공학박사학위논문

Real-time Safety-guaranteed Trajectory Planning using Hamilton-Jacobi Reachability Analysis

도달 가능성 분석을 통한 안정성이 보장된 실시간 경로계획 기법

2021년 2월

서울대학교 대학원

기계항공공학부

서호성

Real-time Safety-guaranteed Trajectory Planning using Hamilton-Jacobi Reachability Analysis

도달 가능성 분석을 통한 안정성이 보장된 실시간 경로계획 기법

지도교수 김 현 진

이 논문을 공학박사 학위논문으로 제출함

2020년 11월

서울대학교 대학원

기계항공공학부

서 호 성

서호성의 공학박사 학위논문을 인준함

2020년 12월

위 원 장 : 김 유 단

부위원장 : 김 현 진

위　　원 : 심 형 보

위　　원 : 김 수 성

위　　원 : 이 현 범

李賢範印

Real-time Safety-guaranteed Trajectory Planning using Hamilton-Jacobi Reachability Analysis

A Dissertation

by

HOSEONG SEO

Presented to the Faculty of the Graduate School of

Seoul National University

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical and Aerospace Engineering

Seoul National University

Supervisor : Professor H. Jin Kim

FEBURARY 2021

Real-time Safety-guaranteed Trajectory Planning using Hamilton-Jacobi Reachability Analysis

HOSEONG SEO

Department of Mechanical and Aerospace Engineering

Seoul National University

APPROVED:

Youdan Kim
____

Youdan Kim, Chair, Ph.D.

Hjin Kim
____

H. Jin Kim, Ph.D.

____

Hyungbo Shim, Ph.D.

____

Suseong Kim, Ph.D.

____

Hyeonbeom Lee, Ph.D.

to my

MOTHER, FATHER, BROTHER, and WIFE

with love

Abstract

Real-time Safety-guaranteed Trajectory Planning using Hamilton-Jacobi Reachability Analysis

Hoseong Seo
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

Guaranteeing safety in planning trajectories of robotic systems is a fundamental requirement for reliable operation. However, it is difficult to verify in advance whether the seemingly-safe planned trajectory will be safe because of unknown disturbances exerted to the system during the actual runtime. This dissertation presents a real-time safety-guaranteed trajectory planning algorithm.

The safety of the system can be guaranteed by the forward reachable sets (FRSs) computed from the Hamilton-Jacobi (HJ) reachability analysis. Considering that the unexpected disturbance can instantaneously drive the system to devastating situations, this work concentrates on real-time safety verification. The proposed reachability analysis consists of 2 stages: the analysis of the linearized (time-varying) dynamics, and nonlinear dynamics that is polynomial of states and disturbances. For the reachability analysis on linear time-varying (LTV) systems, this work presents an ellipsoidal approximation of the FRS based on the generalized Hopf formula. Since the formula gives the explicit solution of the Hamilton-Jacobi-Bellman (HJB) equation, the proposed ellipsoidal approximation encloses the FRS of LTV systems. To incorporate the nonlinearities that are neglected in the derivation of the FRS of LTV systems, this work further proposes an algorithm for computing the funnel by approximating the value function of the HJB equation as a multivariate polynomial of states. From the property of the Bernstein polynomial, the conservativeness condition of the value approximation is converted as linear inequality constraints without any compromise. The proposed funnel computation algorithm consists of multiple linear programs (LPs), and thus computationally much tractable compared to most of the existing works.

Finally, a constrained optimization problem is formulated to generate the reference trajectory of the system. The safety of the system is considered as nonlinear constraints of the optimization such that the funnel and unsafe regions do not intersect. Since the funnel certainly contains the FRS, the system is guaranteed to be safe regardless of disturbances. Also, the fast computation of the funnel allows the optimization problem to be solved in real-time. Consequently, the proposed method enables real-time replanning of the trajectory with safety guarantees even when the system encounters unexpected disturbances in runtime. Flight experiments of obstacle avoidance in a windy environment present the validity of the proposed planning algorithm.

Keywords: Robot safety, Motion and path planning, Optimization and optimal control, Reachability analysis.

