (6.9) - (6.10),$$

$$\bar{x}_0 = x_i, \quad \bar{x}_N = x_f. \tag{6.11b}$$

In the cost function, there are two additional penalty terms for virtual control J_{vc} and trust region J_{tr}. The virtual control penalty enforces the virtual control variables v_k to remain small, and the trust region encourages the optimum to stay in the vicinity of the reference trajectory $\{\hat{x}_k, \hat{u}_k\}_{k=0}^{N-1}$ where the linearization error is small. They are formulated as follows:

$$J_{vc}(v_k) = w_v\|v_k\|_1, \tag{6.12a}$$

$$J_{tr}(x_k, u_k) = w_{tr}(\|x_k - \hat{x}_k\|_2^2 + \|u_k - \hat{u}_k\|_2^2), \tag{6.12b}$$

where $w_v \in \mathbb{R}$ and $w_{tr} \in \mathbb{R}$ are user-defined weight parameters for the virtual control and the trust region, respectively. As a result of the optimization problem (6.11), the solution becomes a new nominal trajectory $\{\bar{x}_k\}_{k=0}^{N}, \{\bar{u}_k\}_{k=0}^{N-1}$ that will be used for the funnel computation in the following section. This type of penalized-trust region-based optimization has been studied for trajectory optimization [98] and general nonlinear programming [25].

6.2.2 CIF update

In this section, we describe how to optimize the CIF around the nominal trajectory obtained from the previous section. The optimization problem derived in this section aims to

make the funnel invariant (6.5) and feasible for the constraints (6.6) and the boundary conditions (6.7f) for locally Lipschitz nonlinear systems. To this end, we construct a SDP whose solution provides the parameters of the invariant set and the feedback gains $\{Q_k\}_{k=0}^{N}, \{K_k\}_{k=0}^{N-1}$.

Nonlinear dynamics

Since the nonlinear dynamics in (6.1) is at least once differentiable, it can be re-written as

$$x_{k+1} = f(t_k, x_k, u_k, w_k), \tag{6.13a}$$

$$= A_k x_k + B_k u_k + F_k w_k + E p_k, \tag{6.13b}$$

$$p_k = \phi_k(q_k), \tag{6.13c}$$

$$q_k = C x_k + D u_k + G w_k. \tag{6.13d}$$

Notice that all nonlinearities are lumped into a vector $p_k \in \mathbb{R}^{n_p}$ represented by a function $\phi_k : \mathbb{R}^{n_q} \to \mathbb{R}^{n_p}$ with its argument $q_k \in \mathbb{R}^{n_q}$. The matrix $E \in \mathbb{R}^{n_x \times n_p}$ is introduced since not all states are affected by the nonlinearities. The matrices A_k, B_k and F_k can be arbitrary, but we specifically choose A_k, B_k, and F_k to be the first order approximation of the nonlinear dynamics f around the nominal trajectory as follows:

$$A_k \coloneqq \left.\frac{\partial f}{\partial x}\right|_{x=\bar{x}_k, u=\bar{u}_k, w=0}, \quad B_k \coloneqq \left.\frac{\partial f}{\partial u}\right|_{x=\bar{x}_k, u=\bar{u}_k, w=0},$$
$$F_k \coloneqq \left.\frac{\partial f}{\partial w}\right|_{x=\bar{x}_k, u=\bar{u}_k, w=0}, \quad \forall\, k \in \mathcal{N}_0^{N-1}.$$

With difference state η_k and input ξ_k, the difference dynamics can be derived as

$$x_{k+1} - \bar{x}_{k+1} = A_k \eta_k + B_k \xi_k + F_k w_k + E(p_k - \bar{p}_k) + f(t_k, \bar{x}_k, \bar{u}_k, 0) - \bar{x}_{k+1},$$

where $\bar{p}_k = \phi_k(\bar{q}_k)$ and $\bar{q}_k = C\bar{x}_k + D\bar{u}_k$. The term $f(t_k, \bar{x}_k, \bar{u}_k, 0) - \bar{x}_{k+1}$ on the right hand side exists because of the dynamical error in the intermediate nominal trajectory $\{\bar{x}_k\}_{k=0}^{N}, \{\bar{u}_k\}_{k=0}^{N-1}$. This error is gradually reduced as the iteration proceeds because the trajectory update (6.11) ensures that the nominal trajectory becomes dynamically feasible for the entire interval. Thus, we intentionally do not consider this error in the funnel update step since it is sufficient for the funnel to satisfy the invariance and feasibility properties

with the converged nominal trajectory that is dynamically feasible. The difference dynamics we consider for the funnel update is consequently written as

$$\begin{aligned}\eta_{k+1} &= A_k\eta_k + B_k\xi_k + F_k w_k + E\delta p_k,\\ \delta p_k &= \phi_k(q_k) - \phi_k(\bar{q}_k),\\ \delta q_k &= C\eta_k + D\xi_k + Gw_k,\end{aligned}$$

where $\delta p_k \coloneqq p_k - \bar{p}_k$ and $\delta q_k \coloneqq q_k - \bar{q}_k$. With the linear feedback controller $\xi_k = K_k\eta_k$, the inclusion $\eta_k \in \mathcal{E}_{Q_k}$ implies that q_k is in a compact set $\mathcal{Q}$ that is given as

$$\begin{aligned}\delta\mathcal{Q}_k &= \mathcal{E}_{C_k^{cl}Q_k(C_k^{cl})^\top} \oplus \{F_k w_k \mid \|w_k\|_2 \le 1\},\\ \mathcal{Q}_k &= \{\bar{q}_k\} \oplus \delta\mathcal{Q}_k, \quad \forall\, k \in \mathcal{N}_0^{N-1},\end{aligned}$$

where $C_k^{cl} \coloneqq C + DK_k$. The assumption that the function f is locally Lipschitz implies that the function ϕ_k is locally Lipschitz as well. Thus, for the compact (closed and bounded) set $\mathcal{Q}_k$, there exists a γ_k such that

$$\begin{aligned}\|\phi_k(q_k) - \phi_k(\bar{q}_k)\|_2 &\le \gamma_k\|q_k - \bar{q}_k\|_2,\\ \forall\, q_k \in \mathcal{Q}_k, &\forall\, k \in \mathcal{N}_0^{N-1}.\end{aligned}$$

Considering them together, the closed-loop system becomes

$$\eta_{k+1} = A_k^{cl}\eta_k + F_k w_k + E\delta p_k, \tag{6.14a}$$

$$\delta q_k = C_k^{cl}\eta_k + Gw_k, \tag{6.14b}$$

$$\|\delta p_k\|_2 \le \gamma_k\|\delta q_k\|_2, \tag{6.14c}$$

$$\|w_k\| \le 1, \tag{6.14d}$$

$$\delta q_k \in \delta\mathcal{Q}, \quad \forall\, k \in \mathcal{N}_0^{N-1}, \tag{6.14e}$$

where $A_k^{cl} \coloneqq A_k + B_k K_k$.

Invariance of a quadratic funnel

Consider a scalar-valued quadratic Lyapunov function V defined by

$$V(k,\eta) = \eta_k^\top Q_k^{-1}\eta_k. \tag{6.15}$$

For the closed-loop system model (6.14), we aim to design $\{Q_k\}_{k=0}^{N}, \{K_k\}_{k=0}^{N-1}$ that satisfies the following quadratic stability condition:

$$V(k+1, \eta_{k+1}) \leq \alpha V(k, \eta_k), \tag{6.16a}$$

$$\forall \, \|\delta p_k\|_2 \leq \gamma_k \|\delta q_k\|_2, \tag{6.16b}$$

$$\forall \, V(k, \eta_k) \geq \|w_k\|_2^2, \tag{6.16c}$$

$$\forall \, k \in \mathcal{N}_0^{N-1}$$

where $0 < \alpha < 1$. The above condition ensures the quadratic stability (6.16a) whenever the locally Lipschitz property of the nonlinearity ϕ_k expressed in (6.16b) holds. The condition (6.16c) exists to obtain the invariance property of the funnel under the presence of the bounded disturbance w_k. In the rest of this subsection, we construct a condition that implies the stability condition (6.16). In the following corollary, we also show that the derived LMI condition ensures the invariance property of the funnel.

Theorem 6.2. *Suppose that there exists* $Q_k \in \mathbb{S}_{++}^{n_x}$, $Y_k \in \mathbb{R}^{n_u \times n_x}$, $\nu_k^p > 0$, $\lambda_k^w > 0$, *and* $0 < \alpha < 1$ *such that* $\lambda_k^w < \alpha$ *and the following matrix inequality holds for all* $k \in \mathcal{N}_0^{N-1}$*:*

$$\begin{bmatrix} \alpha Q_k - \lambda_k^w Q_k & * & * & * & * \\ 0 & \nu_k^p I & * & * & * \\ 0 & 0 & \lambda_k^w I & * & * \\ A_k Q_k + B_k Y_k & \nu_k^p E_k & F_k & Q_{k+1} & * \\ C_k Q_k + D_k Y_k & 0 & G_k & 0 & \nu_k^p \frac{1}{\gamma_k^2} I \end{bmatrix} \succeq 0. \tag{6.17}$$

Then the Lyapunov condition (6.16) *holds for the closed loop system* (6.14) *with* $K_k = Y_k Q_k^{-1}$.

Proof. With the closed-loop system (6.14), the condition (6.16) holds if there exists a $\lambda_k^p > 0$, $\lambda_k^w > 0$, and $0 < \alpha < 1$ such that the following inequality holds by $\mathcal{S}$-procedure [132] for all $\eta_k \in \mathbb{R}^{n_x}, w_k \in \mathbb{R}^{n_w}, \delta p \in \mathbb{R}^{n_p}$:

$$V(k+1, \eta_{k+1}) - \alpha V(k, \eta_k) + \lambda_k^w (V(k, \eta_k) - \|w_k\|_2^2) + \lambda_k^p (\gamma_k^2 \|\delta q_k\|_2^2 - \|\delta p_k\|_2^2) \leq 0. \tag{6.18}$$

This is equivalent to

$$\begin{aligned}
&\begin{bmatrix} A_k^{cl} & E_k & F_k \end{bmatrix}^\top Q_{k+1}^{-1} \begin{bmatrix} A_k^{cl} & E_k & F_k \end{bmatrix} \\
&+\lambda_k^p \begin{bmatrix} C_k^{cl} & 0 & G_k \\ 0 & I & 0 \end{bmatrix}^\top \begin{bmatrix} \gamma_k^2 I & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} C_k^{cl} & 0 & G_k \\ 0 & I & 0 \end{bmatrix} \\
&- \begin{bmatrix} \alpha Q_k^{-1} & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \end{bmatrix} + \lambda_k^w \begin{bmatrix} Q_k^{-1} & * & * \\ 0 & 0 & * \\ 0 & 0 & -I \end{bmatrix} \preceq 0.
\end{aligned}$$

With the appropriate re-arrangement and applying Schur complement, we obtain

$$\begin{bmatrix}
H_k^1 & * & * & * & * \\
0 & \lambda_k^p I & * & * & * \\
0 & 0 & \lambda_k^w I & * & * \\
Q_{k+1}^{-1} A_k^{cl} & Q_{k+1}^{-1} E_k & Q_{k+1}^{-1} F_k & Q_{k+1}^{-1} & * \\
C_k^{cl} & 0 & G_k & 0 & H_k^2
\end{bmatrix} \succeq 0$$

where H_k^1 and H_k^2 are given by

$$\begin{aligned}
H_k^1 &= \alpha Q_k^{-1} - \lambda_k^w Q_k^{-1}, \\
H_k^2 &= (\lambda_k^p)^{-1} \frac{1}{\gamma_k^2} I.
\end{aligned}$$

Multiplying both sides by $\mathrm{diag}\{Q_k, \lambda_p^{-1} I, Q_{k+1}, I\}$ yields

$$\begin{bmatrix}
\alpha Q_k - \lambda_k^w Q_k & * & * & * & * \\
0 & \nu_k^p I & * & * & * \\
0 & 0 & \lambda_k^w I & * & * \\
A_k^{cl} Q_k & \nu_k^p E_k & F_k & Q_{k+1} & * \\
C_k^{cl} Q_k & 0 & G_k & 0 & \nu_k^p \frac{1}{\gamma_k^2} I
\end{bmatrix} \succeq 0,$$

where $\nu_k^p = (\lambda_k^p)^{-1}$. Finally, expanding A_k^{cl} and C_k^{cl} completes the proof. □

Corollary 6.3. *The condition* (6.17) *in Theorem 1 implies the following invariance*

condition for all $k \in \mathcal{N}_0^{N-1}$*:*

$$V(k+1, \eta_{k+1}) \leq 1, \tag{6.19a}$$

$$\forall V(k, \eta_k) \leq 1, \tag{6.19b}$$

$$\forall \|\delta p_k\|_2 \leq \gamma_k \|\delta q_k\|_2, \tag{6.19c}$$

$$\forall \|w_k\|_2 \leq 1. \tag{6.19d}$$

Proof. Observe that (6.18) can be equivalently written as

$$V(k+1, \eta_{k+1}) - \alpha + (\alpha - \lambda_k^w)(1 - V(k, \eta_k)) + \lambda_k^w(1 - \|w_k\|_2^2) + \lambda_k^p(\gamma_k^2 \|\delta q_k\|_2^2 - \|\delta p_k\|_2^2) \leq 0.$$

This implies $V(k+1, \eta_{k+1}) \leq \alpha$ with (6.19b)-(6.19d) since $0 < \lambda_k^w < \alpha$ and $\lambda_k^p > 0$. This is sufficient for the invariance condition (6.19) since $\alpha < 1$. □

Notice that the matrix inequality (6.17) is a LMI once α and λ_k^w are fixed.