Student Number: 2016-30185

Table of Contents

Page

Abstract . . . vi

Table of Contents . . . viii

List of Tables . . . x

List of Figures . . . xi

Chapter

1 Introduction . . . 1

1.1 Literature Survey . . . 6

1.2 Contributions . . . 10

1.3 Outline . . . 11

2 Background . . . 12

2.1 Forward Reachable Set and Funnel . . . 12

2.2 Hamilton-Jacobi Reachability Analysis . . . 14

2.3 The Generalized Hopf Formula . . . 15

3 Reachability Analysis of LTV Systems . . . 18

3.1 Characterization of Forward Reachable Set . . . 18

3.2 Ellipsoidal Approximation of Sets due to Disturbances . . . 23

3.3 Minimum Volume Composition of Ellipsoids . . . 28

3.4 Ellipsoidal Approximation of FRS of LTV Systems . . . 30

4 Reachability Analysis of Nonlinear Systems . . . 37

4.1 Minimal Funnel . . . 37

4.2 Multivariate Bernstein Polynomial . . . 39

4.3 Approximation of Hamiltonian . . . 44

4.4 Funnel Computation . . . 46

4.5 Simulation Results and Discussion 52

5 Safety-guaranteed Trajectory Planning 65

5.1 Problem Statement 65

5.2 Global Reference Planning 67

5.3 Local Replanning with Safety Guarantee 68

5.4 Simulation Results and Discussion 71

6 Experimental Validation 78

6.1 Scenario 78

6.2 Setups 80

6.3 Results and Discussion 81

7 Conclusion 88

Appendix

Abstract (*in Korean*) 98

List of Tables

4.1 Comparison of the funnel computation time 53

5.1 Parameters used in the simulation 71

6.1 Parameters used in the experiment 82

6.2 Mean computation time of the trajectory planning 85

List of Figures

1.1 Illustration of reachability-based planning 3

1.2 Illustration of safety-guaranteed replanning 4

1.3 Schematic diagram of the proposed planning pipeline 5

3.1 FRS of a state-independent 2-state LTV system 22

3.2 Ellipsoidal approximation of the sets due to disturbances 27

3.3 Convergence characteristics of the minimum volume composition algorithm 31

3.4 FRS and its approximation for a 2-state LTV system 35

3.5 Distribution of α in the minimum volume composition of ellipsoids 36

4.1 Comparison of the funnels of the Lotka-Volterra type system 54

4.2 Comparison of the funnels of the Lotka-Volterra type system computed by quadratic and polynomial approximations 56

4.3 Comparison of the funnels of the Lotka-Volterra type system computed by different orders of polynomials 57

4.4 Tradeoff between the computation time and the conservativeness of the funnel 58

4.5 Comparison of the funnels of the unicycle computed by the proposed and SOS program-based methods 60

4.6 Comparison of the funnels of the unicycle and the worst-case state trajectories 61

4.7 Comparison of the funnels of the multirotor and the worst-case state trajectories 64

5.1 Procedure of the proposed trajectory planning 70

5.2 Simulation results of the proposed trajectory optimization 75

5.3 Comparison of the planner with and without the backward reachability analysis 77

6.1 Experiment environment 79

6.2 Snapshot of the experiment 79

6.3 Platform used in the experiment 80

6.4 Structure of modules used in the experiment 82

6.5 Experimental results of the safety-guaranteed replanning 84

6.6 Deviation of states and the corresponding bound 86

6.7 Experimental results with and without updating disturbance bound 87

1

Introduction

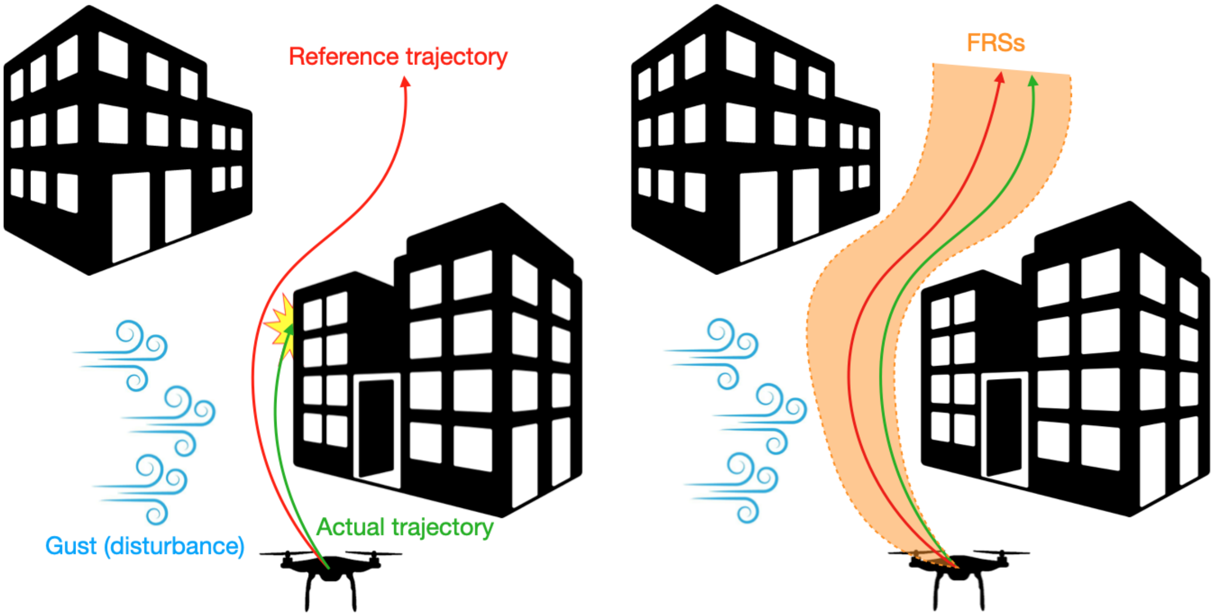
Guaranteeing safety is a core element for exploiting a wide range of robotic systems. Planning an appropriate reference trajectory for the system is necessary to avoid risky regions and accomplish a given mission. Even though the planned trajectory is ostensibly safe, it may lead to hazardous situations due to inevitable disturbances in real environments. To verify the runtime safety of the system, the planner must take into account how far the system strays from the nominal trajectory (i.e. the disturbance-free state trajectory) caused by disturbances. In terms of guaranteeing safety, a funnel, the set of states which can be reached from an initial set of states under adversarial disturbances, can be used.

Evaluating the funnel of the system is computationally burdensome because disturbances in runtime are usually unknown, and the nonlinear dynamics of the system make the evaluation complicated. To overcome this computational issue, methods for the online composition of funnels computed in the offline step are proposed. These funnels are calculated for the predefined (expected) bounded set of disturbances, and then, in the online phase, the funnel suitable for the current disturbance condition is loaded from the precomputed funnels. However, the actual disturbances may exceed the predefined disturbance range. In this case, the previously computed funnels no longer guarantee the safety of the system. To ensure the safety of the system under unexpected disturbances, it is necessary to quickly compute the funnel, considering the additional disturbances in runtime.

The objective of this dissertation is to provide a safe reference trajectory for a vehicle so that the vehicle reaches the goal while avoiding unsafe regions, regardless of disturbances. If the vehicle following a predefined reference trajectory comes across previously unknown obstacles, the vehicle should modify its local maneuver plan to avoid possible collisions. Furthermore, when additional disturbances suddenly interrupt the vehicle, the evasive maneuver must take into account such unexpected disturbances to safely bypass the obstacle. Keeping this circumstance in mind, this work aims for the safety-guaranteed trajectory planning in the following two perspectives. First, the worst-case scenario, rather than a probabilistic perspective, is analyzed to achieve the deterministic safety guarantee as described in Fig. 1.1. Second, the local reference replanning as well as the worst-case analysis must be performed in real-time to cope with sudden changes in the environment during runtime as illustrated in Fig. 1.2. As a means of guaranteeing safety, this work utilizes an outer approximation of the forward reachable set (FRS) of the system. The FRS is the set of states driven by the disturbances from an initial set, where the initial set is the set near the current state. We then solve a trajectory optimization problem to modify the reference trajectory considering the derived FRS and updated environmental information. The planned reference trajectory eventually guarantees the safety of the vehicle by making sure that the corresponding FRS does not intersect with unsafe regions.

This dissertation presents an algorithm for computing a funnel, a tight outer approximation of the FRS of nonlinear systems. To compute the funnel of the nonlinear system with computational efficiency, the FRS of the linearized LTV system is analyzed first. The proposed computation of the FRS of LTV systems derives a sizable computational benefit from the generalized Hopf formula [1]. From the formula, it is shown that the FRS of LTV systems is the Minkowski sum of sets: the initial set and the sets propagated by disturbances. To reduce the computation burden regarding the Minkowski sum of sets, the sets propagated by disturbances are approximated as outer ellipsoids. Accordingly, the FRS of LTV systems is also over-approximated as the Minkowski sum of the initial set and the approximated ellipsoids for the sets due to disturbances.

Then, the funnel is computed by approximating the value function of the Hamilton-Jacobi-Bellman (HJB) equation as a multivariate polynomial of states. The value function is conservatively approximated so that the funnel certainly contains the FRS of nonlinear systems. The conservativeness condition is shown to be inequality constraints between polynomials. By using the property of the Bernstein polynomials [2], the polynomial inequality constraints can be expressed as linear inequality constraints of the coefficients of the approximated (polynomial) value function. Based on the derived linear inequality constraints, sequential linear programming (SLP) is formulated to find the optimal coefficients of the polynomial value function. The ellipsoidal approximation of the FRS of the LTV system is



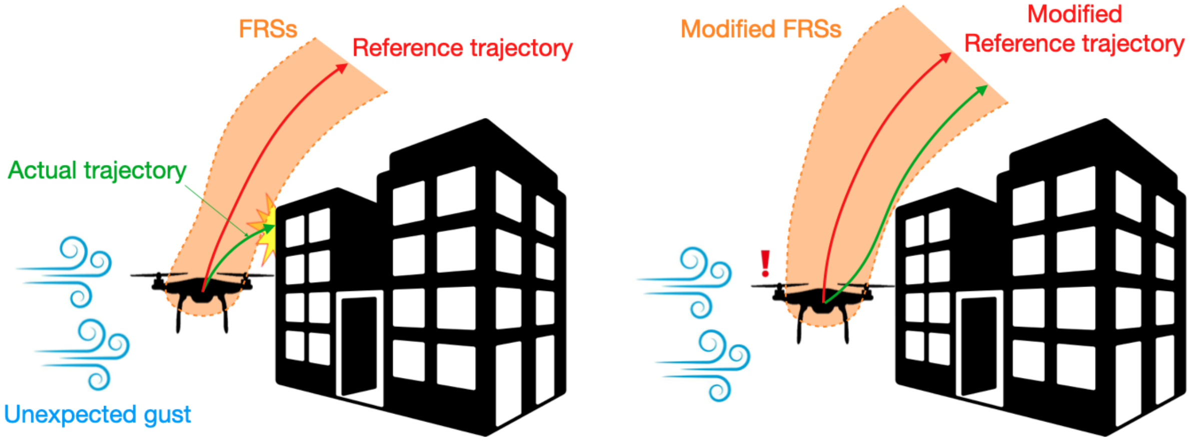
(a) Planning without considering disturbances

(b) Planning with considering disturbances

Figure 1.1: Illustration of reachability-based planning. (a) If a multirotor follows a seemingly collision-free reference trajectory (the solid red line), the multirotor may collide with obstacles (the solid green line) due to disturbances during flight. (b) By considering the FRS (the shaded orange region) in planning the reference trajectory, the multirotor can safely bypass obstacles.

used as the initial guess of the SLP. The proposed funnel computation consists of multiple LPs, and thus it enables a tractable computation of the funnel.

After computing the funnel of nonlinear systems, the optimal reference trajectory is generated as follows. Given a preplanned global reference trajectory used as the initial guess of the local reference trajectory, a two-step approach is proposed to optimize the local reference, as shown in Fig. 1.3. In the first step shaded in blue, the funnel of the nonlinear system is computed along with the current reference. In the second step shaded in red, constrained optimization is performed so that the funnel computed in the first step does not intersect with the unsafe regions. After updating the current reference with the optimal one computed in the second step, we reevaluate the funnel and repeat those processes until the computation of the reference trajectory converges. Finally, the converged reference trajectory guarantees safety while the system follows the corresponding reference.



(a) The multirotor disturbed by unexpected gust

(b) Guaranteeing safety by replanning

Figure 1.2: Illustration of safety-guaranteed replanning of the reference trajectory. (a) Additional disturbances during runtime degrade the safety guarantee of the reference trajectory (the solid red line) since the precomputed FRS (the shaded orange region) cannot capture the unexpected deviation (the solid green line) due to the additional disturbances. (b) By rapidly modifying the FRS considering the additional disturbances, the corresponding reference trajectory can be replanned so that the multirotor is guaranteed to be collision-free regardless of the unexpected disturbances.

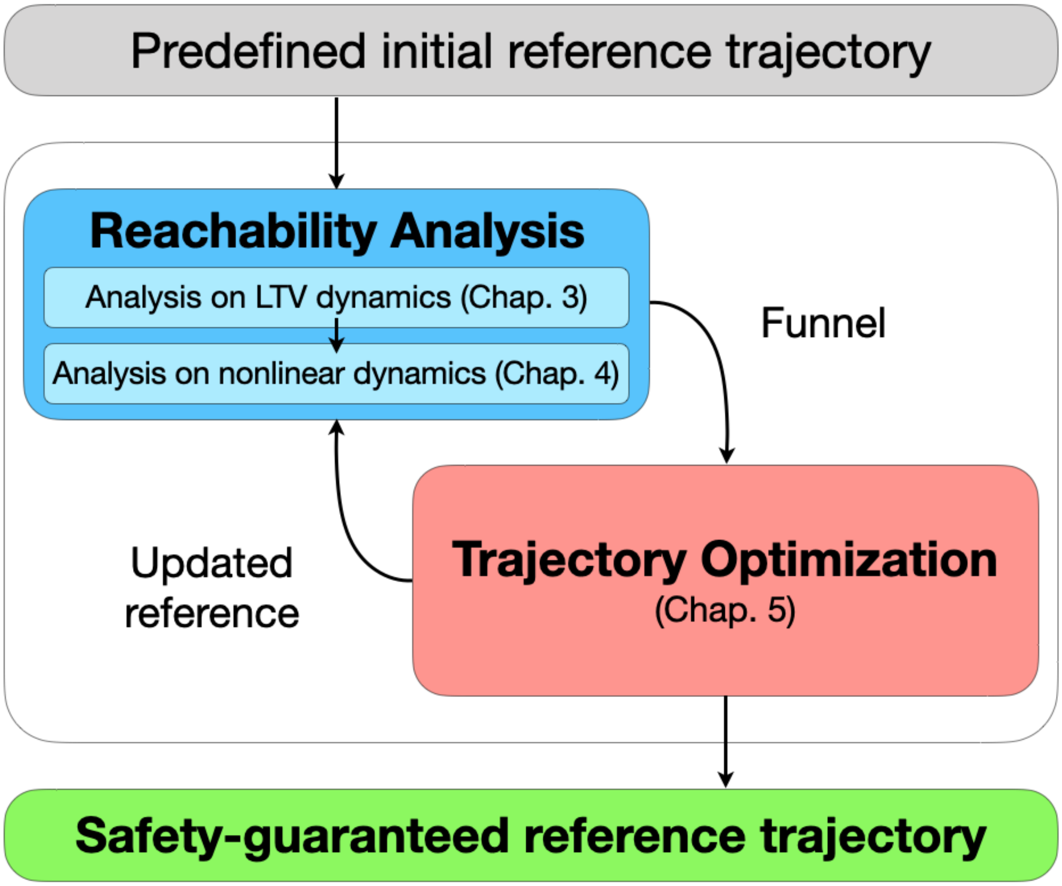

Predefined initial reference trajectory
Reachability Analysis
Analysis on LTV dynamics (Chap. 3)
Analysis on nonlinear dynamics (Chap. 4)
Funnel
Updated reference
Trajectory Optimization
(Chap. 5)
Safety-guaranteed reference trajectory

Figure 1.3: Schematic diagram of the proposed planning pipeline

1.1 Literature Survey

There exists an extensive body of literature in safe trajectory planning against bounded disturbance. Originally, the reachable set of the system can be obtained by solving the corresponding HJB equation [3]. Since it is difficult to compute the analytic solution of the HJB equation, some works utilize a numerical method to solve the HJB equation and evaluate the reachable set [4, 5, 6]. Other works parameterize the solution of the HJB equation, i.e. the value function, as a polynomial of states. Then, an optimization problem is formulated to find the best approximation of the value function [7, 8, 9, 10, 11]. Works in other category represent the reachable set as a semi-algebraic set, and develop the propagation law of the corresponding geometry [12, 13, 14]. The most closely related works to the objective of this study are categorized as the following sections.

1.1.1 Hamilton-Jacobi Reachability Analysis

Hamilton-Jacobi (HJ) reachability analysis [15] is a method to derive the FRS through solving a corresponding HJB equation. However, solving HJB equations by gridding the state space leads to the exponential computational complexity in the dimension of the state, and this has limited the use of the HJ reachability analysis in real-time planning. To alleviate this exponential complexity, the Lax and Hopf theory [16, 17] aims to provide an analytic solution to the HJB equation. Until very recently, state-dependent dynamics could not have been addressed. The state-of-the-art works in such efforts include treatment of a linear state-dependent dynamics [1, 18], or a linearization-based approach for a nonlinear dynamics [19]. Such approaches can only determine whether a particular state is reachable or not and do not provide the reachable set itself. It is not possible to guarantee the safety of all states starting from the initial set by judging whether the particular state is reachable or not. The proposed approach, whereas, presents the explicit form of the FRS of LTV systems.

1.1.2 Two-Player Differential Game

In an attempt to utilize the HJ reachability analysis in planning a safety-guaranteed trajectory, a pursuit-evasion differential game is formulated. Works in this class formulate the reachability problem as a two-player differential game, and compute the numerical solution of the HJB equation regarding the game by gridding state space and time. The game is consist of two systems: the planning system that corresponds to the reference trajectory, and the tracking system identical to the closed-loop dynamics. [4] performs HJ reachability analysis to compute the maximum tracking error resulting from the difference between the planning and tracking systems. The planning system can adjust its maneuver so that the tracking system does not collide with obstacles considering the maximum possible error. As an extension of [4], [5] considers various planning systems and computes the maximum tracking error for each planning model. In the online phase, the difficulty of environments (e.g. the distance between obstacles) determines an appropriate planning system to be used, considering the maximum tracking error. With the aid of a sampling-based planner, [20] provides safety-guaranteed exploration in unknown environments. This work builds a graph of states by recursively using the forward and backward reachability, and collision avoidance is guaranteed from the maximum tracking error computed offline.

This class of methods may lack scalability because solving HJB equations by gridding the state space leads to exponential computational complexity in the dimension of the state. Also, they require a reevaluation of the maximum tracking error when the system parameters are changed. The maximum error computed in the offline phase may not guarantee safety when additional disturbances suddenly interrupt the system. Although many efforts have attempted to solve the HJ reachability problem efficiently (for example, via system decomposition [21] and warm-start [22]), the computation time is still inappropriate for real-time applications.

1.1.3 Lyapunov Theory and SOS programming

Sum-of-squares (SOS) programming has been utilized to compute a funnel. Works in this category aim to find the optimal polynomial function that tightly approximates the value function of the HJB equation. In [7], the funnels of nonlinear systems combined with an LQR controller are computed around a given reference trajectory. But, the trajectory planning problem is not addressed. In an attempt to use the funnels in robust planning, [8] proposes the concept of a funnel library, which is the composition of funnels calculated around a finite number of predefined reference trajectories. The funnel library of the system is constructed during the offline phase. Then during the online phase, the optimal sequence and duration of the reference trajectory are planned so that the funnels do not collide with obstacles. [9] generates a safety-guaranteed flight trajectory for a multirotor system using the funnel library. It computes the funnels that describe the maximal tracking error of the translational dynamics in the offline phase, for different bounds on external force disturbances. In [10] and [11], the funnel associated with admissible parameters is precomputed in the offline step, and then the optimal (safe) parameter are selected in the online step.

Although the funnels provide almost exact bounds of states under bounded disturbances, those approaches may not be scalable because it is necessary to reconstruct the funnel library from scratch if some of the system parameters such as inertia and control gains are changed. Also, the planned trajectory from the funnel library no longer guarantees safety when the funnel library does not cover the magnitude of the actual disturbance encountered in runtime. The heavy computational burden of SOS programming makes it difficult to use the funnels for verifying the runtime safety of the system.

1.1.4 Tube-based Model Predictive Control

Works in tube-based model predictive control (TMPC) compute the reachable sets in runtime and approximate them as convex geometries, usually ellipsoids. [13] computes robust forward invariant tubes and the optimal control inputs at the same time, based on the boundness of differential inequalities [12]. Invariant tubes also provide theoretical enclosures of the system under the bounded disturbances. However, the computation of the robust forward invariant tubes requires the solution of a matrix-valued differential equation at each sampling time. So, the real-time generation of the reference trajectory based on the invariant tubes may not be possible for high-dimensional systems. [14] proposes an efficient approximation of the disturbance invariant sets and a robust planning method, based on a nonlinear model predictive control (NMPC). However, it requires numerical integration of all the states in the initial set, so lacks a theoretical guarantee of safety since the FRS of the system is likely to be underestimated.