Computing the funnel via SDP

The goal of computing the CIF is to bound the effects of disturbances going forward in time by minimizing the size of the funnel while satisfying the invariance and the feasibility of the

boundary conditions. To this end, the funnel computation is posed as the following SDP:

$$\underset{\substack{Q_k,\mu_k^Q,\forall\, k\in\mathcal{N}_0^N,\\ Y_k,\mu_k^K,\nu_k^p,\forall\, k\in\mathcal{N}_0^{N-1}}}{\text{minimize}} \quad w_Q\sum_{k=0}^{N}\mu_k^Q + w_K\sum_{k=0}^{N-1}\mu_k^K + \sum_{k=0}^{N-1}J_{trf}(Q_k,Y_k) \tag{6.20a}$$

$$\text{subject to} \quad Q_k \preceq \mu_k^Q I, \forall\, k\in\mathcal{N}_0^N, \tag{6.20b}$$

$$\begin{bmatrix} \mu_k^K I & Y_k \\ Y_k^\top & Q_k \end{bmatrix} \succeq 0, \forall\, k\in\mathcal{N}_0^{N-1}, \tag{6.20c}$$

$$\text{condition (6.17)}, \tag{6.20d}$$

$$\begin{bmatrix} ((b_i^x)_k - (a_i^x)_k^\top \bar{x}_k)^2 & (a_i^x)_k^\top Q_k^\top \\ Q_k(a_i^x)_k & Q_k \end{bmatrix} \succeq 0, i=1,\dots,m_x, \tag{6.20e}$$

$$\begin{bmatrix} ((b_j^u)_k - (a_j^u)_k^\top \bar{u}_k)^2 & (a_j^u)_k^\top Y_k^\top \\ Y_k(a_j^u)_k & Q_k \end{bmatrix} \succeq 0, j=1,\dots,m_u, \tag{6.20f}$$

$$Q_0 \succeq Q_i, \quad Q_N \preceq Q_f, \tag{6.20g}$$

where (6.20c) is equivalent to (6.7d) which can be derived by Schur complement with $Y_k = K_kQ_k$. The LMI constraints in (6.20e)-(6.20f) are the funnel feasibility conditions that are equivalent to (6.10) that can be derived by Schur complement. The cost J_{trf} is given as

$$J_{trf} = w_{trf}\sum_{k=0}^{N-1}\left(\|Q_k-\hat{Q}_k\|_F^2 + \|Y_k-\hat{Y}_k\|_F^2\right),$$

where $w_{trf}\in\mathbb{R}$ is a user-defined parameter, $\|\cdot\|_F$ is the Frobenius norm, and $\hat{Y}_k = \hat{K}_k\hat{Q}_k$ for all $k\in\mathcal{N}_0^{N-1}$. This cost, similar to the trust region penalty J_{tr}, penalizes the difference between the current solution $\{Q_k,Y_k\}_{k=0}^{N-1}$ and the previous solution $\{\hat{Q}_k,\hat{Y}_k\}_{k=0}^{N-1}$ which is beneficial for the better convergence performance.

The choice of parameters in the proposed method affects the performance of the control law in (6.3). The weights w_Q and w_K in (6.7a) balances the size of the state funnel $\mathcal{E}_{Q_k}$ and input funnel $\mathcal{E}_{K_kQ_kK_k^\top}$. For example, a relatively larger w_Q compared to w_K drives the algorithm to put more effort on minimizing the size of the state funnel over the input funnel, and vice versa. The choices of the decay rate α and the slack variable λ_k^w resulted from $\mathcal{S}$-procedure in (6.18) also affects the control performance. As the decay rate decreases,

the controller places greater emphasis on faster convergence to the nominal trajectory due to the condition outlined in (6.16a). Likewise, the larger λ_k^w places more emphasis on the convergence to the nominal trajectory. This is attributed to the term $V(k+1, \eta_{k+1}) - \alpha V(k, \eta_k)$ in (6.19) becoming smaller (more negative) as $\lambda_k^w(V(k, \eta_k) - \|w_k\|_2^2)$ increases.

Local Lipschitz constant estimation via sampling

To compute the LMI (6.17), the Lipschitz constant γ_k in (6.16) should be available. We estimate the Lipschitz constant by employing a sampling method. It is worth mentioning that the sampling method for the estimation of the Lipschitz constant γ_k brings about an algebraic loop: to estimate the Lipschitz constant γ_k, the funnel variables Q_k and K_k should be available, whereas the computation Q_k and K_k in (6.17) requires the constant γ_k. However, a well-behaved iterative scheme with the sampling method for γ_k can make the funnel computation converge [97].

By sampling a set of N_s pairs of state and disturbance $\{\eta_k^s, w_k^s\}_{s=1}^{N_s}$ from the ellipsoid $\mathcal{E}_Q$ and the set $\{w \in \mathbb{R}^{n_w} \mid \|w\|_2 \leq 1\}$, respectively, we compute

$$\delta_k^s = \frac{\|p_k^s - \bar{p}_k\|}{\|q_k^s - \bar{q}_k\|}, \quad s = 1, \ldots, N_s, \tag{6.21}$$

where p_k^s and q_k^s are computed by (6.13). Depending on the discretization method, only Ep_k might be available instead of p_k. So, it might not be possible to compute (6.21). In that case, we instead solve the following optimization to obtain the value δ_k^s:

$$\delta_k^s = \underset{\Delta}{\text{minimize}} \quad \|\Delta\|_2 \tag{6.22a}$$

$$\text{subject to } \eta_{k+1}^s - A_k^{cl}\eta_k^s - F_k w_k^s + \bar{x}_{k+1} - f(\bar{x}_k, \bar{u}_k, 0) = E\Delta(C_k^{cl}\eta_k^s + G_k w_k^s), \tag{6.22b}$$

where $\Delta \in \mathbb{R}^{n_p \times n_q}$. After obtaining δ_k^s by (6.21) or (6.22), the following maximization operation is performed to estimate the local Lipschitz constant:

$$\gamma_k = \underset{s=1,\ldots,N_s}{\text{maximize}} \ \delta_k^s, \quad \forall\, k \in \mathcal{N}_0^{N-1}. \tag{6.23}$$

It is worth noting that the disadvantage of the illustrated sampling-based method is that the computed γ_k might be lower than the true level of nonlinearity. To handle this issue,

one may able to use a probabilistic approach for overestimating the local Lipschitz constant from samples provided in [28].

Another way to estimate the local Lipschitz constant is to use the optimization-based approach provided in [99, Section 6.5.1]. To illustrate, we consider

$$\Gamma_k^* = \underset{\eta_k, w_k}{\text{maximize}} \quad \frac{1}{2}\delta_k^*(\eta_k, w_k)^2 \tag{6.24a}$$

$$\text{subject to } \eta_k Q_k^{-1} \eta_k \leq 1, \tag{6.24b}$$

$$\|w_k\|_2 \leq 1, \tag{6.24c}$$

where

$$\delta_k^*(\eta_k, w_k) = \underset{\Delta}{\text{minimize}} \quad \|\Delta\|_2 \tag{6.25a}$$

$$\text{subject to } \eta_{k+1} - A_k^{cl}\eta_k - F_k w_k + \bar{x}_{k+1} - f(\bar{x}_k, \bar{u}_k, 0) = E\Delta(C_k^{cl}\eta_k + G_k w_k). \tag{6.25b}$$

The inner optimization (6.25) aims to find the smallest matrix Δ in terms of the matrix 2-norm for the given η_k, w_k, and the outer optimization (6.24) finds the values of η_k, w_k that maximize δ_k^*. After solving these optimization problems for each k, the local Lipschitz constant can be obtained by computing

$$\gamma_k = \sqrt{2\Gamma_k^*}.$$

To make the outer optimization computationally tractable, one could potentially utilize an analytic upper bound for the problem's optimal value. The constraint (6.25b) can be rewritten as

$$y(\eta_k, w_k) = E\Delta(C_k^{cl}\eta_k + G_k w_k), \tag{6.26a}$$

$$= \underbrace{\begin{bmatrix} E(e_1^\top (C_k^{cl}\eta_k + G_k w_k)) & \cdots & E(e_{n_q}^\top (C_k^{cl}\eta_k + G_k w_k)) \end{bmatrix}}_{:= H(\eta_k, w_k)} \vec{\Delta}_k, \tag{6.26b}$$

where $y(\eta_k, w_k) = \eta_{k+1} - A_k^{cl}\eta_k - F_k w_k + \bar{x}_{k+1} - f(\bar{x}_k, \bar{u}_k, 0)$ and $\vec{\Delta}_k \in \mathbb{R}^{n_p n_q}$ is a concatenated vector that stacks the columns of Δ_k. Then, consider the following optimization:

$$\delta_k^*(\eta_k, w_k) = \underset{\vec{\Delta}_k}{\text{minimize}} \ \|\vec{\Delta}_k\|_2 \quad \text{subject to} \quad y(\eta_k, w_k) = H(\eta_k, w_k)\vec{\Delta}_k. \tag{6.27}$$

Since (6.27) is a minimum-norm least squares problem, the solution is $\vec{\Delta}_k = H^\dagger(\eta_k, w_k)y(\eta_k, w_k)$ [20] where $H^\dagger$ represents the peusoinverse of matrix H. The optimal value δ_k^* of (6.27) is the upper bound of the inner optimization (6.25) as $\|\Delta\|_2 \leq \|\Delta\|_F = \|\vec{\Delta}_k\|_2$ where $\|\cdot\|_F$ is the Frobenius norm. Then, the outer optimization (6.24) can be transformed into

$$\Gamma_k^* = \underset{\eta_k, w_k}{\text{maximize}} \quad \frac{1}{2}y(\eta_k, w_k)^\top H^\dagger(\eta_k, w_k)^\top H^\dagger(\eta_k, w_k)y(\eta_k, w_k) \tag{6.28a}$$

$$\text{subject to } \eta_k Q_k^{-1}\eta_k \leq 1, \tag{6.28b}$$

$$\|w_k\|_2 \leq 1, \tag{6.28c}$$

More details in the derivation and the computation results can be found in [99, Section 6.5.1].

6.2.3 Algorithm details and summary

To start the algorithm, we need to generate an initial guess hat is used as a reference trajectory for the first iteration. It is worth noting that the initial guess does not need to be feasible to constraints for the proposed method. The first way is to employ a straight-linear interpolation [77] for the initial nominal trajectory $\{x_k\}_{k=0}^N, \{u_k\}_{k=0}^{N-1}$. Then the feedback gain $\{K_k\}_{k=0}^{N-1}$ can be obtained by solving a discrete-time linear quadratic regulator problem with a linearized model of (6.1) evaluated around the nominal trajectory. The initial guess for the ellipsoid variable $\{Q_k\}_{k=0}^N$ can then be set to a diagonal matrix having user-defined diameters. The second way to generate the initial guess is to use the separate synthesis; the nominal trajectory is generated by the SCP algorithm without considering the funnel. This provides the dynamically-feasible trajectory that can be used as the initial guess of the nominal trajectory $\{x_k\}_{k=0}^N, \{u_k\}_{k=0}^{N-1}$ for the proposed method. Then, the feedback gain $\{K_k\}_{k=0}^{N-1}$ and the ellipsoid variable $\{Q_k\}_{k=0}^N$ can be obtained via solving (6.20) with $w_{trf} = 0$ while ignoring the funnel feasibility. The second way is more systematical since it exploits the result of the separate synthesis and hence gives a better initial guess compared to the solution computed by the straight-line interpolation in the first way.

To set the stopping criteria, we define $\Delta_{vc}, \Delta_{dyn}, \Delta_T$ and Δ_F as

$$\Delta_{vc} = \sum_{k=0}^{N-1} \|v_k\|_1, \ \Delta_{dyn} = \sum_{k=0}^{N-1} \|f(t_k, \bar{x}_k, \bar{u}_k, 0) - \bar{x}_{k+1}\|_2,$$
$$\Delta_T = \|x_N - \hat{x}_N\|_2^2 + \sum_{k=0}^{N-1} \|x_k - \hat{x}_k\|_2^2 + \|u_k - \hat{u}_k\|_2^2,$$
$$\Delta_F = \|Q_N - \hat{Q}_N\|_F^2 + \sum_{k=0}^{N-1} \|Q_k - \hat{Q}_k\|_F^2 + \|Y_k - \hat{Y}_k\|_F^2.$$

Then the stopping criteria is given as the following logical statement:

$$(\Delta_{vc} < \Delta_{vc}^{tol}) \wedge (\Delta_{dyn} < \Delta_{dyn}^{tol}) \wedge (\Delta_T < \Delta_T^{tol}) \wedge (\Delta_F < \Delta_F^{tol}), \tag{6.29}$$

where $\Delta_{vc}^{tol}, \Delta_{dyn}^{tol}, \Delta_T^{tol}$ and Δ_F^{tol} are user-defined tolerance parameters. The proposed algorithm is summarized in Algorithm 3.

Algorithm 3 Joint synthesis

Input: $(\hat{x}_k, \hat{u}_k \hat{Q}_k, \hat{K}_k)$

for $i = 1 \dots N_{max}$ **do**

optimize $\bar{x}_k, \bar{u}_k$ by (6.11)

estimate γ_k via (6.23) or (6.28)

optimize Q_k, K_k by (6.20)

if (6.29) is True **then**

break

end if

update $(\hat{x}_k, \hat{u}_k \hat{Q}_k, \hat{K}_k) \leftarrow (\bar{x}_k, \bar{u}_k, Q_k, K_k)$

end for

Output: $(\bar{x}_k, \bar{u}_k, Q_k, K_k)$

While the convergence guarantee of the proposed method has not been a focus of this paper, one can construct a safety alternative that is assured not to diverge by modifying the proposed algorithm with results from [25, 97]. Instead of updating the trajectory and the funnel sequentially in each iteration, the safety approach performs updating only

the trajectory with a fixed funnel until convergence of the nominal trajectory is achieved. This part of the process, being solely trajectory optimization, benefits from the established convergence results in [25]. The subsequent phase involves computing the Lipschitz constant and updating the funnel with the computed nominal trajectory, with convergence analysis from [97, Theorem 6.12]. Since each phase of the safety approach has a guaranteed convergence, it prevents the overall solution from diverging.

6.3 Numerical simulation

In this section, we validate the proposed method via two robotic applications with a unicycle model and a 6-DoF free-flying spacecraft. For both examples, we used an Apple MacBook Pro having M1 Pro with 8-core CPU.

6.3.1 Unicycle model

We consider the motion of a unicycle-type model under different disturbance conditions, represented by w_1 and w_2

$$\text{Model I: } \begin{bmatrix} \dot{r_x} \\ \dot{r_y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} u_v \cos\theta + 0.1w_1 \\ u_v \sin\theta + 0.1w_2 \\ u_\theta \end{bmatrix}, \quad \text{Model II: } \begin{bmatrix} \dot{r_x} \\ \dot{r_y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} (u_v + 0.1w_1)\cos\theta \\ (u_v + 0.1w_1)\sin\theta \\ u_\theta + 0.1w_2 \end{bmatrix}, \tag{6.30a}$$

$$\text{Model III: } \begin{bmatrix} \dot{r_x} \\ \dot{r_y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} u_v \cos(\theta + 0.03w_1) \\ u_v \sin(\theta + 0.03w_1) \\ u_\theta + 0.05w_2 \end{bmatrix}, \tag{6.30b}$$

where r_x, r_y, and θ are x-axis position, y-axis position, are heading angle, respectively, and $u_v \in \mathbb{R}$ is velocity and $u_\theta \in \mathbb{R}$ is angular velocity. The scalars w_1 and w_2 represent disturbances or model mismatch. Model I depicts direct disturbances on the translational motion, Model II introduces disturbances affecting both the velocity and rotational control inputs, and Model III captures disturbances influencing the orientation and the rotation control. These models are considered to have a comprehensive understanding of how the system behaves according to different types of the disturbances. It is worth noting that in

Model II and Model III, the disturbances introduce additional nonlinearities to the system, while in Model I, they appear as linear additive terms.