1.1.5 Feature of the Work

Most of the existing studies mainly focus on how to utilize funnels that are computed in the offline step. Accordingly, if system parameters or disturbance conditions used in the offline phase are different from those in the online phase, there is a common limitation that the previously computed funnels can no longer be used to ensure the safety of the system. On the other hand, the proposed work focuses on the quick computation of the funnel itself. Thus, the proposed algorithm has the potential to react rapidly enough to the changes, and can be used for verifying the safety even if additional disturbances are applied to the system during runtime.

1.2 Contributions

The contributions of this dissertation are summarized as follows.

- **Characterization of the FRS of LTV systems**: A Hopf-based method that provides an outer approximation of the FRS of LTV systems is presented. It should be noted that the previous methods based on the Lax and Hopf theory do not provide an FRS but determine if a single state is in the FRS or not.

- **Tight ellipsoidal approximation of the FRS of LTV systems**: The proposed method finds the minimal volume ellipsoid that closely over-approximates the FRS of LTV systems. There exists no closed-form formula for a minimum volume composition of ellipsoids, and thus a fixed-point iteration is proposed for computational efficiency.

- **Polynomial approximation of the Hamiltonian**: Closed-form polynomial expressions that bound above the Hamiltonian of the HJB equation are presented. It should be noted that solving the HJB equation with grid-based approaches [23] requires nonlinear optimization on disturbances for each state on the grid. The proposition significantly reduces the burden required to compute the Hamiltonian.

- **LP-based computation of the funnel of nonlinear systems**: Thanks to the useful property of Bernstein polynomials, the proposed algorithm computes the funnel via LPs, which are computationally more tractable compared to SOS programming. To the best of the author's knowledge, this dissertation is the first work that utilizes Bernstein polynomials in computing the funnel.

- **Real-time planning with guaranteed safety**: A trajectory planning algorithm combined with the proposed funnel computation method is presented. Also, the performance of the proposed planning algorithm is validated through a flight experiment using a multirotor.

1.3 Outline

The outline of the thesis is as follows. Chapter 2 provides preliminaries for the HJ reachability analysis, which are thoroughly used in this work. Chapter 3 presents the analytic expression and tight outer approximation of the FRS for LTV systems. Chapter 4 provides the LP-based algorithm for computing the funnel of nonlinear systems. Chapter 5 describes how to generate the safety-guaranteed reference trajectory using the funnel. Chapter 6 provides experimental results, and Chapter 7 ends the thesis with concluding remarks.

2

Background

This section briefly summarizes the background materials of this work. The definition of FRS and funnel is presented first. As a tool for computing the FRS, the key concept of the HJ reachability analysis and the generalized Hopf formula are provided.

2.1 Forward Reachable Set and Funnel

Consider a nonlinear dynamics $f : \mathbb{R}^n \times \mathcal{W} \times [t_0, t_N] \to \mathbb{R}^n$ as

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

where $x(t) \in \mathbb{R}^n$ is state, $w(t) \in \mathcal{W} \subset \mathbb{R}^m$ is unknown disturbance, $t_0 \in \mathbb{R}$ is the initial time, and $t_N > t_0 \in \mathbb{R}$ represents a finite time horizon. n and $m \in \mathbb{R}$ are the dimension of the state and disturbance, respectively. $\mathcal{W}$ is a convex and bounded set of disturbances. It is assumed that the dynamics f is Lipschitz continuous in x and uniformly continuous in w. Also, without loss of generality, the system (2.1) is assumed to be centered at the origin, i.e. the solution of the following nominal dynamics represents $x(t) = 0$ for all $t \in [t_0, t_N]$:

$$\dot{x}(t) = f\big(x(t), 0, t\big), \quad x(t_0) = 0. \tag{2.2}$$

The FRS $\mathcal{X}(t) \subset \mathbb{R}^n$ of the system (2.1) is the set of all states to which a system can be driven from a given initial set of states $\mathcal{X}_0 \subset \mathbb{R}^n$ after a certain duration $t > t_0 \in \mathbb{R}$, in the presence of the disturbance [3].

Definition 2.1 (Forward reachable set)**.** *For a given initial set of states $\mathcal{X}_0$ and bounded set of disturbances $\mathcal{W}$, a forward reachable set $\mathcal{X} : [t_0, t_N] \to \mathcal{P}(\mathbb{R}^n)$ of the system* (2.1) *is*

$$\mathcal{X}(t) := \left\{ y \;\middle|\; \begin{array}{l} \forall \tau \in [t_0, t],\ \exists w(\tau) \in \mathcal{W}, \\ \dot{x}(\tau) = f\big(x(\tau), w(\tau), \tau\big), \\ x(t_0) \in \mathcal{X}_0,\ y = x(t) \end{array} \right\}, \tag{2.3}$$

where $\mathcal{P}(\mathbb{R}^n)$ represents the power set of $\mathbb{R}^n$.

The FRS of the system (2.1) is uniquely determined for the given initial set of states $\mathcal{X}_0$ and set of disturbances $\mathcal{W}$. With the definition of the FRS, the funnel $\mathcal{F}(t) \subset \mathbb{R}^n$ of the system (2.1) is defined as the following.

Definition 2.2 (Funnel)**.** *A funnel $\mathcal{F} : [t_0, t_N] \to \mathcal{P}(\mathbb{R}^n)$ of the system* (2.1) *is a set that contains the forward reachable set $\mathcal{X}(t)$ in* (2.3) *such that*

$$\mathcal{X}(t) \subseteq \mathcal{F}(t) \quad \forall t \in [t_0, t_N]. \tag{2.4}$$

In short, the funnel is an outer approximation of the FRS. Note that any subset of $\mathbb{R}^n$ can be a valid candidate for the funnel as far as the subset satisfies the condition (2.4). Thus, computing the funnel that tightly encloses the FRS is important to prevent unintended conservativeness due to approximation.

2.2 Hamilton-Jacobi Reachability Analysis

To compute the FRS, the reachability problem [3] is considered. Let $l : \mathbb{R}^n \to \mathbb{R}$ be a convex function such that $\mathcal{X}_0 = \{y \,|\, l(y) \leq 0\}$. The function l determines whether a particular state is in $\mathcal{X}_0$ or not. Consider the cost functional $J : \mathbb{R}^n \times \mathbb{R}^m \times [t_0, t_N] \to \mathbb{R}$ defined as

$$J(x, w, t) = l(x(t_0)) + \int_{t_0}^{t} L(w(\tau))\, d\tau,$$

where the function $L : \mathbb{R}^m \to \mathbb{R}$ is the running cost for the boundedness of disturbances such that

$$L(w) = \begin{cases} 0 & w \in \mathcal{W}, \\ \infty & w \notin \mathcal{W}. \end{cases}$$

The value function $V : \mathbb{R}^n \times [t_0, t_N] \to \mathbb{R}$ is the minimum of the cost over all possible disturbances:

$$V(x, t) = \min_{w}\; J(x, w, t). \tag{2.5}$$

The following theorem describes a temporal-spatial variation of the value function.

Theorem 2.1 (HJB equation [16])**.** *Suppose f in (2.1) is Lipschitz in (x, t) and l is Lipschitz in x. For $(x, \tau) \in \mathbb{R}^n \times [t_0, t]$, V in (2.5) is the unique viscosity solution to*

$$\begin{aligned} \frac{\partial V(x,\tau)}{\partial t} + H\Big(x, \frac{\partial V(x,\tau)}{\partial x}, \tau\Big) = 0 \quad &\textit{in} \quad \mathbb{R}^n \times (t_0, t), \\ V(x, \tau) = l(x) \quad &\textit{on} \quad \mathbb{R}^n \times \{\tau = t_0\}, \end{aligned} \tag{2.6}$$

where the Hamiltonian $H : \mathbb{R}^n \times \mathbb{R}^n \times [t_0, t] \to \mathbb{R}$ is

$$H(x, \lambda, \tau) \coloneqq \max_{w \in \mathcal{W}} \big(\lambda \cdot f(x, w, \tau)\big). \tag{2.7}$$

The reachability problem implies that there exists a sequence of disturbances that drives x inside the subzero level set of $V(x, t_0)$ to the subzero level set of $V(x, t)$. In other words, the sequence of optimal disturbance $w^\star(\tau)$ can steer x in the zero level set of $V(x,t_0)$ to the zero level set of $V(x,t)$. Thus, the FRS can be expressed in terms of the value function:

$$\mathcal{X}(t) = \{x \mid V(x,t) \le 0\}. \tag{2.8}$$

2.3 The Generalized Hopf Formula

Considering that it is difficult to solve the HJB equation for most state-dependent nonlinear systems, grid-based methods such as the level-set method [23] are utilized to obtain a numerical solution. This makes it intractable to compute the FRS for high-dimensional systems over 5D. To alleviate this computational complexity, the generalized Hopf formula that provides an analytic solution to the HJB equation for state-independent systems is utilized. Since the dynamics (2.1) depends on the state, the linearization of the dynamics (2.1) is considered. Then, the coordinate transform [18] that converts the linear dynamics to state-independent is employed.

The linearization of (2.1) can be expressed as

$$\dot{x}(t) = A(t)x(t) + D(t)w(t), \tag{2.9}$$

where $A(t) \in \mathbb{R}^{n\times n}$ and $D(t) \in \mathbb{R}^{n\times m}$ are the time-varying matrices from the linearization. The corresponding FRS of the LTV system (2.9) is defined as

$$\underline{\mathcal{X}}(t) \coloneqq \left\{ y \;\middle|\; \begin{array}{l} \forall\tau \in [t_0,t],\ \exists w(\tau) \in \mathcal{W}, \\ \dot{x}(\tau) = A(\tau)x(\tau) + D(\tau)w(\tau), \\ x(t_0) \in \mathcal{X}_0,\ y = x(t) \end{array} \right\}. \tag{2.10}$$

As proposed in [18], we consider a time-varying coordinate transform such that $x(t) = \Phi(t,t_0)z(t)$, where $\Phi(t,t_0) \in \mathbb{R}^{n\times n}$ is the state transition matrix of a homogenous system $\dot{x}(t) = A(t)x(t)$ from t_0 to t. From (2.9), the state-independent dynamics in the z coordinate is derived as

$$\begin{aligned}\dot{z}(t) &= \dot{\Phi}(t_0,t)x(t) + \Phi(t_0,t)\dot{x}(t) \\ &= -\Phi(t_0,t)A(t)x(t) + \Phi(t_0,t)\big(A(t)x(t) + D(t)w(t)\big) \\ &= \Phi(t_0,t)D(t)w(t) \coloneqq D_z(t)w(t),\end{aligned} \tag{2.11}$$

where $\dot{\Phi}(t,t_0) = A(t)\Phi(t,t_0)$ is used.

Similar to (2.5), a cost functional and the corresponding value function of the reachability problem in the z coordinate are

$$\begin{aligned}J_z(z,w,t) &= l_z(z(t_0)) + \int_{t_0}^{t} L(w(\tau))d\tau, \\ V_z(z,t) &= \min_{w} \; J_z(z,w,t),\end{aligned} \tag{2.12}$$

where the subscript z represents the function expressed in the z coordinate. The generalized Hopf formula which presents the analytic expression of the value function in (2.12) is summarized as follows.

Theorem 2.2 (The generalized Hopf formula [24, 1]). *For a given $(z,\tau) \in \mathbb{R}^n \times [t_0,t]$, the solution to the HJB equation associated with the dynamics* (2.11) *and cost* (2.12) *is*

$$V_z(z,t) = -\min_{\lambda}\left\{ -z\cdot\lambda + h_z(\lambda) + \int_{t_0}^{t} H_z(\lambda,\tau)d\tau \right\} \tag{2.13}$$

where the Hamiltonian $H_z : \mathbb{R}^n \times [t_0,t] \to \mathbb{R}$ is

$$H_z(\lambda,\tau) \coloneqq \max_{w\in\mathcal{W}} \big(\lambda\cdot(D_z(\tau)w)\big), \tag{2.14}$$

and $h_z : \mathbb{R}^n \to \mathbb{R}$ is the Fenchel-Legendre transformation of l_z defined as

$$h_z(\lambda) \coloneqq \max_{z} \big(z\cdot\lambda - l_z(z)\big). \tag{2.15}$$

Since the dynamics (2.11) is not a function of z, the corresponding Hamiltonian is also independent of z. This is the main reason that the value function can be expressed as an analytic form through the generalized Hopf formula. Compared with solving the HJB equation with numerical methods, the generalized Hopf formula provides a much more computationally tractable way. Similar to (2.8), the FRS of the linearized system (2.11) can be expressed in terms of the value function as

$$\mathcal{Z}(t) = \{y \mid V_z(y, t) \leq 0\}. \tag{2.16}$$

Considering the coordinate transform that we employed, the FRS of the linearized system (2.9) can also be expressed as

$$\underline{\mathcal{X}}(t) = \{\Phi(t, t_0)y \mid y \in \mathcal{Z}(t)\}. \tag{2.17}$$

By using the formula, it is possible to compute the value $V_z(z, t)$ in (2.13) for a given z with the temporal discretization and numerical optimization algorithms. Thus, determining whether a particular z is in the FRS is also possible with much less computational load compared to grid-based methods. However, simply checking whether one state is inside the FRS is not sufficient to ensure the safety of all states in the initial set. Since an explicit representation of $\mathcal{Z}(t)$ in (2.16) cannot be obtained using the generalized Hopf formula alone, the characterization of the shape of $\mathcal{Z}(t)$ is proposed in Chapter 3. Also, it should be noted that $\mathcal{X}(t)$ in (2.3) and $\underline{\mathcal{X}}(t)$ in (2.10) are not identical due to the linearization error, and thus $\underline{\mathcal{X}}(t)$ cannot be used for guaranteeing the safety of the nonlinear system. In Chapter 4, an algorithm for computing the funnel of nonlinear systems via a polynomial approximation of the value function $V(x, t)$ in (2.5) is presented.

3

Reachability Analysis of LTV Systems

This chapter is dedicated to computing the explicit form of the FRS of LTV systems and its ellipsoidal approximation based on the generalized Hopf formula presented in Theorem 2.2. Section 3.1 proposes that the FRS of the transformed LTV system $\mathcal{Z}(t)$ in (2.16) is the Minkowski sum of sets: the initial set and the sets propagated by disturbances. Then, as the first step to approximate $\mathcal{Z}(t)$, Section 3.2 proposes an ellipsoid that encloses the sets due to disturbances. Section 3.3 presents an iterative algorithm for computing the minimal ellipsoid that encloses the Minkowski sum of ellipsoids to avoid overly conservative approximation. Finally, the ellipsoidal approximation of the FRS of the linearized LTV system $\underline{\mathcal{X}}(t)$ in (2.10) is presented in Section 3.4.

3.1 Characterization of Forward Reachable Set

The objective of this section is to find an analytic expression of the FRS of the transformed LTV system in (2.11). Consider the following ellipsoidal initial set of states

$$\mathcal{X}_0 = \mathcal{E}(Q_0) \coloneqq \left\{ Q_0^{\frac{1}{2}} v \mid \|v\|_2 \leq 1, v \in \mathbb{R}^n \right\} \subset \mathbb{R}^n,$$

and the initial value function

$$l(x) = x^\top Q_0^{-1} x - 1,$$

where $\mathcal{E} : \mathbb{S}_{++}^n \to \mathcal{P}(\mathbb{R}^n)$ describes an ellipsoid centered at the origin, $Q_0 \in \mathbb{S}_{++}^n$ is the shape matrix of the ellipsoid $\mathcal{E}(Q_0)$, and $\mathbb{S}_{++}^n$ represents a set of $n \times n$ symmetric positive definite matrices. Since $z(t_0) = \Phi(t_0, t_0)x(t_0) = x(t_0)$, the initial set of states in the z coordinate is identical to the initial set of states in the x coordinate. The initial set and the initial value function in the z coordinate are

$$\begin{aligned} \mathcal{Z}(t_0) &= \mathcal{E}(Q_0), \\ l_z(z) &= z^\top Q_0^{-1} z - 1. \end{aligned} \tag{3.1}$$

From (2.15), the convex conjugate of $l_z(z)$ is

$$h_z(\lambda) = \frac{1}{4}\lambda^\top Q_0 \lambda + 1. \tag{3.2}$$

Now, the following assumption on the disturbance is presented.

Assumption 3.1 (Independently bounded disturbance)**.** *Each element of disturbances w_i, the i-th element of w, affects the system independently. Also, without loss of generality, w_i is normalized as $-1 \leq w_i \leq 1$ for all $i \in \{1, \ldots, m\}$, i.e. $\mathcal{W} = [-1, 1]^m$.*

Based on Assumption 3.1, the Hamiltonian in (2.14) can be expressed as

$$H_z(\lambda, t) = \sum_{i=1}^{m} |\lambda^\top D_{z,i}(t)| \tag{3.3}$$

with the optimal disturbance

$$w^\star(t) = \operatorname{sign}(D_z(t)^\top \lambda),$$

where $D_{z,i}(t)$ is the i-th column of $D_z(t)$ in (2.11).