For all unicycle models, we consider $N = 30$ nodes evenly distributed over a time horizon of 10 s i.e., $t_0 = 0$ and $t_f = 10$. The continuous-time model (6.30) is discretized by following a variational approach [36, Chap. 10.4] to obtain the matrices A_k, B_k, F_k. The initial boundary set $\mathcal{X}_0$ and the final boundary set $\mathcal{X}_f$ in (6.8) have the following parameters: $x_0 = [0, 0, 0]^\top$, $Q_i = Q_f = \text{diag}([0.2^2 \text{ (m)}, 0.2^2 \text{ (m)}, 10^2 \text{ (deg)}]^\top)$, and $x_f = [8, 4, 0]^\top$. There are multiple circular obstacles the unicycle robot should avoid, which leads to nonconvex constraints on the state represented in set $\mathcal{X}$. All obstacles have a diameter of 1.0m, and their center positions are illustrated in Figure 6.3. The input constraints for the set $\mathcal{U}$ are given as: $0 \leq u_v \leq 1.5$ and $|u_\theta| \leq 1.0$ (rad). The cost function for the trajectory J_t is a quadratic function of the input given by $u_v^2 + u_\theta^2$. Both weight parameters w_Q, w_K in (6.7a) are chosen as 1. The decay rate α is set as 0.99 and the parameter λ_k^w is set as 0.2 for all k. The tolerance parameters $\Delta_{vc}^{tol}, \Delta_{dyn}^{tol}, \Delta_T^{tol}$ and Δ_F^{tol} are all 10^{-8}. The number of samples N_s used for the Lipschitz constant γ_k estimation is set as 100, for each k, so a total of 3,000 samples are used for each iteration. We use an interior-point method solver, Clarabel, for both the trajectory update (6.11) and the funnel update (6.20), using CVXPY in Python.

To test the invariance and the feasibility properties, we sample 100 points at the surface of the ellipse $\mathcal{E}_{Q_k}$ at $k = 0$, and then generate the corresponding 100 trajectories with the nonlinear dynamics (6.1) and the control law (6.3) under the presence of the disturbances. In this generation process, we randomly set the disturbance $w = (w_1, w_2)$ such that $\|w\|_2 = 1$ and keep them constant during the entire horizon for each sample. Note that making the disturbance constant during the entire horizon increases the impact of the disturbance compared to varying the disturbance randomly for each interval. The computed nominal trajectory and the CIF for all unicycle models (6.30) are depicted in Figure 6.3, and the input results are given in Figure 6.4. The test results of the invariance property for the trajectory samples is given in Figure 6.6 where the radius r_k^Q is defined as

$$r_k^Q \coloneqq (x_k^s - \bar{x}_k)^\top Q_k^{-1} (x_k^s - \bar{x}_k) \tag{6.31}$$

for each sample s and time k. The result shows that the nominal trajectory and the CIF

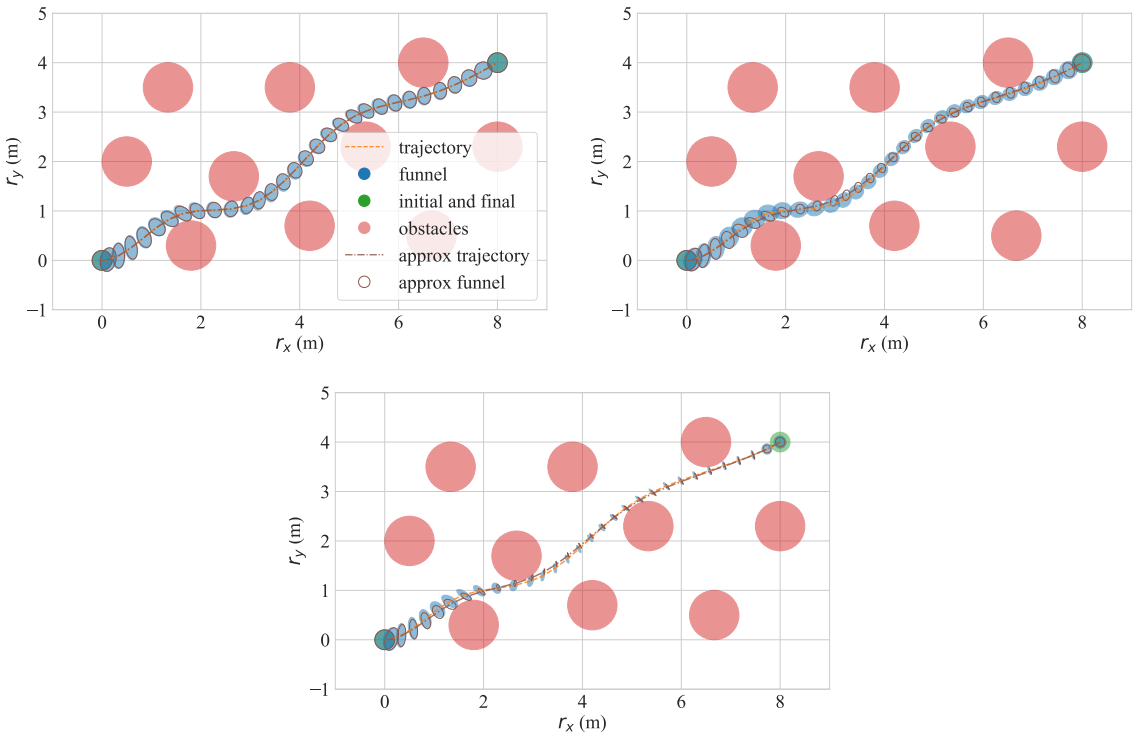

Figure 6.3: Nominal trajectories and synthesized funnels (projected on position coordinates) of Model I (top-left), Model II (top-right), and Model III (bottom). Each figure shows the nominal trajectory (orange line), the projection of the state ellipsoid in the funnel (blue ellipse), and the approximated funnel generated with the linear closed-loop system (brown ellipse).

satisfy the invariance and feasibility conditions. For the initial guess, we employ the first method illustrated in Sec. 6.2.3 using the straight-linear interpolation. The convergence performance in Figure 6.5 shows that the proposed approach makes the trajectory and the CIF satisfy the tolerances as the iteration count increases. Table 6.1 summarizes the average computational time of each subproblem within the iterations.

To obtain a baseline solution to compare against, we compute an approximate funnel that is generated with the linear closed-loop system where the higher-order terms are ignored, that can be established by setting $E_k = 0$ and $\gamma_k = 0$ in (6.14), as considered in [87, 78]. It is worth noting that the approximate funnel, which is used for the comparison with the proposed method, can yield more optimal solutions compared to [87, 78] under the linear approximation. This is because the approximate funnel is computed by simultaneously optimizing the linear feedback gains and the invariant set parameters as decision variables, whereas [87] determines the invariant set variables by the uncertainty forward equation and [78] sets the feedback gains by solving a discrete-time linear quadratic regulator problem. The approximate nominal trajectory and funnel are depicted in Figure 6.3, and the invariance test $r_k^Q \leq 1$ in (6.31) with the trajectory samples is given in Figure 6.6. We can see that the value of r_k^Q for the approximate funnel is greater than 1 especially for Model III since the bounded disturbances contribute the nonlinearity of the system. This violation shows that the approximate CIF does not necessarily guarantee the invariance property for the original nonlinear system, which can result in safety issues for safety-critical nonlinear systems. As the contribution of higher order terms increase, e.g., for large Lipshitz constants, these violations can become more pronounced.

Table 6.1: Average computational time (s) for each iteration

Subproblem	Trajectory update	Estimate γ_k	Funnel update
Model I	0.026	0.863	0.577
Model II	0.026	0.895	0.691
Model II	0.025	0.904	0.880

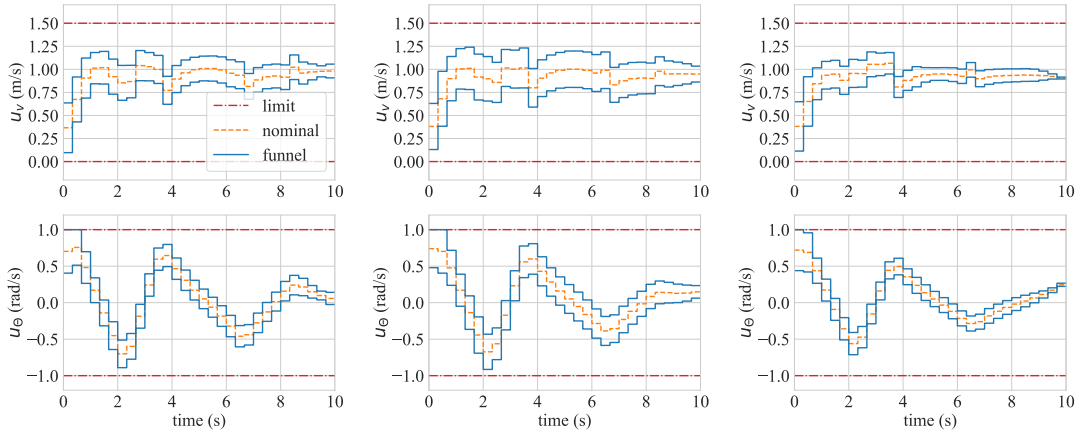

Figure 6.4: Nominal trajectories and synthesized input funnels (projected on each input coordinate) of Model I (left), Model II (middle), and Model III (right). The zeroth-order hold on the input is used to generate the nominal trajectory.

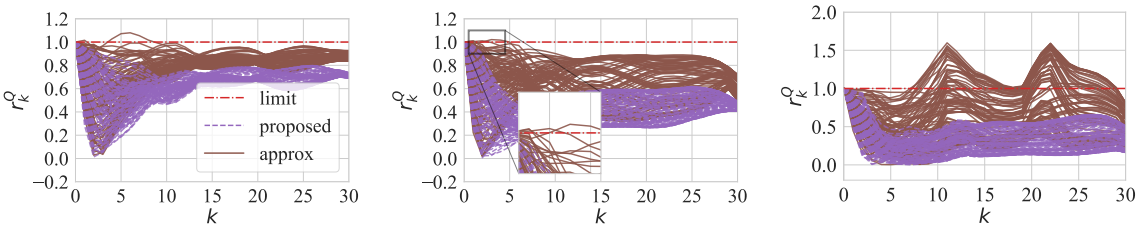

Figure 6.6: Invariance property tests for Model I (left), Model II (middle), and Model III (right).

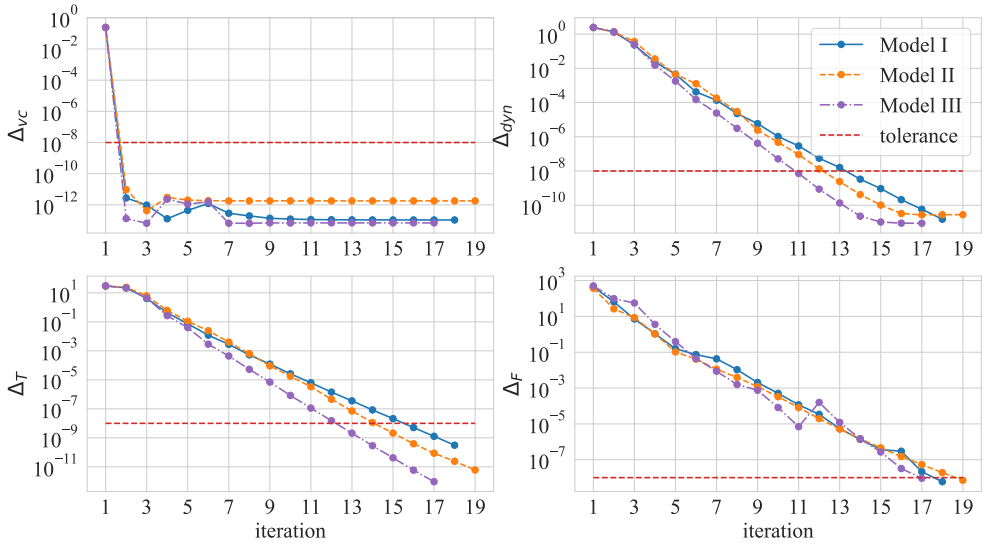

Figure 6.5: Convergence performance of the proposed method for the unicycle models.

6.3.2 6-DoF free-flying spacecraft

We consider the following 6-DoF free-flying spacecraft dynamics [77, 69] under the presence of disturbances:

$$\dot{r}_{\mathcal{I}} = v_{\mathcal{I}}, \tag{6.32a}$$

$$\dot{v}_{\mathcal{I}} = m^{-1}(T_{\mathcal{I}} + \beta_T w_T), \tag{6.32b}$$

$$\dot{\Phi} = R(\Phi)\omega_{\mathcal{B}}, \tag{6.32c}$$

$$\dot{\omega}_{\mathcal{B}} = J^{-1}(M_{\mathcal{B}} + \beta_M w_M - \omega_{\mathcal{B}} \times J\omega_{\mathcal{B}}). \tag{6.32d}$$

The state of the system (6.32) consists of the inertial position $r_{\mathcal{I}} \in \mathbb{R}^3$, the inertial velocity $v_{\mathcal{I}} \in \mathbb{R}^3$, the ZYX Euler angles $\Phi \in \mathbb{R}^3$, and the body angular velocity $\omega_{\mathcal{B}} \in \mathbb{R}^3$. The control input consists of the inertial thrust $T_{\mathcal{I}} \in \mathbb{R}^3$ and the body torque $M_{\mathcal{B}} \in \mathbb{R}^3$. The constant $m \in \mathbb{R}$ is the mass, matrix $J \in \mathbb{R}^{3\times 3}$ is the inertia matrix, and R maps the Euler angles to the rotation matrix

$$R(\Phi) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix},$$

with $\Phi = [\phi, \theta, \psi]^\top$. For the mass and inertia matrix parameters, we choose $m = 7.2$ (kg) and $J = 0.1083 I_{3\times 3}$ (kgm^2) where $I_{3\times 3}$ is the 3 by 3 identity matrix. The vector $w_T \in \mathbb{R}^3$ and $w_M \in \mathbb{R}^3$ affect the system as disturbances by acting on the control inputs $T_{\mathcal{I}}$ and $M_{\mathcal{B}}$ with coefficients $\beta_T = 10^{-3}$ and $\beta_M = 10^{-6}$. Further details about the free-flying system dynamics can be found in [77].

For the free-flying spacecraft example, we consider $N = 15$ nodes evenly distributed over a time horizon of 200 s. The initial boundary set $\mathcal{X}_0$ and the final boundary set $\mathcal{X}_f$ in (6.8) have the following parameters:

$$x_0 = [r_0^\top, v_0^\top, \Phi_0^\top, \omega_0^\top]^\top, r_0 = [0,0,3]^\top(\text{m}), v_0 = [0,0,0]^\top(\text{m/s}), \Phi_0 = [-30, 25, 5]^\top(\text{deg}),$$
$$\omega_0 = [0,0,0]^\top(\text{deg/s}), x_f = [r_f^\top, v_f^\top, \Phi_f^\top, \omega_f^\top]^\top, r_f = [3,3,0]^\top(\text{m}), v_f = \Phi_f = \omega_f = [0,0,0]^\top,$$
$$Q_i = Q_f$$
$$= \text{diag}\left(\left[0.2^2, 0.2^2, 0.2^2, 0.02^2, 0.02^2, 0.02^2, (5^\circ)^2, (5^\circ)^2, (5^\circ)^2, (0.1^\circ)^2, (0.1^\circ)^2, (0.1^\circ)^2\right]\right).$$

As state constraints for the set $\mathcal{X}$, there are two cylindrical obstacles to avoid, and all obstacles have a diameter of 0.8m, and their center positions are illustrated in Figure 6.7. In addition, we have constraints on the velocity and the angular velocity that are $\|v_{\mathcal{I}}\|_2 \leq 0.4$ (m/s) and $\|\omega_{\mathcal{B}}\|_2 \leq 1$ (deg). The input constraints for the set $\mathcal{U}$ are given as: $\|T_{\mathcal{I}}\|_2 \leq 10$ (mN) and $\|M_{\mathcal{B}}\|_2 \leq 50$ (μNm). The decay rate α is set as 0.99 and the parameter λ_k^w is set as 0.1 for all k. Similar to the unicycle examples, the tolerance parameters $\Delta_{vc}^{tol}, \Delta_{dyn}^{tol}, \Delta_T^{tol}$ and Δ_F^{tol} are all 10^{-8}. The number of samples N_s used for the Lipschitz constant γ_k estimation is set as 256 for each k. We provide the initial guess using the second method illustrated in Sec. 6.2.3 that uses the result of the separate synthesis, and these used initial trajectory and funnel are illustrated in Figure 6.7. Starting from the initial guess, the proposed algorithm converges at 6 iterations. We use Clarabel for the trajectory update (6.11) and MOSEK for the funnel update (6.20), using CVXPY in Python3. Mosek's solve time is observed to scale better than Clarabel's for large problem sizes. So, for the free-flyer system, we have used Mosek for the funnel update. The average computational time (s) of the trajectory update, the estimation of γ_k, and the funnel update at each iteration are 0.024, 2.698, and 10.072, respectively.

The results of the synthesized trajectory and funnel projected on position coordinates are illustrated in Figure 6.7. It is clear that the resulting funnel is feasible to the obstacle avoidance constraints although the initial guess has infeasible trajectory and funnel. Similar to the test performed for the unicycle models, to test the invariance of the synthesized trajectory and funnel, we sample 300 at the surface of the initial funnel $\mathcal{E}_{Q_0}$ and generate the corresponding 300 trajectories by propagating the system dynamics (6.32) with the control law (6.3) consisting of the open-loop input in the nominal trajectory and the feedback control from the funnel. For each sample, we randomly set the disturbance $w = [w_T^\top, w_M^\top]^\top$ such that $\|w\|_2 = 1$. The input results of the nominal trajectory and the samples are illustrated in Figure 6.8. We can see that the input history of the samples remain feasible within the given input constraints. Finally, we obtain the values of r_k^Q by computing (6.31) for each trajectory sample, illustrated in Figure 6.9. The result shows that r_k^Q for each sample and for all time k maintain less than 1 and hence the trajectory samples remain inside the funnel.

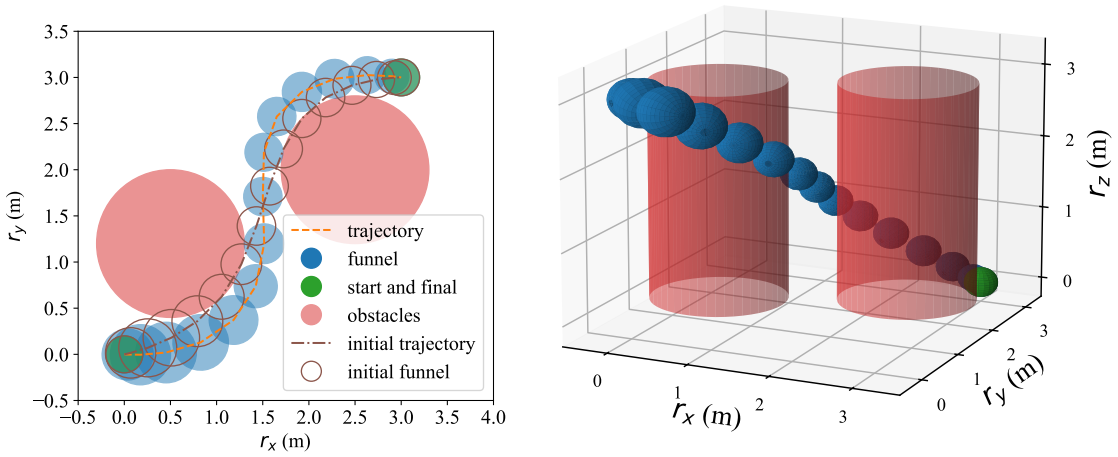

Figure 6.7: Nominal trajectories and synthesized funnel projected on $r_x r_y$ positions (left) and $r_x r_y r_z$ positions (right), respectively, for the free-flying spacecraft.

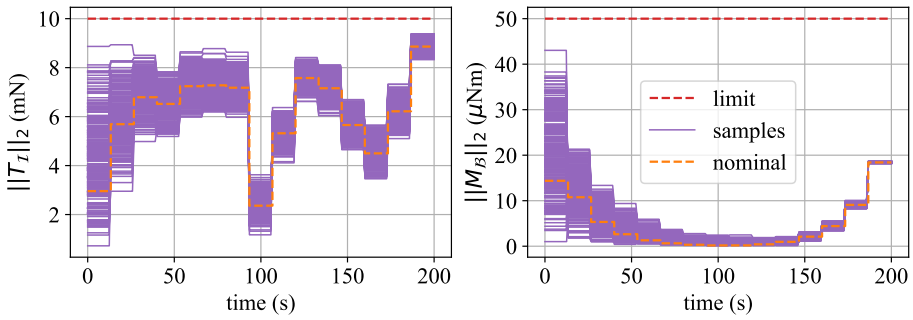

Figure 6.8: The thrust and moment results of nominal trajectory and trajectory samples for the free-flying spacecraft.

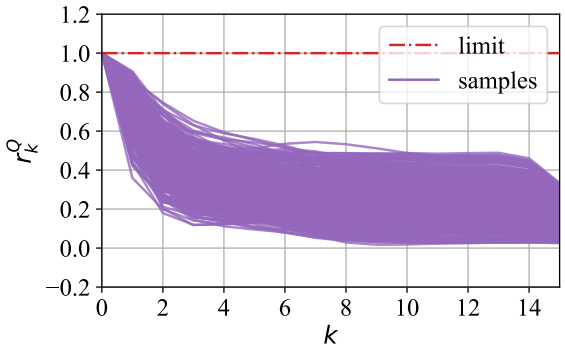

Figure 6.9: Invariance property tests for trajectory samples of the free-flying spacecraft.

6.4 Conclusions

This paper introduces a method for joint trajectory optimization and funnel synthesis for locally Lipschitz nonlinear systems under the presence of disturbances. The proposed method has a recursive approach in which both nominal trajectory and funnel are iteratively updated. The trajectory update step optimizes the nominal trajectory to satisfy the feasibility of the funnel. Then, the funnel update step solves an SDP to guarantee the invariance property of the funnel. The numerical evaluation for a unicycle model and a 6-DoF free-flying spacecraft shows that the converged trajectory and funnel satisfy the invariance and feasibility properties under the disturbances.

Chapter 7

JOINT SYNTHESIS VIA SUCCESSIVE CONVEXIFICATION

This chapter extends the joint synthesis framework by employing a penalized trust region (PTR) variant of the successive convexification method, which is aligned with the prox-linear optimization structure. This ensures convergence guarantees and handles nonconvexities in a principled way.

Chapter-specific notation. This chapter uses the same notation as that of Chapter 3.

7.1 Continuous-time joint synthesis problem

7.1.1 Nonlinear systems

Consider the following continuous-time nonlinear systems:

$$\dot{x}(t) = f(t, x(t), u(t)), \quad t \in [t_0, t_f], \tag{7.1}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state and $u(t) \in \mathbb{R}^{n_u}$ is the control input, and $f : [t_0, t_f] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is assumed to be continuously twice differentiable with respect to its second and third arguments. The time instances t_0 and t_f are the initial and final time, respectively, such that $t_0 < t_f < \infty$. They are assumed to be fixed through this section.

The nominal trajectory, denoted by $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$, or more compactly $(\bar{x}, \bar{u})$, is assumed to be dynamically feasible; that is, it satisfies

$$\dot{\bar{x}}(t) = f(t, \bar{x}, \bar{u}). \tag{7.2}$$

With the nominal trajectory, the original nonlinear system dynamics (7.1) can be expressed as the following linear fractional form:

$$\dot{x}(t) = A(t, \bar{z}(t))x(t) + B(t, \bar{z}(t))u(t) + Ep(t), \tag{7.3a}$$

$$p(t) = \phi(t, q(t)), \tag{7.3b}$$

$$q(t) = Cx(t) + Du(t), \tag{7.3c}$$

where $\bar{z}(t) \in \mathbb{R}^{n_x+n_u}$ denotes the concatenation of the nominal state and input, i.e., $\bar{x}$ and $\bar{u}$, $\bar{z} = [\bar{x}^\top, \bar{u}^\top]^\top$. The matrix-valued functions A and B are the Jacobian matrices evaluated at the nominal trajectory given by

$$A(t, \bar{z}(t)) = \left.\frac{\partial f}{\partial x}\right|_{z=\bar{z}(t)}, \; B(t, \bar{z}(t)) = \left.\frac{\partial f}{\partial u}\right|_{z=\bar{z}(t)}.$$

The vector $p(t) \in \mathbb{R}^{n_p}$ represents the remainder term arising from the linearization. The selector matrix E, composed of zeros and ones, is introduced to account for the fact that some components of the state dynamics may already be linear. The function $\phi : [t_0, t_f] \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_p}$ captures all nonlinearities in this system, with its argument given by the vector $q(t) \in \mathbb{R}^{n_q}$. The argument q is constructed to be linear in the state and input, using the selector matrices $C \in \mathbb{R}^{n_q \times n_x}$ and $D \in \mathbb{R}^{n_q \times n_u}$.

For the pair (q,p), we assume that they can be decomposed as follows:

$$p = \begin{bmatrix} p_1 \\ \vdots \\ p_{n_\phi} \end{bmatrix}, \quad q = \begin{bmatrix} q_1 \\ \vdots \\ q_{n_\phi} \end{bmatrix},$$

where $n_\phi \in \mathbb{R}_+$ represents the number of nonlinearity channels, and each $p_i \in \mathbb{R}^{n_{p_i}}$ and $q_i \in \mathbb{R}^{n_{q_i}}$ corresponds to the output and input of i-th nonlinear channel, respectively. For each channel $i \in \mathbb{Z}_{[1,n_\phi]}$, we assume the existence of a nonnegative scalar-valued function $\gamma_i : [t_0, t_f] \to \mathbb{R}_+$ such that, for any compact set $\Omega \subseteq \mathbb{R}^{n_{q_i}}$, the following inequality holds:

$$\|p_i(t) - \bar{p}_i(t)\|_2 \leq \gamma_i(t)\|q_i(t) - \bar{q}_i(t)\|_2, \quad \forall q, \bar{q} \in \Omega,$$

where $\bar{p}_i(t) = \phi(t, \bar{q}_i(t))$. Further details on the construction of such nonlinearity channels can be found in [99].

7.1.2 Incremental form of dynamics and funnels

To derive an incremental system dynamics that represents the system behavior with respect to the nominal trajectory, we define deviation variables:

$$\eta(t) \coloneqq x(t) - \bar{x}(t), \quad \xi(t) = u(t) - \bar{u}(t),$$
$$\delta q(t) \coloneqq q(t) - \bar{q}(t), \quad \delta p(t) = p(t) - \bar{p}(t).$$

With the deviation variables, the incremental system dynamics is given by

$$\dot{\eta}(t) = A(t, \bar{z}(t))\eta(t) + B(t, \bar{z}(t))\xi(t) + E\delta p(t), \tag{7.4a}$$

$$\delta q(t) = C\eta(t) + D\xi(t). \tag{7.4b}$$

Consider a scalar-valued Lyapunov function $V : [t_0, t_f] \times \mathbb{R}^{n_x} \to \mathbb{R}_+$ defined by

$$V(t, \eta(t)) = \eta(t)^\top Q(t)^{-1}\eta(t),$$

where $Q : [t_0, t_f] \to \mathbb{S}^{n_x}_{++}$ is piecewise continuous and continuously differentiable. A *state funnel* is defined as the sub-level set of V:

$$\mathcal{E}_\eta(t) = \{\eta \mid \eta(t)^\top Q(t)^{-1}\eta(t) \leq 1\}.$$

For incremental system dynamics (7.4), we employ the linear feedback control $\eta(t) = K(t)\xi(t)$ where $K : [t_0, t_f] \to \mathbb{R}^{n_u \times n_x}$ is the piecewise continuous feedback gain.

7.1.3 Funnel's invariance by Lyapunov condition and DMI

To ensure the invariance of the funnel, we use the Lyapunov condition

$$\dot{V}(t, \eta(t)) \leq -\alpha V(t, \eta(t)), \quad t \in [t_0, t_f], \tag{7.5}$$

where $\alpha \in \mathbb{R}_+$ is the decay rate. It is worth noting that setting $\alpha = 0$ is still sufficient for the invariance of the funnel.

Now we derive a differential matrix inequality (DMI) implying the Lyapunov condition (7.5). The DMI is given by

$$\begin{bmatrix} H_1(t) - \dot{Q}(t) & \star & \star \\ N_2(t)E^\top & -N_2(t) & \star \\ CQ(t) + DY(t) & 0 & -N_1(t) \end{bmatrix} \preceq 0, \tag{7.6}$$

where the matrix block H_1 is defined by:

$$H_1(t) = A(t, \bar{z}(t))Q(t) + Q(t)A(t, \bar{z}(t))^\top + B(t, \bar{z}(t))Y(t) + Y(t)^\top B(t, \bar{z}(t))^\top + \alpha Q(t), \tag{7.7}$$

where the change of variable $Y(t) = K(t)Q(t)$ is applied.

Remark 7.1. *In* (3.25)*, the nominal trajectory $\bar{z}$ was assumed to be fixed, which allowed the DMI* (7.6) *to be linear in decision variables. However, in the joint synthesis formulation, $\bar{z}$ is no longer predefined but is instead treated as a decision variable to be optimized. As a consequence, the DMI becomes nonlinear, since the system matrices A and B now depend on $\bar{z}$ in a non-affine manner when the system is not linear.*

7.1.4 Funnel constraints

We consider the following state and input constraint sets:

$$\mathcal{X} = \bigcap_{i=1}^{m_x} \{x \mid h_i(x) \leq 0\}, \quad \mathcal{U} = \bigcap_{j=1}^{m_u} \{u \mid (a_j^g(t))^\top u \leq b_j^g(t)\}, \tag{7.8}$$

where $h_i : \mathbb{R}^{n_x} \to \mathbb{R}$ is assumed to be twice continuously differentiable for all $i \in \mathbb{Z}_{[1,m_x]}$, and $m_x, m_u \in \mathbb{R}_+$ are the number of the state and input constraint sets, respectively. The state constraint set $\mathcal{X}$ is defined by general nonlinear function h_i whereas the input constraint set is defined by a finite set of linear inequalities, and thus forms a polyhedron.