Proposition 3.1 (FRS of a state-independent LTV system)**.** *For a given initial set* $\mathcal{Z}(t_0)$ *in* (3.1) *and Hamiltonian* H_z *in* (3.3)*, the FRS of the system* (2.11) *is*

$$\mathcal{Z}(t) = \mathcal{Z}(t_0) \oplus \mathcal{D}(t, t_0), \tag{3.4}$$

where the set due to the disturbance $\mathcal{D}(t, t_0) \subset \mathbb{R}^n$ *is the Minkowski sum of* $\mathcal{D}_i(t, t_0) \subset \mathbb{R}^n$ *such that*

$$\begin{aligned} \mathcal{D}(t, t_0) &= \bigoplus_{i=1}^{m} \mathcal{D}_i(t, t_0), \\ \mathcal{D}_i(t, t_0) &= \left\{ \int_{t_0}^{t} D_{z,i}(\tau) \text{sign}\Big(D_{z,i}(\tau)^\top Q_0^{-\frac{1}{2}} v \Big) d\tau \;\Big|\; \|v\|_2 \le 1, v \in \mathbb{R}^n \right\}. \end{aligned} \tag{3.5}$$

Proof. Consider the property of Fenchel-Legendre transform [25] that the derivative of a convex function is the optimal argument of its conjugate. Let $\lambda^\star \in \mathbb{R}^n$ be the minimizer of (2.13) such that

$$V_z(z, t) = z \cdot \lambda^\star - h_z(\lambda^\star) - \int_{t_0}^{t} H_z(\lambda^\star, \tau) d\tau. \tag{3.6}$$

Since h_z in (3.2) and H_z in (3.3) are convex in λ, the first-order optimality condition is

$$\frac{\partial}{\partial \lambda}\Big(- z \cdot \lambda + h_z(\lambda) + \int_{t_0}^{t} H_z(\lambda, \tau) d\tau \Big)\Big|_{\lambda=\lambda^\star} = 0.$$

The trajectory of z along $\lambda^\star$ can be expressed as

$$z^\star(t) \coloneqq z(t; \lambda^\star) = \frac{\partial h_z(\lambda)}{\partial \lambda}\Big|_{\lambda=\lambda^\star} + \int_{t_0}^{t} \frac{\partial H_z(\lambda, \tau)}{\partial \lambda}\Big|_{\lambda=\lambda^\star} d\tau. \tag{3.7}$$

Substituting (3.7) into (3.6) yields the value function expressed in $\lambda^\star$ as

$$\begin{aligned} V_z(\lambda^\star, t) &= p^{\star\top} \left(\frac{\partial h_z(\lambda)}{\partial \lambda}\Big|_{\lambda=\lambda^\star} + \int_{t_0}^{t} \frac{\partial H_z(\lambda, \tau)}{\partial \lambda}\Big|_{\lambda=\lambda^\star} d\tau \right) - h_z(\lambda^\star) - \int_{t_0}^{t} H_z(\lambda^\star, \tau) d\tau \\ &= \frac{1}{4} \lambda^{\star\top} Q_0 \lambda^\star - 1, \end{aligned} \tag{3.8}$$

since

$$\begin{aligned}\frac{\partial h_z(\lambda)}{\partial \lambda}\Big|_{\lambda=\lambda^\star} &= \frac{1}{2}Q_0\lambda^\star,\\ \frac{\partial H_z(\lambda,\tau)}{\partial \lambda}\Big|_{\lambda=\lambda^\star} &= \sum_{i=1}^{m} D_{z,i}(\tau)\text{sign}\big(D_{z,i}(\tau)^\top\lambda^\star\big).\end{aligned}$$

The subzero level set of (3.8) is

$$V_z(\lambda^\star,t)\le 0 \iff \lambda^\star \in \big\{2Q_0^{-\frac{1}{2}}v \mid \|v\|_2\le 1, v\in\mathbb{R}^n\big\}. \tag{3.9}$$

By composing (3.7) and (3.9), we obtain

$$V_z(z,t)\le 0 \iff z\in\mathcal{Z}(t_0)\oplus\mathcal{D}(t,t_0).$$

Therefore, the proposed set $\mathcal{Z}(t)$ in (3.4) is the FRS of the system (2.11). □

Remark 3.1. *This analysis can be extended with other assumptions on the disturbance such as $\|w(t)\|_2\le 1$ for all $t\in[t_0,t_N]$. Different forms of the Hamiltonian and the value function can be derived similarly.*

Remark 3.2. *The value function presented in* (3.8) *is independent of time. Thus, the Hamiltonian and its time derivative are 0 for all $t\in[t_0,t_N]$ if the initial value of V_z is 0.*

Fig. 3.1 presents the comparison of the FRS $\mathcal{Z}(t)$ and the Minkowski sum of sets in (3.4) for a state-independent 2-state LTV system. The level set toolbox [23] is utilized to compute the FRS $\mathcal{Z}(t)$ by solving the HJB equation derived from (2.12). Also, numerical integration of (3.5) with 0.01 seconds of time step is performed to obtain $\mathcal{D}_i(t,t_0)$ and $\mathcal{D}(t,t_0)$. As expected, the shapes of two sets $\mathcal{Z}(t)$ and $\mathcal{Z}(t_0)\oplus\mathcal{D}(t,t_0)$ are almost identical, which validates Proposition 3.1.

Although the explicit form of the FRS $\mathcal{Z}(t)$ is analyzed, the real-time computation of the Minkowski sum of sets in (3.4) is intractable as the number of state increases. Methods for addressing such issues are discussed in the next section.

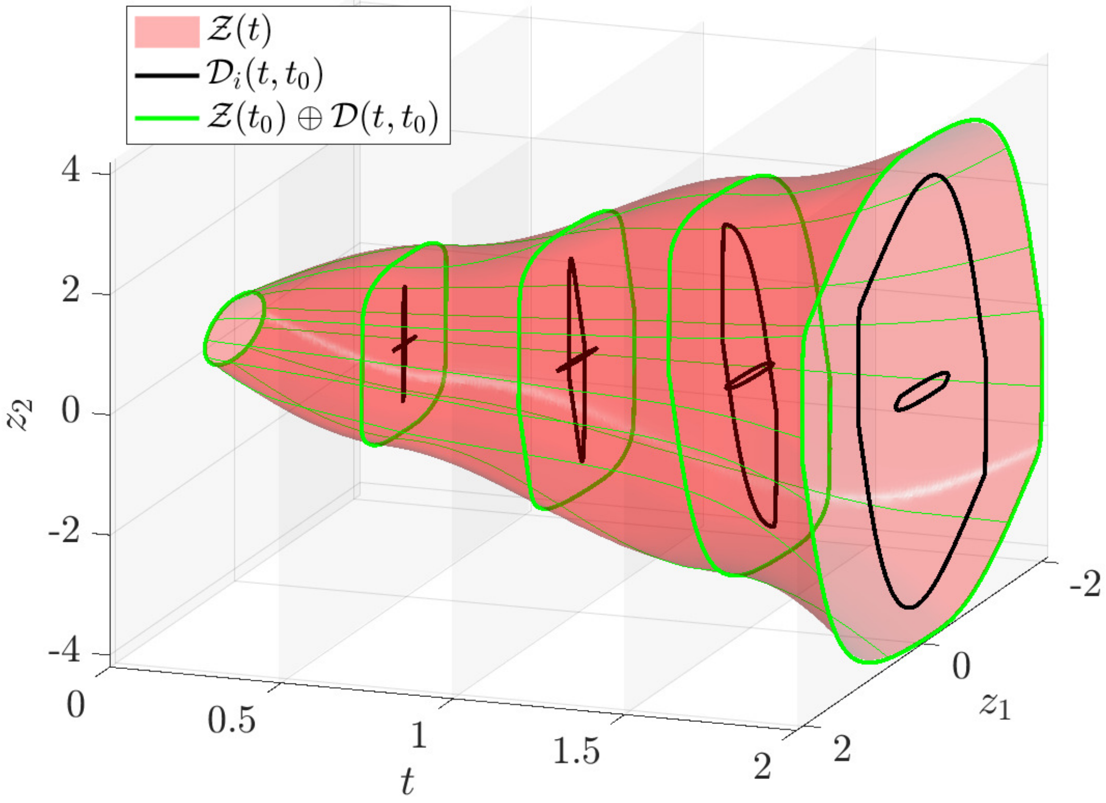


Figure 3.1: FRS of a state-independent 2-state LTV system: $\dot{z}_1(t) = 0.4\cos(t)w_1(t) - 0.4t^2w_2(t)$, $\dot{z}_2(t) = 0.08tw_1(t) + 2.8\cos(3t)w_2(t)$ with the initial shape matrix $Q_0 = \text{diag}(0.25, 0.25)$. The red surface describes the zero level set of the solution of the HJB equation, the black lines represent the regions due to two channels of disturbance respectively. The green lines are the Minkowski sum of sets in (3.4).