The goal of imposing these constraints is to ensure that all states and inputs resulting from the funnel remain within their respective constraint sets. This can be logically written by the following logical statement:

$$\text{If } \eta \in \mathcal{E}_\eta(t), \text{ then } \bar{x}(t) + \eta \in \mathcal{X} \text{ and } \bar{u}(t) + K(t)\eta \in \mathcal{U}, \quad \forall t \in [t_0, t_f].$$

To achieve this, we impose the following pointwise-in-time LMIs:

$$0 \preceq \begin{bmatrix} h_i(\bar{x}(t))^2 & a_i^h(\bar{x}(t))^\top Q(t) \\ Q(t) a_i^h(\bar{x}(t)) & Q(t) \end{bmatrix}, \tag{7.9a}$$

$$0 \preceq \begin{bmatrix} (b_j^g(t) - a_j^g(t)^\top \bar{u}(t))^2 & a_i^g(t)^\top Y(t) \\ Y(t) a_i^h(t) & Q(t) \end{bmatrix}, \tag{7.9b}$$

where $a_i^h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ is defined by

$$a_i^h(\bar{x}(t)) \coloneqq \nabla_x h_i(x)\Big|_{x=\bar{x}(t)}.$$

Both constraints (7.9) are nonconvex with respect to decision variables.

Now we are ready to construct the continuous-time joint synthesis problem.

7.1.5 Problem formulation

Problem 7.2. *Continuous-time joint synthesis.*

$$\min_{\bar{x},\bar{u},Q,Y,N_1,N_2} \; J_{\mathrm{f}}(Q(t_0)) + \int_{t_0}^{t_f} J_{\mathrm{t}}(\bar{x}(\tau),\bar{u}(\tau))\,d\tau \tag{7.10a}$$

$$\text{s.t. } (7.2),(7.6),(7.9),$$

$$Q(t) \succ 0, \tag{7.10b}$$

$$(N_1(t),N_2(t)) \in \mathcal{N}(t), \tag{7.10c}$$

$$x(t_f) = x_f, \quad Q(t_f) \preceq Q_f, \tag{7.10d}$$

$$\forall t \in [t_0,t_f].$$

where $J_{\mathrm{f}} : \mathbb{S}^{n_x}_{++} \to \mathbb{R}$ *represents the convex funnel cost and* $J_{\mathrm{t}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ *is the convex trajectory running cost. The vector* $x_f \in \mathbb{R}^{n_x}$ *is the terminal state and* $Q_f \in \mathbb{S}^{n_x}_{++}$ *is the terminal state funnel. The inclusion* (7.10c) *for the multiplier matrices can be represented by linear equality constraints as follows:*

$$N_1(t) = \mathrm{blkdiag}(\lambda_1^{\gamma}(t)I_{n_{q_1}},\ldots,\lambda_{n_\phi}^{\gamma}(t)I_{n_{q_{n_\phi}}}), \tag{7.11a}$$

$$N_2(t) = \mathrm{blkdiag}(\lambda_1^{\gamma}(t)\gamma_1^2(t)I_{n_{p_1}},\ldots,\lambda_{n_\phi}^{\gamma}(t)\gamma_{n_\phi}^2(t)I_{n_{p_{n_\phi}}}), \tag{7.11b}$$

where $\lambda_i^{\gamma} : [t_0,t_f] \to \mathbb{R}_+$ *are positive-valued functions as decision variables with constraints* $\lambda_i^{\gamma}(t) \geq 0$.

The constructed joint synthesis problem is nonconvex because of constraints (7.2),(7.6)*, and* (7.9)*. Hence, in the next section, we develop a solution method based on nonconvex optimization.*

7.2 Solution method by successive convexification

Our key idea to solve the continuous-time nonconvex joint synthesis problem (7.10) is to apply continuous-time successive convexification (SCvx) [40], a variant of sequential convex programming originally developed for solving optimal control problems. In this chapter, we present how the joint synthesis problem can be reformulated as an equivalent optimal control problem. We then detail the process of discretization and convexification required

to make the problem tractable for an iterative solution method using sequential convex programming techniques.

7.2.1 Reformulation to an optimal control problem

To reformulate the joint synthesis problem into a standard optimal control framework, we introduce a slack variable $Z : [t_0, t_f] \to \mathbb{S}^{n_x}$ that satisfies the differential linear matrix equality constraint:

$$\dot{Q}(t) = Z(t). \tag{7.12}$$

This substitution allows us to explicitly treat $Q(t)$ as a state variable with its own dynamics, which is necessary to cast the problem into an optimal control problem. With this reformulation, the original DMI (7.6) constraint can be equivalently decomposed into the funnel dynamics (7.12) and the following pointwise-in-time (nonlinear) matrix inequality:

$$\begin{bmatrix} H_1(t) - Z(t) & \star & \star \\ N_2(t)E^\top & -N_2(t) & \star \\ CQ(t) + DY(t) & 0 & -N_1(t) \end{bmatrix} \preceq 0. \tag{7.13}$$

To simplify the analysis and facilitate optimization, we introduce vectorized variables :

$$q_v \coloneqq \mathrm{vec}(Q), \quad y_v = \mathrm{vec}(Y), \quad z_v = \mathrm{vec}(Z).$$

Using these, we define the augmented state and input variables as:

$$\mathbf{s} = \begin{bmatrix} \bar{x} \\ q \end{bmatrix} \in \mathbb{R}^{n_s}, \quad \mathbf{a} = \begin{bmatrix} \bar{u} \\ y_v \\ z_v \\ \lambda^\gamma \end{bmatrix} \in \mathbb{R}^{n_a},$$

where $n_s = n_x + n_x^2$ and $n_a = n_u + n_x n_u + n_x^2 + n_\phi$. The corresponding augmented system dynamics are then given by:

$$\dot{\mathbf{s}} = F(t, \mathbf{s}(t), \mathbf{a}(t)) \coloneqq \begin{bmatrix} f(t, \bar{x}(t), \bar{u}(t)) \\ Z(t) \end{bmatrix}. \tag{7.14}$$

With this reformulation, the original continuous-time joint synthesis problem (7.10) is equivalently cast as the following optimal control problem:

$$\min_{\mathbf{s},\mathbf{a}} J_{\mathrm{f}}(Q(t_0)) + \int_{t_0}^{t_f} J_{\mathrm{t}}(\bar{x}(\tau), \bar{u}(\tau))\mathrm{d}\tau \tag{7.15a}$$

$$\text{s.t. } \dot{\mathbf{s}}(t) = F(t, \mathbf{s}(t), \mathbf{a}(t)), \tag{7.15b}$$

$$L_l^{in}(t, \mathbf{s}(t), \mathbf{a}(t)) \preceq 0, \quad l \in \mathbb{Z}_{[1,m_l]}, \tag{7.15c}$$

$$x(t_f) = x_f, \quad Q(t_f) \preceq Q_f, \quad \forall t \in [t_0, t_f]. \tag{7.15d}$$

Here all (possibly nonconvex) matrix inequality constraints are collected into L_l^{in}. Specifically, the index sets for L_l^{in} are organized as follows:

- $l \in \mathbb{Z}_{[1,m_{inv}]}$: invariance condition (7.13), where $m_{inv} = 1$,
- $l \in \mathbb{Z}_{[m_{inv}+1,m_{inv}+n_\phi]}$: constraints on $\lambda_l^\gamma \geq 0$ for each $l \in \mathbb{Z}_{[1,n_\phi]}$,
- $l \in \mathbb{Z}_{[m_{inv}+n_\phi+1,m_l]}$: state constraints (7.9a) and input constraints (7.9b) where $m_l = m_{inv} + n_\phi + m_x + m_u$.

7.2.2 Time discretization and control parameterization

We discretize the time horizon $[t_0, t_f]$ using a uniform time grid defined by:

$$t_k = t_0 + \frac{k}{N}(t_f - t_0), \quad k \in \mathbb{Z}_{[0,N]}, \tag{7.16a}$$

$$\triangle_k = \triangle(t_k), \tag{7.16b}$$

where $N \in \mathbb{Z}_{++}$ is the number of subintervals. The symbol $\triangle$ serves as a placeholder for any time-varying variable, and we denote its value at $t = t_k$ by $\triangle_k := \triangle(t_k)$. Each $\triangle_k$ is referred to as a node point. We apply a continuous first-order hold (FOH) interpolation over each time interval $[t_k, t_{k+1}]$ for the control inputs as follows:

$$\mathbf{a}(t) = \lambda_k^m(t)\mathbf{a}_k + \lambda_k^p(t)\mathbf{a}_{k+1}, \quad \forall t \in [t_k, t_{k+1}), \tag{7.17a}$$

$$\lambda_k^m(t) = \frac{t_{k+1} - t}{t_{k+1} - t_k}, \quad \lambda_k^p(t) = \frac{t - t_k}{t_{k+1} - t_k}, \quad \forall k \in \mathbb{Z}_{[0,N-1]}. \tag{7.17b}$$

We adopt a multiple-shooting scheme [16] to enforce the augmented dynamics (7.14). Over each subinterval $[t_k, t_{k+1})$, the augmented state evolves according to

$$\mathbf{s}(t) = \mathbf{s}(t_k) + \int_{t_k}^{t} F(\tau, \mathbf{s}(\tau), \mathbf{a}(\tau))\mathrm{d}\tau.$$

Enforcing continuity across subintervals yields the discrete augmented dynamics constraint:

$$\mathbf{s}_{k+1} = \underbrace{\mathbf{s}(t_k) + \int_{t_k}^{t_{k+1}} F(\tau, \mathbf{s}(\tau), \mathbf{a}(\tau))\mathrm{d}\tau}_{:= F_d(t_k, \mathbf{s}_k, \mathbf{a}_k, \mathbf{a}_{k+1})}. \tag{7.18}$$

Using this discretization and control parameterization, the original continuous-time optimal control problem is approximated by the following discrete-time optimal control problem:

$$\min_{\mathbf{s}_k, \mathbf{a}_k} \; J_{\mathrm{f}}(Q(t_0)) + \sum_{k=0}^{N} J_{\mathrm{t}}(\bar{x}_k, \bar{u}_k) \tag{7.19a}$$

$$\text{s.t. } \mathbf{s}_{k+1} = F_d(t_k, \mathbf{s}_k, \mathbf{a}_k, \mathbf{a}_{k+1}), \quad \forall k \in \mathbb{Z}_{[0,N-1]}, \tag{7.19b}$$

$$L_l^{in}(t_k, \mathbf{s}_k, \mathbf{a}_k) \preceq 0, \quad l \in \mathbb{Z}_{[1,m_l]}, \quad \forall k \in \mathbb{Z}_{[0,N]}, \tag{7.19c}$$

$$x_N = x_f, \quad Q(t_f) \preceq Q_f, \quad \forall t \in [t_0, t_f]. \tag{7.19d}$$

Here, the continuous-time constraint satisfaction (7.15c) is approximated via nodal constraint enforcement (7.19c), meaning that the constraints are imposed only at the discretized node points.

7.2.3 Convex Subproblem Construction in SCvx

To solve nonconvex discrete-time optimal control problem (7.19), we employ the SCvx method. SCvx iteratively constructs and solves a sequence of convex subproblems that locally approximate the original nonconvex problem. At each iteration, a convex subproblem is formed around a reference solution and optimized to update the trajectory. This process continues until convergence criteria are met. We denote the reference solution, which is either the initial guess at the first iteration or the previous iteration's solution, by $\hat{\square}$ where the placeholder $\square$ refers to any decision variables (e.g., $\hat{\mathbf{s}}_k, \hat{\mathbf{a}}_k$).

The discrete-time augmented system dynamics (7.18) is linearized around the reference solution $(\hat{\mathbf{s}}, \hat{\mathbf{a}})$, resulting in the following discrete-time linear constraints:

$$\mathbf{s}_{k+1} = A_k \mathbf{s}_k + B_k^m \mathbf{a}_k + B_k^p \mathbf{a}_{k+1} + z_k, \quad \forall k \in \mathbb{Z}_{[0,N-1]}, \tag{7.20}$$

where A_k,B_k^m, B_k^p, and z_k are appropriately sized matrices and vectors that can be obtained via variational method [72].

The matrix inequality constraints (7.19c) are similarly linearized around the reference solution as follows:

$$L_l^{in}(t_k, \hat{\mathbf{s}}_k, \hat{\mathbf{a}}_k) + \sum_{i=0}^{n_s} \frac{\partial L_l^{in}}{\partial \mathbf{s}_i}(t_k, \hat{\mathbf{s}}_k, \hat{\mathbf{a}}_k)(\mathbf{s}_i - \hat{\mathbf{s}}_i) + \sum_{i=0}^{n_s} \frac{\partial L_l^{in}}{\partial \mathbf{a}_i}(t_k, \hat{\mathbf{s}}_k, \hat{\mathbf{a}}_k)(\mathbf{a}_i - \hat{\mathbf{a}}_i) \preceq 0 \tag{7.21}$$

where $\mathbf{s}_i$ and $\mathbf{a}_i$ denote the i-th element of the augmented state and input vectors, respectively. Assuming $L_l^{in} \in \mathbb{S}^{n_{L_l}}$, the partial derivative $\frac{\partial L_l^{in}}{\partial \mathbf{s}_i}$ is defined as the matrix of directional derivative of each entry of L_l^{in} with respect to the scalar variable $\mathbf{s}_i$, evaluated at the reference solution. That is,

$$\left[\frac{\partial L_l^{in}}{\partial \mathbf{s}_i}\right]_{p,q} := \frac{\partial \left[L_l^{in}\right]_{p,q}}{\partial \mathbf{s}_i}, \quad \forall p, q \in \mathbb{Z}_{[1,n_{L_l}]},$$

and similarly for $\frac{\partial L_l^{in}}{\partial \mathbf{a}_i}$.

The resulting convex subproblem to be solved at each SCvx iteration is formulated as:

$$\min_{\mathbf{s}_k, \mathbf{a}_k} \; J_{\text{all}} + w_{vc} \sum_{k=0}^{N-1} \|v_k\|_1 + w_{tr} \sum_{k=0}^{N} \left(\|\mathbf{s}_k - \hat{\mathbf{s}}_k\|_2^2 + \|\mathbf{a}_k - \hat{\mathbf{a}}_k\|_2^2\right) \tag{7.22a}$$

$$\text{s.t. } \mathbf{s}_{k+1} = A_k \mathbf{s}_k + B_k^m \mathbf{a}_k + B_k^p \mathbf{a}_{k+1} + z_k + v_k, \quad \forall k \in \mathbb{Z}_{[0,N-1]}, \tag{7.22b}$$

$$(7.21), \quad l \in \mathbb{Z}_{[1,m_l]}, \quad \forall k \in \mathbb{Z}_{[0,N]}, \tag{7.22c}$$

$$Q_N \preceq Q_f, \quad \forall t \in [t_0, t_f], \tag{7.22d}$$

where $J_{\text{all}} = J_{\text{f}}(Q(t_0)) + \sum_{k=0}^{N} J_{\text{t}}(\bar{x}_k, \bar{u}_k)$. Here slack variables v_k are introduced to softly enforce the linearized dynamics, allowing minor violations. This improves convergence by avoiding infeasibility possibly caused by linearization. The ℓ_1-norm penalty on v_k, weighted by w_{vc}, promotes sparsity in constraint violations, making the solver favor feasible updates whenever possible. The term weighted by $w_{\text{tr}} \in \mathbb{R}_{++}$ defines a trust region penalty, which keeps the new solution close to the reference and prevents large deviations between iterations. This subproblem is a semidefinite programming (SDP) problem and can be solved efficiently using standard convex optimization solvers.

7.2.4 Details in convexification

In the previous subsection, all pointwise-in-time matrix inequality constraints were collectively represented using the unified notation L_l^{in}. In this section, we unpack these constraints and describe how each type is individually linearized around the reference trajectory.

Consider the DMI for the funnel's invariance (7.13). The nonlinearities in the inequality arise primarily from bilinear terms such as $A(t,\bar{z}(t))Q(t)$, $B(t,\bar{z}(t))Y(t)$, and their transposes, as defined in the matrix $H_1(t)$ (7.7). Each bilinear term can be linearized around the reference trajectory $(\hat{x},\hat{u})$. The terms $A(t,\bar{z}(t))Q(t)$ and $B(t,\bar{z}(t))Y(t)$ can be approximated as:

$$\begin{aligned}
A(t,\bar{z}(t))Q(t) &\approx A(t,\hat{z}(t))Q(t) + \sum_{i=1}^{n_x} \frac{\partial A}{\partial x_i}(t,\hat{z}(t))\hat{Q}(t)(\bar{x}_i - \hat{x}_i) \\
&\quad + \sum_{i=1}^{n_u} \frac{\partial B}{\partial u_i}(t,\hat{z}(t))\hat{Q}(t)(\bar{u}_i - \hat{u}_i), \\
B(t,\bar{z}(t))Q(t) &\approx B(t,\hat{z}(t))Y(t) + \sum_{i=1}^{n_x} \frac{\partial B}{\partial x_i}(t,\hat{z}(t))\hat{Y}(t)(\bar{x}_i - \hat{x}_i) \\
&\quad + \sum_{i=1}^{n_u} \frac{\partial B}{\partial u_i}(t,\hat{z}(t))\hat{Y}(t)(\bar{u}_i - \hat{u}_i),
\end{aligned}$$

where the second-order partial derivatives of the system dynamics are defined as:

$$\begin{aligned}
\frac{\partial A}{\partial x_i} &:= \frac{\partial^2 f}{\partial x_i \partial x}, \quad \frac{\partial A}{\partial u_i} := \frac{\partial^2 f}{\partial u_i \partial x}, \\
\frac{\partial B}{\partial x_i} &:= \frac{\partial^2 f}{\partial x_i \partial u}, \quad \frac{\partial B}{\partial u_i} := \frac{\partial^2 f}{\partial u_i \partial u}.
\end{aligned}$$

Next, consider the state and input constraints in (7.9a) and (7.9b), respectively. The nonlinearities arise from terms $h_i(\bar{x}(t))^2$, $Q(t)a_i^h(\bar{x}(t))$, and $(b_j^g(t) - a_j^g(t)^\top \bar{u}(t))^2$. Each term can be linearized as follows:

$$\begin{aligned}
h_i(\bar{x}(t))^2 &\approx h_i(\hat{x}(t))^2 + 2h_i(\hat{x}(t))a_i^h(\hat{x}(t))^\top(\bar{x}(t) - \hat{x}(t)), \\
Q(t)a_i^h(\bar{x}(t)) &\approx Q(t)a_i^h(\hat{x}(t)) + \hat{Q}(t)\frac{\partial^2 h_i}{\partial x^2}(\hat{x}(t))(\bar{x}(t) - \hat{x}(t)), \\
(b_j^g(t) - a_j^g(t)^\top \bar{u}(t))^2 &\approx (b_j^g(t) - a_j^g(t)^\top \hat{u}(t))^2 \\
&\quad + 2(b_j^g(t) - a_j^g(t)^\top \hat{u}(t)) \left(-a_j^g(t)^\top\right)(\bar{u}(t) - \hat{u}(t)).
\end{aligned}$$

This completes the convexification step required for each constraint in (7.15).

7.2.5 Summary of algorithm

The proposed algorithm is summarized in Algorithm 4. The constant $\epsilon \in \mathbb{R}_{++}$ serves as a stopping criteria and $N_{max} \in \mathbb{R}_{++}$ is the maximum number of iterations. The similar strategy for the initial guess, illustrated in Chapter 6.2.3, can be employed. Particularly, the initial guess for the variables Z_k and λ_k^γ can be chosen arbitrarily (for example, zero), since they appear linearly in all constraints.

One advantage of the proposed joint synthesis method over that of Chapter 6 is its convergence guarantee. Because it employs SCvx in the prox-linear form, the joint synthesis inherits the convergence properties illustrated in Lemma 2.4.1.

Algorithm 4 Joint synthesis

Input: Initial guess on $(\hat{\mathbf{s}}_k, \hat{\mathbf{a}}_k)$

for $i = 1 \dots N_{max}$ **do**

Optimize $\mathbf{s}_k, \mathbf{a}_k$ by solving (7.22)

Update $(\hat{\mathbf{s}}_k, \hat{\mathbf{a}}_k) \leftarrow (\mathbf{s}_k, \mathbf{a}_k)$

if $\sum_{k=0}^{N} \left(\|\mathbf{s}_k - \hat{\mathbf{s}}_k\|_2^2 + \|\mathbf{a}_k - \hat{\mathbf{a}}_k\|_2^2 \right) < \epsilon$ **then**

break

end if

end for

Output: $(\mathbf{s}_k, \mathbf{a}_k)$

7.3 Numerical simulations

The goal of this section is to demonstrate the proposed joint synthesis algorithm. Here, we consider the following double integrator system:

$$x = \begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{v}_x \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ a_x - c_d\|v\|_2 v_x \\ a_y - c_d\|v\|_2 v_y \end{bmatrix},$$

where r_x and r_y denote x and y positions, and v_x and v_y are the corresponding velocity components. The control input $u = (a_x, a_y)$ represents the x and y accelerations, respectively. The constant c_d represents the drag coefficients set to $c_d = 0.1$ and $v = (v_x, v_y)$ denotes the velocity vector. The terms $c_d\|v\|_2 v_x$ and $c_d\|v\|_2 v_y$ represent aerodynamic drags in x and y coordinates. The following constraints and associated parameters are considered for the simulation:

$$|v_x| \leq 2.0 \text{ (m/s)}, \ |a_x| \leq 2.5 \text{ m/s}^2,$$
$$|v_y| \leq 2.0 \text{ (m/s)}, \ |a_y| \leq 2.5 \text{ m/s}^2,$$
$$Q_f = \text{diag}(0.2^2, 0.2^2, 0.1^2, 0.1^2).$$

Additionally, obstacle avoidance is considered, which is illustrated Figure 7.1.

The resulting state funnel projected onto x and y coordinates is given in Figure 7.1. The following trajectory and funnel cost functions are considered:

$$J_{\text{f}}(Q(t_0)) = w_{\text{f}} \cdot \text{trace}(Q_0^{-1}), \quad J_{\text{t}}(\bar{x}_k, \bar{u}_k) = \|\bar{u}\|_2^2,$$

where $w_{\text{f}} \in \mathbb{R}$ is the weight on the funnel cost. The objective of the funnel is to maximize the funnel entry and that of the trajectory is to minimize the input energy. The total time of simulation is set to $t_f = 5$ seconds, which is uniformly divided into $N = 9$ subintervals. The average Mosek's solve time per iteration 0.2 seconds.

Two results are compared: one with priority on the trajectory cost given in the top-left ($w_{\text{f}} = 100$) and one with priority on the funnel cost ($w_{\text{f}} = 10^4$) given in the top-right. It can be observed that when the funnel cost is prioritized, the trajectory introduces a curve

in its early segment to enlarge the funnel entry. In contrast, when the trajectory cost is prioritized, the path remains nearly straight in order to minimize input energy. The state funnel projected onto the velocity coordinates and the input funnel is given in Figure 7.2. We can observe that the generated funnels satisfy the constraint in every node point.

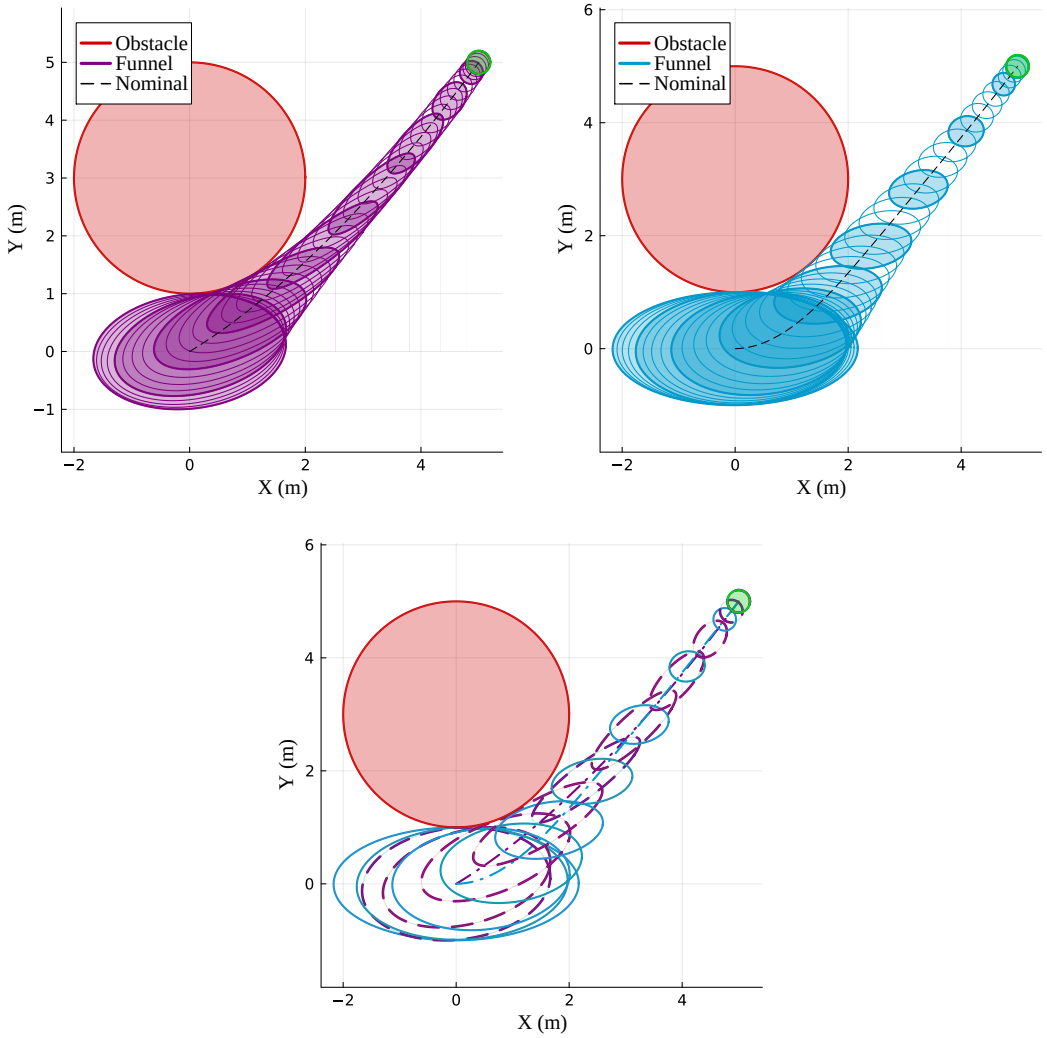

Figure 7.1: State funnels projected onto the x and y plane. Top-left: result with priority on the trajectory cost. Top-right: result with priority on the funnel cost. In top two figures, filled ellipsoids indicate funnels at discrete node points, while unfilled ellipsoids depict intermediate funnels between nodes. Bottom: Overlay of both computed funnels together.

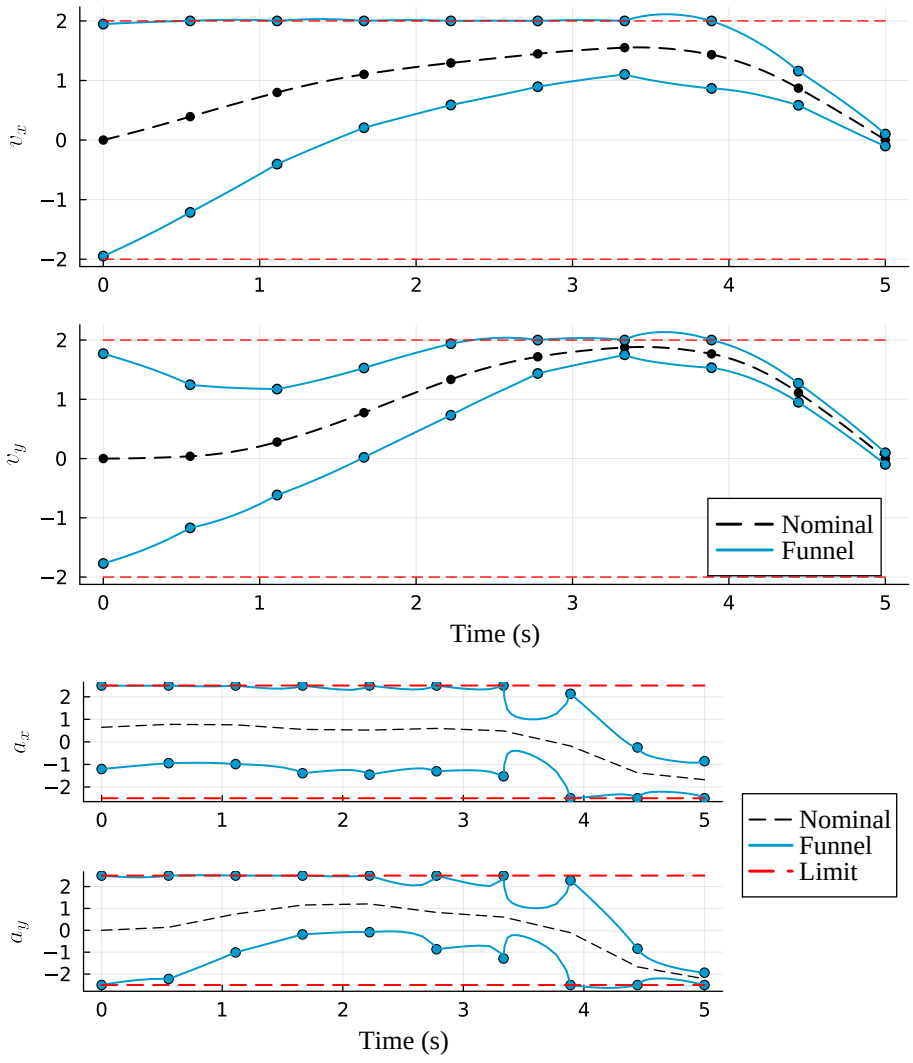

Figure 7.2: State funnels projected onto the v_x and v_y coordinates. Input funnels projected onto the a_x and a_y coordinates.