3.2 Ellipsoidal Approximation of Sets due to Disturbances

This section discusses an ellipsoidal approximation of the set due to disturbances $\mathcal{D}(t, t_0)$ in (3.4), which is the main component of the FRS of the state-independent LTV system. As the first step, an ellipsoid which encloses $\mathcal{D}_i(t, t_0)$ in (3.5) is proposed.

Proposition 3.2 (Outer approximation of $\mathcal{D}_i$)**.** *The set* $\mathcal{D}_i(t_b, t_a)$ *in* (3.5) *is the subset of* $\mathcal{E}(Z_i(t_b, t_a))$ *if*

$$Z_i(t_b, t_a) = (t_b - t_a) \int_{t_a}^{t_b} \left(D_{z,i}(\tau) D_{z,i}(\tau)^\top + \epsilon I_n \right) d\tau \in \mathbb{S}_{++}^n, \tag{3.10}$$

where $I_n \in \mathbb{R}^{n \times n}$ *is the identity matrix, and* $\epsilon \in \mathbb{R}$ *is an arbitrarily small positive scalar.*

Proof. The objective is to prove

$$d^\top Z_i(t_b, t_a)^{-1} d - 1 \leq 0$$

for all $d \in \mathcal{D}_i(t_b, t_a)$. Substituting $Z_i(t_b, t_a)$ in (3.10) into the above inequality yields

$$(t_b - t_a) - d^\top \left(\int_{t_a}^{t_b} \left(D_{z,i}(\tau) D_{z,i}(\tau)^\top + \epsilon I_n \right) d\tau \right)^{-1} d \geq 0.$$

From the positive semi-definiteness condition of the Schur complement [26], the left-hand side of the above inequality is the Schur complement of the following matrix:

$$M(t_b, t_a) = \begin{bmatrix} t_b - t_a & d^\top \\ d & \int_{t_a}^{t_b} \left(D_{z,i}(\tau) D_{z,i}(\tau)^\top + \epsilon I_n \right) d\tau \end{bmatrix}.$$

Let $s(\tau) \in \mathbb{R}$ be sign $\left(D_{z,i}(\tau)^\top Q^{-\frac{1}{2}} v \right)$ for all $\tau \in [t_a, t_b]$. Now, consider the case when $s(\tau)$ is constant and the case when $s(\tau)$ varies over τ.

Case 1) Suppose that $s(\tau) = 1$ for all $\tau \in [t_a, t_b]$. Then, $d = \int_{t_a}^{t_b} D_{z,i}(\tau) d\tau$, and $M(t_b, t_a)$ is the integration of the following matrix:

$$
\begin{aligned}
M(t_b, t_a) &= \int_{t_a}^{t_b} m(\tau) d\tau, \\
m(\tau) &= \begin{bmatrix} 1 & D_{z,i}(\tau)^\top \\ D_{z,i}(\tau) & D_{z,i}(\tau) D_{z,i}(\tau)^\top + \epsilon I_n \end{bmatrix} \in \mathbb{S}_{++}^{n+1}.
\end{aligned}
$$

Note that $m(\tau)$ and $M(t_b, t_a)$ are positive definite matrices, and the right bottom block of $M(t_b, t_a)$ is also positive definite. From the positive semi-definiteness condition, the Schur complement of $M(t_b, t_a)$ is positive semi-definite.

Case 2) Let $t' \in [t_a, t_b]$ be the time instant when $s(\tau)$ changes. Assume that $s(\tau) = 1$ if $\tau \leq t'$ and $s(\tau) = -1$ if $\tau > t'$. In this case, $M(t_b, t_a)$ can be expressed as the sum of two matrices

$$
\begin{aligned}
M(t_b, t_a) &= \int_{t_a}^{t'} m_1(\tau) d\tau + \int_{t'}^{t_b} m_2(\tau) d\tau, \\
m_1(\tau) &= \begin{bmatrix} 1 & D_{z,i}(\tau)^\top \\ D_{z,i}(\tau) & D_{z,i}(\tau) D_{z,i}(\tau)^\top + \epsilon I_n \end{bmatrix} \in \mathbb{S}_{++}^{n_x+1}, \\
m_2(\tau) &= \begin{bmatrix} 1 & -D_{z,i}(\tau)^\top \\ -D_{z,i}(\tau) & D_{z,i}(\tau) D_{z,i}(\tau)^\top + \epsilon I_n \end{bmatrix} \in \mathbb{S}_{++}^{n_x+1}.
\end{aligned}
$$

Since all the matrices that compose $M(t_b, t_a)$ are positive definite, the Schur complement of $M(t_b, t_a)$ is also positive semi-definite. □

The computation of the outer ellipsoid in (3.10) requires the integration of a time-varying matrix. For computational simplicity, the time-varying system matrices in (2.9) are approximated as piecewise constant matrices between some sampled time indices, namely t_{k-1} and t_k, where $t_0 \leq t_{k-1} \leq t_k \leq t_N$. Let $A_k := A(t_{k-1}) \in \mathbb{R}^{n \times n}$ and $D_k := D(t_{k-1}) \in \mathbb{R}^{n \times m}$ be the constant system matrices for $t \in [t_{k-1}, t_k)$. Using the relation $D_z(t) = \Phi(t_0, t) D(t)$ in (2.11), a term in the right-hand side of (3.10) can be written as

$$
\int_{t_{k-1}}^{t_k} D_{z,i}(\tau) D_{z,i}(\tau)^\top d\tau = \Phi(t_0, t_{k-1}) N_i(t_k, t_{k-1}) \Phi(t_0, t_{k-1})^\top,
$$

where $N_i(t_k, t_{k-1}) \in \mathbb{R}^{n\times n}$ is the matrix such that

$$N_i(t_k, t_{k-1}) = \int_0^{t_k - t_{k-1}} \exp(-A_k\tau)\big(D_{k,i}D_{k,i}^\top\big)\exp(-A_k^\top\tau)d\tau,$$

and $D_{k,i} \in \mathbb{R}^n$ is the i-th column of D_k. So, $N_i(t_k, t_{k-1})$ is the solution of the following Lyapunov equation:

$$\begin{aligned} &A_k N_i(t_k, t_{k-1}) + N_i(t_k, t_{k-1})A_k^\top = \\ &\qquad D_{k,i}D_{k,i}^\top - \exp(-A_k(t_k - t_{k-1}))D_{k,i}D_{k,i}^\top \exp(-A_k^\top(t_k - t_{k-1})). \end{aligned} \tag{3.11}$$

Note that the above equation has a unique solution whenever A_k is nonsingular [27]. Even if A_k is singular, it is possible to compute the solution using [28]. Consequently, $Z_i(t_k, t_{k-1})$ in (3.10) is expressed as

$$Z_i(t_k, t_{k-1}) = (t_k - t_{k-1})\Phi(t_0, t_{k-1})N_i(t_k, t_{k-1})\Phi(t_0, t_{k-1})^\top + \epsilon(t_k - t_{k-1})^2 I_n.$$

Now, let us focus on the outer approximation of $\mathcal{D}_i(t_k, t_0)$ in (3.5). One may consider a direct integration of (3.10) from t_0 to t_k which yields

$$\mathcal{D}_i(t_k, t_0) \subseteq \mathcal{E}(Z_i(t_k, t_0)), \tag{3.12}$$

where

$$Z_i(t_k, t_0) = \sum_{j=1}^{k} \frac{Z_i(t_j, t_{j-1})}{\alpha_j}, \quad \alpha_j = \frac{t_j - t_{j-1}}{t_k - t_0} \in \mathbb{R}.$$

Also, $\mathcal{D}_i(t_k, t_0)$ can be approximated as the Minkowski sum of ellipsoids as the following:

$$\mathcal{D}_i(t_k, t_0) = \bigoplus_{j=1}^{k} \mathcal{D}_i(t_j, t_{j-1}) \subseteq \bigoplus_{j=1}^{k} \mathcal{E}(Z_i(t_j, t_{j-1})). \tag{3.13}$$

It is evident that (3.12) contains (3.13), from the formula of the ellipsoidal bounding of the

Minkowski sum of ellipsoids [29], which represents

$$\bigoplus_{j=1}^{k} \mathcal{E}(Z_i(t_j, t_{j-1})) \subseteq \mathcal{E}\left(\sum_{j=1}^{k} \frac{Z_i(t_j, t_{j-1})}{\alpha_j}\right),$$

for all $\alpha_j > 0$ such that $\sum \alpha_j = 1$. Consequently, (3.12) is just one of many ellipsoids which includes $\mathcal{D}_i(t_k, t_0)$, and thus (3.13) is used for the outer approximation of $\mathcal{D}_i(t_k, t_0)$. For a compact approximation, the minimum volume ellipsoid is defined as

$$Y_i(t_k, t_0) \coloneqq \min_{\alpha_1, \ldots, \alpha_k} \left(\log\det\left(\sum_{j=1}^{k} \frac{Z_i(t_j, t_{j-1})}{\alpha_j}\right)\right).$$

The computation of the minimum volume ellipsoid will be detailed in Section 3.3.