Chapter 8
CONCLUDING REMARKS

This thesis investigates robust predictive control for incrementally quadratic uncertain nonlinear systems. Two optimization-based approaches have been proposed: funnel synthesis and joint synthesis. Funnel synthesis constructs time-varying invariant sets and state feedback controllers around a given nominal trajectory to ensure robustness and constraint satisfaction of the closed-loop system. The joint synthesis algorithm integrates nominal trajectory generation and funnel computation into a unified framework, reducing conservatism.

Chapter 3 formulates a generalized funnel synthesis problem using incremental quadratic constraints (δQCs), enabling the treatment of broader nonlinearities such as sector-bounded and L-smooth dynamics that extend beyond the classical Lipschitz condition. The core of the formulation is a differential linear matrix inequality (DLMI) derived from a time-varying Lyapunov function, resulting in a convex optimization problem over time-varying Lyapunov matrices and state feedback gains. Chapters 4 and 5 develop solution strategies for this problem, with a particular focus on ensuring continuous-time constraint satisfaction (CTCS).

Three convex approaches are introduced to enforce CTCS in practice. The first approach introduces additional intermediate constraint-checking points, allowing the DLMI to be verified more tightly without increasing the number of decision variables. This method requires solving only a single semidefinite program (SDP) with additional LMI constraints, making it computationally attractive. The second approach formulates a sequential convexification scheme using subgradient-based updates to penalize violations of the DLMI at a finite set of sample points. While this method avoids increasing the number of LMIs, it requires solving multiple SDPs iteratively, offering direct control over inter-sample constraint violation. The third approach approximates the system matrices in such a way

that the DLMI can be exactly transformed into a finite number of LMIs. This leads to a non-iterative and checking-point-free method, but typically results in conservative funnel shapes due to the structural approximation. Each approach offers trade-offs between computational efficiency, accuracy, and conservatism, and their applicability may vary depending on the system and performance requirements.

Chapter 6 and Chapter 7 propose joint synthesis algorithms that compute both the nominal trajectory and the associated funnel in a unified optimization framework. This joint formulation reduces the conservatism that arises in conventional funnel synthesis approaches, where the nominal trajectory is assumed fixed and decoupled from the feedback design. By jointly optimizing the nominal trajectory and funnel, the algorithm balances trajectory cost and tracking performance (as reflected in the funnel size), while ensuring robustness to uncertainties and satisfying system constraints. In particular, the method developed in Chapter 7 employs a SCvx scheme based on the prox-linear method, thereby inheriting its theoretical convergence guarantees.

8.1 Future Directions

8.1.1 Customized SDP solvers for funnel synthesis

While the current work demonstrates that SDP formulations for funnel synthesis are computationally tractable, the reliance on general-purpose SDP solvers limits scalability for large-scale or high-dimensional systems. For instance, as discussed in Chapter 4, solving the SDP for a unicycle model with relatively small state and input dimensions takes only about 0.2 s, whereas the same procedure for a 6-DoF quadrotor requires over 20 s. This gap is still large even after accounting for differences in state and input dimensions as well as the number of node points, indicating that existing SDP solvers struggle with large-scale problems. This limitation becomes a significant barrier when applying joint synthesis methods, which require solving SDPs iteratively, to high-dimensional systems.

Future research should focus on tailored SDP solvers that exploit problem-specific structure such as block sparsity, low-rank matrix updates, and separability across time discretization. Custom implementations could integrate warm-start capabilities, exploit

parallel hardware architectures [27], and leverage interior-point [26, 52, 123] or first-order methods [44, 70, 137] to accelerate convergence. This would enable real-time or embedded deployment of funnel synthesis for high-frequency control loops.

8.1.2 Uncertainty quantification for multiplier matrices

For funnel computation, a valid set of multiplier matrices must be specified in order to employ δQCs. For instance, when Lipschitz (3.19) or L-smooth (3.20) conditions are considered, the corresponding constants need to be computed. In Chapter 4, these values were either derived analytically or estimated via sampling-based approaches. However, both approaches have limitations. The analytically derived constants are typically global values that hold uniformly over the entire domain, which often leads to overly conservative bounds. In contrast, sampling-based estimates can be less conservative but risk underestimation, thereby weakening the theoretical guarantees of funnel invariance.

8.1.3 From δQC to incremental IQC

The proposed funnel synthesis formulation laid out in Chapter 3 models the uncertainty using δQC. One way to extend the types of uncertainty the formulation can consider is to extend δQC to incremental integral quadratic constraints (δIQC). Importantly, every uncertainty or nonlinearity that satisfies a δQC also satisfies a δIQC, but the converse does not hold. Thus, the δIQC framework is strictly more general and can capture a wider range of dynamics and interconnection structures.

A major challenge, however, is that the Lyapunov condition (3.24) derived for δQC-based analysis no longer applies in the δIQC setting. To leverage the expressive power of δIQCs, new Lyapunov-type or dissipation-based conditions [133] must be developed to certify invariance. Future research should therefore focus on deriving such conditions and associated differential matrix inequality.

BIBLIOGRAPHY

[1] Ahmet Behçet Açıkmeşe. *Stabilization, observation, tracking and disturbance rejection for uncertain/nonlinear and time-varying systems*. PhD thesis, Purdue University, 2002.

[2] Ahmet Behçet Açıkmeşe and Martin Corless. Robust output tracking for uncertain/nonlinear systems subject to almost constant disturbances. *Automatica*, 38(11):1919–1926, 2002.

[3] Ahmet Behçet Açıkmeşe and Martin Corless. Robust tracking and disturbance rejection of bounded rate signals for uncertain/non-linear systems. *International Journal of Control*, 76(11):1129–1141, 2003.

[4] Behçet Açıkmeşe, John M. Carson III, and David S. Bayard. A robust model predictive control algorithm for incrementally conic uncertain/nonlinear systems. *International Journal of Robust and Nonlinear Control*, 21(5):563–590, 2011.

[5] Behçet Açıkmeşe and Martin Corless. Stability analysis with quadratic lyapunov functions: Some necessary and sufficient multiplier conditions. *Systems & Control Letters*, 57(1):78–94, 2008.

[6] Behçet Açıkmeşe and Martin Corless. Observers for systems with nonlinearities satisfying incremental quadratic constraints. *Automatica*, 47(7):1339–1348, 2011.

[7] Erling D Andersen and Knud D Andersen. The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High performance optimization*, pages 197–232. Springer, 2000.

[8] James Anderson, John C. Doyle, Steven H. Low, and Nikolai Matni. System level synthesis. *Annual Reviews in Control*, 47:364–393, 2019.

[9] D. Angeli. A lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002.

[10] Juan Carlos Arceo and Jimmy Lauber. Copositive conditions for lmi-based controller and observer design. *IFAC-PapersOnLine*, 53(2):7959–7964, 2020. 21st IFAC World Congress.

[11] V. Balakrishnan. Construction of lyapunov functions in robustness analysis with multipliers. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 3, pages 2021–2025 vol.3, 1994.

[12] V. Balakrishnan, Y. Huang, A. Packard, and J. Doyle. Linear matrix inequalities in analysis with multipliers. In *Proceedings of 1994 American Control Conference - ACC '94*, volume 2, pages 1228–1232 vol.2, 1994.

[13] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253, 2017.

[14] B. R. Barmish. Necessary and sufficient conditions for quadratic stabilizability of an uncertain system. *Journal of Optimization Theory and Applications*, 46(4):399–408, 1985.

[15] Maximilian Behr, Peter Benner, and Jan Heiland. Solution formulas for differential sylvester and lyapunov equations. *Calcolo*, 56(4):51, 2019.

[16] 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, 1984. 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984.

[17] Ross Boczar, Laurent Lessard, Andrew Packard, and Benjamin Recht. Exponential stability analysis via integral quadratic constraints. *arXiv preprint arXiv:1706.01337*, 2017.

[18] 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, 2019.

[19] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

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

[21] John Brewer. Kronecker products and matrix calculus in system theory. *IEEE Transactions on circuits and systems*, 25(9):772–781, 2003.

[22] Jyot Buch and Peter Seiler. Finite horizon robust synthesis using integral quadratic constraints. *International Journal of Robust and Nonlinear Control*, 31(8):3011–3035, 2021.

[23] R.L. Burden and J.D. Faires. *Numerical Analysis*. Brooks/Cole, Cengage Learning, 2011.

[24] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.

[25] 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, 2011.

[26] Govind M Chari and Behçet Açikmeşe. Qoco: A quadratic objective conic optimizer with custom solver generation. *arXiv preprint arXiv:2503.12658*, 2025.

[27] Govind M Chari, Abhinav G Kamath, Purnanand Elango, and Behçet Açikmeşe. Fast monte carlo analysis for 6-dof powered-descent guidance via gpu-accelerated sequential convex programming. In *AIAA SciTech 2024 Forum*, page 1762, 2024.

[28] Glen Chou, Necmiye Ozay, and Dmitry Berenson. Model error propagation via learned contraction metrics for safe feedback motion planning of unknown systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 3576–3583. IEEE, 2021.

[29] M. Corless and D. Da. *New Criteria for Robust Stability*, pages 329–337. Springer US, Boston, MA, 1989.

[30] Martin Corless. Control of uncertain nonlinear systems. *Journal of Dynamic Systems, Measurement, and Control*, 115(2B):362–372, 06 1993.

[31] Martin Corless. *Robust stability analysis and controller design with quadratic Lyapunov functions*, pages 181–203. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.

[32] Martin Corless and George Leitmann. Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. *IEEE Transactions on Automatic Control*, 26(5):1139–1144, 1981.

[33] Luis D'Alto and Martin Corless. Incremental quadratic stability. *Numerical Algebra, Control and Optimization*, 3(1):175–201, 2013.

[34] S. Di Cairano, H. Park, and I. Kolmanovsky. Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering. *International Journal of Robust and Nonlinear Control*, 22(12):1398–1427, 2012.

[35] Luca Dieci and Timo Eirola. Positive definiteness in the numerical solution of riccati differential equations. *Numerische Mathematik*, 67:303–313, 1994.

[36] Moritz Diehl and Sébastien Gros. Numerical optimal control. *Optimization in Engineering Center (OPTEC)*, 258, 2011.

[37] John Doyle. Analysis of feedback systems with structured uncertainties. In *IEE Proceedings D (Control Theory and Applications)*, volume 129, pages 242–250. IET, 1982.

[38] Dmitriy Drusvyatskiy and Adrian S Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018.

[39] Dmitriy Drusvyatskiy and Courtney Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178(1):503–558, 2019.

[40] 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.

[41] Jiří Fejlek and Stefan Ratschan. Computing funnels using numerical optimization based falsifiers. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4318–4324, 2022.

[42] Gowtham Garimella, Matthew Sheckells, Joseph L Moore, and Marin Kobilarov. Robust obstacle avoidance using tube nmpc. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[43] Paul J Goulart and Yuwen Chen. Clarabel: An interior-point solver for conic programs with quadratic objectives. *arXiv preprint arXiv:2405.12762*, 2024.

[44] Dennis Gramlich, Tobias Holicki, Carsten W. Scherer, and Christian Ebenbauer. A structure exploiting sdp solver for robust controller synthesis. *IEEE Control Systems Letters*, 7:1831–1836, 2023.

[45] Allal Guessab. Approximations of differentiable convex functions on arbitrary convex polytopes. *Applied Mathematics and Computation*, 240:326–338, 2014.

[46] Navid Hashemi, Justin Ruths, and Mahyar Fazlyab. Certifying incremental quadratic constraints for neural networks via convex optimization. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pages 842–853. PMLR, 07 – 08 June 2021.

[47] Bin Hu and Peter Seiler. Exponential decay rate conditions for uncertain linear systems using integral quadratic constraints. *IEEE Transactions on Automatic Control*, 61(11):3631–3637, 2016.

[48] Alberto Isidori. *Nonlinear control systems: an introduction*. Springer, 1985.

[49] Inkyu Jang, Hoseong Seo, and H. Jin Kim. Fast computation of tight funnels for piecewise polynomial systems. *IEEE Control Systems Letters*, 6:2234–2239, 2022.

[50] R. E. Kalman and J. E. Bertram. Control system analysis and design via the "second method" of lyapunov: I—continuous-time systems. *Journal of Basic Engineering*, 82(2):371–393, 06 1960.

[51] Abhinav G Kamath, Purnanand Elango, Taewan Kim, Skye Mceowen, Yue Yu, John M Carson, 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*, page 2003, 2023.

[52] Chung-Yao Kao and Alexandre Megretski. On the new barrier function and specialized algorithms for a class of semidefinite programs. *SIAM Journal on Control and Optimization*, 46(2):468–495, 2007.

[53] Tosio Kato. *Perturbation theory for linear operators*, volume 132. Springer Science & Business Media, 2013.

[54] H.K. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002.

[55] P.P. Khargonekar, I.R. Petersen, and K. Zhou. Robust stabilization of uncertain linear systems: quadratic stabilizability and h/sup infinity / control theory. *IEEE Transactions on Automatic Control*, 35(3):356–361, 1990.

[56] 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 (IEEE Cat. No. CH37301)*, volume 5, pages 3576–3581. IEEE, 2002.

[57] Taewan Kim, Purnanand Elango, and Behçet Açıkmeşe. Joint synthesis of trajectory and controlled invariant funnel for discrete-time systems with locally lipschitz nonlinearities. *International Journal of Robust and Nonlinear Control*, 34(6):4157–4176, 2024.

[58] Taewan Kim, Purnanand Elango, Danylo Malyuta, and Behçet Açıkmeşe. Guided policy search using sequential convex programming for initialization of trajectory optimization algorithms. In *2022 American Control Conference (ACC)*, pages 3572–3578, 2022.

[59] Taewan Kim, Purnanand Elango, Taylor P. Reynolds, Behçet Açıkmeşe, and Mehran Mesbahi. Optimization-based constrained funnel synthesis for systems with lipschitz nonlinearities via numerical optimal control. *IEEE Control Systems Letters*, 7:2875–2880, 2023.

[60] Taewan Kim, Abhinav G Kamath, Niyousha Rahimi, Jasper Corleis, Behçet Açıkmeşe, and Mehran Mesbahi. Six-degree-of-freedom aircraft landing trajectory planning with runway alignment. *arXiv preprint arXiv:2405.16680*, 2024.

[61] Taewan Kim, Wonchul Kim, Seungwon Choi, and H Jin Kim. Path tracking for a skid-steer vehicle using model predictive control with on-line sparse gaussian process. *IFAC-PapersOnLine*, 50(1):5755–5760, 2017.

[62] Jonas Koenemann, Andrea Del Prete, Yuval Tassa, Emanuel Todorov, Olivier Stasse, Maren Bennewitz, and Nicolas Mansard. Whole-body model-predictive control applied to the hrp-2 humanoid. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3346–3351. IEEE, 2015.

[63] Johannes Köhler, Raffaele Soloperto, Matthias A Müller, and Frank Allgöwer. A computationally efficient robust model predictive control framework for uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 66(2):794–801, 2020.

[64] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–383. IEEE, 2016.

[65] W. Langson, I. Chryssochoos, S.V. Raković, and D.Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.

[66] Antoine P. Leeman, Johannes Köhler, Andrea Zanelli, Samir Bennani, and Melanie N. Zeilinger. Robust nonlinear optimal control via system level synthesis. *IEEE Transactions on Automatic Control*, 70(7):4780–4787, 2025.

[67] Antoine P. Leeman, Jérôme Sieber, Samir Bennani, and Melanie N. Zeilinger. Robust optimal control for nonlinear systems with parametric uncertainties via system level synthesis. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 4784–4791, 2023.

[68] G. Leitmann. Guaranteed asymptotic stability for some linear systems with bounded uncertainties. *Journal of Dynamic Systems, Measurement, and Control*, 101(3):212–216, 09 1979.

[69] Thomas Lew, Riccardo Bonalli, and Marco Pavone. Chance-constrained sequential convex programming for robust trajectory optimization. In *2020 European Control Conference (ECC)*, pages 1871–1878. IEEE, 2020.

[70] Zishuo Li, Bo Yang, Jiayun Li, Jiaqi Yan, and Yilin Mol. Linear model predictive control under continuous path constraints via parallelized primal-dual hybrid gradient algorithm. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 159–164. IEEE, 2023.

[71] A. Liapounoff. Problème général de la stabilité du mouvement. *Annales de la Faculté des sciences de l'Université de Toulouse pour les sciences mathématiques et les sciences physiques*, 2e série, 9:203–474, 1907.

[72] 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.

[73] Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[74] Anirudha Majumdar and Russ Tedrake. Funnel libraries for real-time robust feedback motion planning. *The International Journal of Robotics Research*, 36(8):947–982, 2017.

[75] Aleksandr Ivanovich Malikov and Diana Igorevna Dubakina. Numerical methods for solving optimization problems with differential linear matrix inequalities. *Russian Mathematics*, 64:64–74, 2020.

[76] Danylo Malyuta. Real-time optimization beyond convexity: Provable methods for aerospace control systems. Technical report, University of Washington, 2020. General Exam Report, unpublished.

[77] 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.

[78] Zachary Manchester and Scott Kuindersma. Robust direct trajectory optimization using approximate invariant funnels. *Autonomous Robots*, 43(2):375–387, 2019.

[79] Yuanqi Mao, Daniel Dueri, Michael Szmuk, and Behçet Açıkmeşe. Successive convexification of non-convex optimal control problems with state constraints. *Ifac-PapersOnline*, 50(1):4063–4069, 2017.

[80] 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, 2016.

[81] Yuanqi Mao, Michael Szmuk, Xiangru Xu, and Behçet Açıkmeşe. Successive convexification: A superlinearly convergent algorithm for non-convex optimal control problems. *arXiv preprint arXiv:1804.06539*, 2018.

[82] M. Mason. The mechanics of manipulation. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 544–548, 1985.

[83] D.Q. Mayne, M.M. Seron, and S.V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[84] Skye Mceowen, Daniel J Calderone, Aman Tiwary, Jason S Zhou, Taewan Kim, Purnanand Elango, and Behçet Açıkmeşe. Auto-tuned primal-dual successive convexification for hypersonic reentry guidance. In *AIAA SCITECH 2025 Forum*, page 1317, 2025.

[85] A. Megretski. Power distribution approach in robust control. *IFAC Proceedings Volumes*, 26(2, Part 2):27–30, 1993. 12th Triennal Wold Congress of the International Federation of Automatic control. Volume 2 Robust Control, Design and Software, Sydney, Australia, 18-23 July.

[86] Alexandre Megretski and Anders Rantzer. System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42(6):819–830, 2002.

[87] Florian Messerer and Moritz Diehl. An efficient algorithm for tube-based robust nonlinear optimal control with optimal linear feedback. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6714–6721, 2021.

[88] Lazaros Moysis, Mahendra Kumar Gupta, Vikas Mishra, Muhammad Marwan, and Christos Volos. Observer design for rectangular descriptor systems with incremental quadratic constraints and nonlinear outputs—application to secure communications. *International Journal of Robust and Nonlinear Control*, 30(18):8139–8158, 2020.

[89] Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[90] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

[91] Ricardo CLF Oliveira and Pedro LD Peres. Stability of polytopes of matrices via affine parameter-dependent Lyapunov functions: Asymptotically exact LMI conditions. *Linear algebra and its applications*, 405:209–228, 2005.

[92] JM Park, DW Kim, YS Yoon, HJ Kim, and KS Yi. Obstacle avoidance of autonomous vehicles based on model predictive control. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 223(12):1499–1516, 2009.

[93] P.A. Parrilo. Semidefinite programming based tests for matrix copositivity. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, volume 5, pages 4624–4629 vol.5, 2000.

[94] S. Prajna, P.A. Parrilo, and A. Rantzer. Nonlinear control synthesis by convex optimization. *IEEE Transactions on Automatic Control*, 49(2):310–314, 2004.

[95] S.V. Raković, E.C. Kerrigan, K.I. Kouramas, and D.Q. Mayne. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3):406–410, 2005.

[96] Taylor Reynolds, Danylo Malyuta, Mehran Mesbahi, Behçet Açıkmeşe, and John M Carson. A real-time algorithm for non-convex powered descent guidance. In *AIAA Scitech 2020 Forum*, page 0844, 2020.

[97] Taylor Reynolds, Danylo Malyuta, Mehran Mesbahi, Behçet Açıkmeşe, and John M. Carson. Funnel synthesis for the 6-dof powered descent guidance problem. *AIAA Scitech 2021 Forum*, 2021.

[98] 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, 2020.

[99] Taylor Patrick Reynolds. *Computational Guidance and Control for Aerospace Systems*. University of Washington, 2020.

[100] Taylor Patrick Reynolds, Danylo Malyuta, Mehran Mesbahi, and Behçet Açıkmeşe. Temporally-interpolated funnel synthesis for nonlinear systems. In *2nd RSS Workshop on Robust Autonomy: Tools for Safety in Real-World Uncertain Environments (RSS 2020)*, 2020.

[101] Taylor Patrick Reynolds, Danylo Malyuta, Mehran Mesbahi, and Behçet Açıkmeşe. Temporally-interpolated funnel synthesis for nonlinear systems. In *2nd RSS Workshop on Robust Autonomy: Tools for Safety in Real-World Uncertain Environments (RSS 2020)*, 2020.

[102] Arthur Richards and Jonathan P. How. Robust variable horizon model predictive control for vehicle maneuvering. *International Journal of Robust and Nonlinear Control*, 16(7):333–351, 2006.

[103] R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*. Springer, 1st edition, 1998.

[104] Nicolas Rouche, Patrick Habets, and Michel Laloy. *Stability theory by Liapunov's direct method*, volume 4. Springer, 1977.

[105] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013.

[106] Carsten W Scherer. Dissipativity and integral quadratic constraints: Tailored computational robustness tests for complex interconnections. *IEEE Control Systems Magazine*, 42(3):115–139, 2022.

[107] Peter Seiler. Stability analysis with dissipation inequalities and integral quadratic constraints. *IEEE Transactions on Automatic Control*, 60(6):1704–1709, 2015.

[108] Peter Seiler, Robert M. Moore, Chris Meissen, Murat Arcak, and Andrew Packard. Finite horizon robustness analysis of ltv systems using integral quadratic constraints. *Automatica*, 100:135–143, 2019.

[109] Peter Seiler, Andrew Packard, and Gary J Balas. A dissipation inequality formulation for stability analysis with integral quadratic constraints. In *49th IEEE Conference on Decision and Control (CDC)*, pages 2304–2309. IEEE, 2010.

[110] Peter Seiler and Raghu Venkataraman. Trajectory-based robustness analysis for nonlinear systems. *International Journal of Robust and Nonlinear Control*, 34(2):910–926, 2024.

[111] Hoseong Seo, Donggun Lee, Clark Youngdong Son, Inkyu Jang, Claire J. Tomlin, and H. Jin Kim. Real-time robust receding horizon planning using hamilton–jacobi reachability analysis. *IEEE Transactions on Robotics*, 39(1):90–109, 2023.

[112] Hoseong Seo, Clark Youngdong Son, and H. Jin Kim. Fast funnel computation using multivariate bernstein polynomial. *IEEE Robotics and Automation Letters*, 6(2):1351–1358, 2021.

[113] Hoseong Seo, Clark Youngdong Son, and H. Jin Kim. Fast funnel computation using multivariate bernstein polynomial. *IEEE Robotics and Automation Letters*, 6(2):1351–1358, 2021.

[114] Jongho Shin and H Jin Kim. Nonlinear model predictive formation flight. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(5):1116–1125, 2009.

[115] Eduardo D Sontag and Yuan Wang. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24(5):351–359, 1995.

[116] 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*, 2018.

[117] 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, 2020.

[118] 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.

[119] 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.

[120] Russ Tedrake, Ian R. Manchester, Mark Tobenkin, and John W. Roberts. Lqr-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8):1038–1052, 2010.

[121] Mark M. Tobenkin, Ian R. Manchester, and Russ Tedrake. Invariant funnels around trajectories using sum-of-squares programming. *IFAC Proceedings Volumes*, 44(1):9218–9223, 2011. 18th IFAC World Congress.

[122] Ufuk Topcu, Andrew Packard, and Peter Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44(10):2669–2675, 2008.

[123] Lieven Vandenberghe, V. Ragu Balakrishnan, Ragnar Wallin, Anders Hansson, and Tae Roh. *Interior-Point Algorithms for Semidefinite Programming Problems Derived from the KYP Lemma*, pages 195–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[124] Mario E. Villanueva, Rien Quirynen, Moritz Diehl, Benoît Chachuat, and Boris Houska. Robust mpc via min–max differential inequalities. *Automatica*, 77:311–321, 2017.

[125] Jan C Willems. Dissipative dynamical systems part i: General theory. *Archive for rational mechanics and analysis*, 45(5):321–351, 1972.

[126] Jan C Willems. Dissipative dynamical systems part ii: Linear systems with quadratic supply rates. *Archive for rational mechanics and analysis*, 45(5):352–393, 1972.

[127] Xiangru Xu, Behçet Açikmeşe, and Martin J. Corless. Observer-based controllers for incrementally quadratic nonlinear systems with disturbances. *IEEE Transactions on Automatic Control*, 66(3):1129–1143, 2021.

[128] V. A. Yakubovich. The solution of some matrix inequalities encountered in automatic control theory. *Dokl. Akad. Nauk SSSR*, 143(6):1304–1307, 1962.

[129] V. A. Yakubovich. Solution of certain matrix inequalities encountered in non-linear control theory. *Dokl. Akad. Nauk SSSR*, 156(2):278–281, 1964.

[130] V. A. Yakubovich. The method of matrix inequalities in the stability theory of nonlinear control systems. ii. absolute stability in a class of nonlinearities with a condition on the derivative. *Avtomat. i Telemekh.*, 26(4):577–590, 1965. English transl.: *Autom. Remote Control*, **26** (1965), 577–592.

[131] V. A. Yakubovich. Frequency conditions for the absolute stability of control systems with several nonlinear or linear nonstationary blocks. *Avtomat. i Telemekh.*, 6:5–30, 1967.

[132] V. A. Yakubovich. The s-procedure in non-linear control theory. *Vestnik Leningrad Univ. Mathe.*, 4:73–93, 1977.

[133] He Yin, Peter Seiler, and Murat Arcak. Backward reachability using integral quadratic constraints for uncertain nonlinear systems. *IEEE Control Systems Letters*, 5(2):707–712, 2021.

[134] Yongsoon Yoon, Jongho Shin, H Jin Kim, Yongwoon Park, and Shankar Sastry. Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. *Control Engineering Practice*, 17(7):741–750, 2009.

[135] Shuyou Yu, Hong Chen, and Frank Allgöwer. Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2650–2655, 2011.

[136] Shuyou Yu, Christoph Maier, Hong Chen, and Frank Allgöwer. Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems. *Systems & Control Letters*, 62(2):194–200, 2013.

[137] Yue Yu, Purnanand Elango, Ufuk Topcu, and Behçet Açıkmeşe. Proportional–integral projected gradient method for conic optimization. *Automatica*, 142:110359, 2022.

[138] Kemin Zhou and P. Khargonekar. Stability robustness bounds for linear state-space models with structured uncertainty. *IEEE Transactions on Automatic Control*, 32(7):621–623, 1987.

Appendix A

ADDITIONAL LEMMAS AND PROOFS

Lemma A.1. *Let* $\mathcal{E}_\eta = \{\eta \in \mathbb{R}^{n_x} \mid \eta^\top Q^{-1}\eta \leq 1\}$ *with* $Q \in \mathbb{S}^{n_x}_{++}$*, and consider* $K \in \mathbb{R}^{n_u \times n_x}$*. Define*

$$\mathcal{E}_\xi = \Big\{(KQK^\top)^{1/2}y \;\Big|\; \|y\|_2 \leq 1\Big\}.$$

Then the condition $\eta \in \mathcal{E}_\eta$ *is equivalent to* $K\eta \in \mathcal{E}_\xi$*.*

Proof. We can equivalently re-write $\mathcal{E}_\eta$ as

$$\mathcal{E}_\eta = \{Q^{1/2}z \mid \|z\|_2 \leq 1\},$$

since Q is PD, so invertible. So applying K to it gives a condition

$$Kx \in \{KQ^{1/2}y \mid \|y\|_2 \leq 1\}.$$

Take the full singular value decomposition of $KQ^{1/2}$ as $KQ^{1/2} = U\Sigma V^\top$ where U and V are orthogonal and Σ is diagonal. Note that $KQK^\top = U\Sigma^2U^\top$, so $(KQK^\top)^{1/2} = U\Sigma U^\top$. We have

$$\begin{aligned}\{KQ^{1/2}y \mid \|y\|_2 \leq 1\} &= \{U\Sigma V^\top y \mid \|y\|_2 \leq 1\},\\ &= \{U\Sigma z \mid \|z\|_2 \leq 1\},\\ &= \{(KQK^\top)^{1/2}y \Big| \|y\|_2 \leq 1\},\end{aligned}$$

where we used that $V^\top$and $U^\top$ are orthogonal and invertible, so they preserve the unit ball. □