Fig. 3.2 illustrates the inclusion relationship between $\mathcal{D}_i(t, t_0)$, $\mathcal{E}(Z_i(t, t_0))$, and $\mathcal{E}(Y_i(t, t_0))$. Numerical integration from $t_0 = 0$ to $t_N = 2$ with 0.01 seconds of time step is performed to compute Z_i in (3.10). As proposed, $\mathcal{E}(Z_i(t, t_0))$ and $\mathcal{E}(Y_i(t, t_0))$ tightly enclose $\mathcal{D}_i(t, t_0)$ without any invasion. Also, $\mathcal{E}(Z_i(t, t_0))$ is slightly larger than $\mathcal{E}(Y_i(t, t_0))$, and thus the proposed minimum volume composition $Y_i(t, t_0)$ can prevent unintended conservatism of the ellipsoidal approximation.

From (3.5), $\mathcal{D}(t_k, t_0)$ is enclosed by the Minkowski sum of ellipsoids as

$$\mathcal{D}(t_k, t_0) = \bigoplus_{i=1}^{m} \mathcal{D}_i(t_k, t_0) \subseteq \bigoplus_{i=1}^{m} \bigoplus_{j=1}^{k} \mathcal{E}(Z_i(t_j, t_{j-1})).$$

Consequently, the outer approximation of $\mathcal{Z}(t_k)$ in (3.4) is

$$\mathcal{Z}(t_k) \subseteq \mathcal{E}(Q_0) \oplus \bigoplus_{i=1}^{m} \bigoplus_{j=1}^{k} \mathcal{E}(Z_i(t_j, t_{j-1})). \tag{3.14}$$

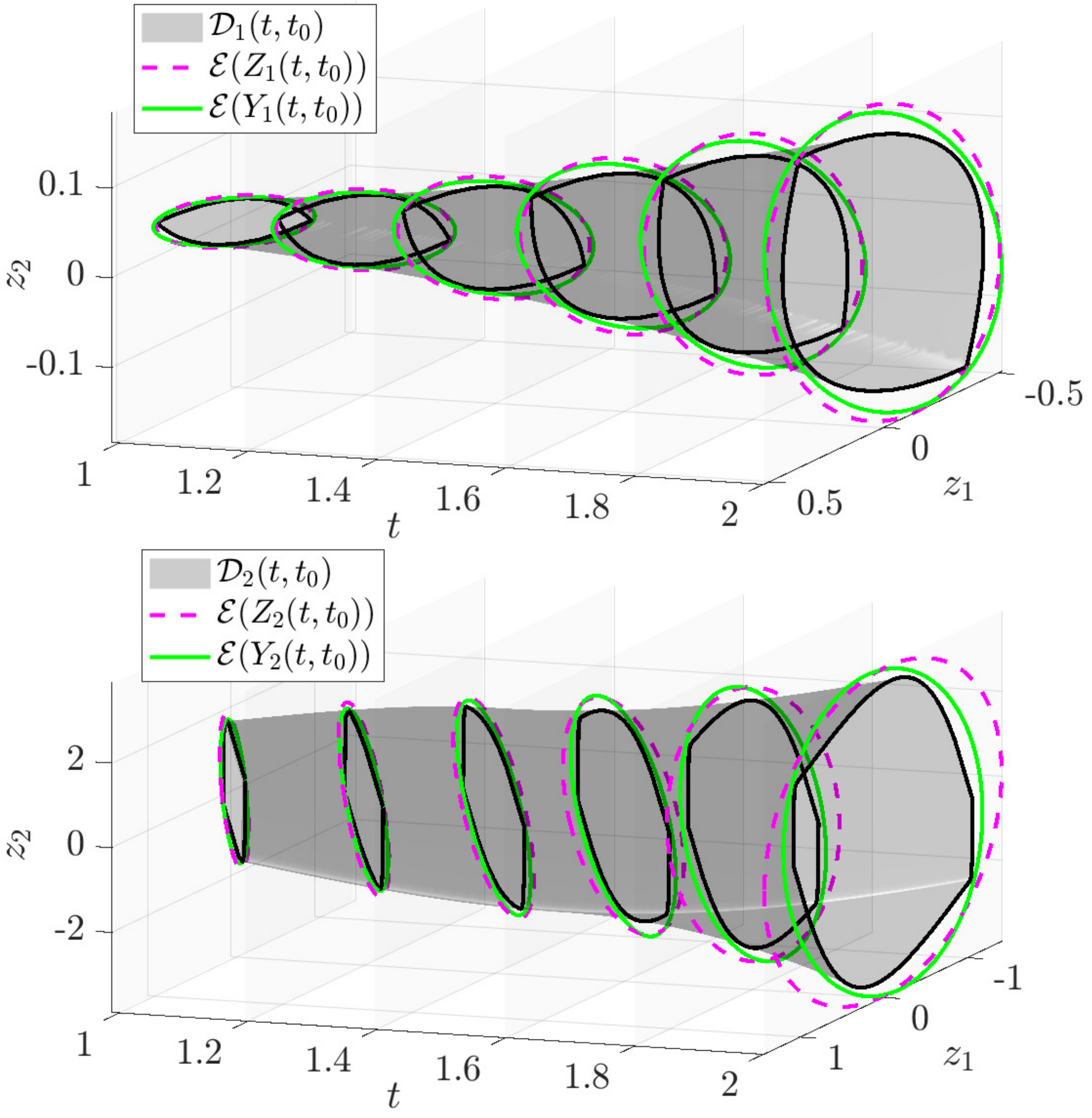


Figure 3.2: Results of the ellipsoidal approximation for the 2-state LTV system discussed in Fig. 3.1. The upper and lower figures are the results due to w_1 and w_2, i.e., the first and second channels of disturbance, respectively. The shaded gray surfaces are the sets due to disturbance (identical to the black lines of Fig. 3.1), the dashed magenta lines represent the numerical integration of $Z_i(t_k, t_{k-1})$ in (3.10), and the green lines describe the proposed approximation.

3.3 Minimum Volume Composition of Ellipsoids

The Minkowski sum of ellipsoids is not an ellipsoid in general, making it difficult to represent the set $\mathcal{Z}(t)$ in (3.14) as an analytic form. The objective of this section is to find a shape matrix of the minimum volume ellipsoid which contains the Minkowski sum of multiple ellipsoidal regions. Consider N number of shape matrices, namely $B_i \in \mathbb{S}^n_{++}$ for all $i = \{1, \dots, N\}$. As mentioned in Section 3.2, the ellipsoidal bounding formula [29] presents

$$\bigoplus_{i=1}^{N} \mathcal{E}(B_i) \subseteq \mathcal{E}(B(\alpha)), \quad B(\alpha) = \sum_{i=1}^{N} \frac{B_i}{\alpha_i}$$

with the following constraints

$$\sum_{i=1}^{N} \alpha_i = 1, \quad \alpha_i > 0, \tag{3.15}$$

where $\alpha \in \mathbb{R}^N$ is the concatenation of coefficients $\alpha_i \in \mathbb{R}$.

Since the objective is a compact composition of ellipsoidal regions, the coefficient α must be determined under some appropriate criteria. This work aims to compute α which minimizes the volume of $B(\alpha)$ such that

$$\alpha^{det} \coloneqq \arg\min \big(\log\det(B(\alpha)) \big). \tag{3.16}$$

However, numerical optimization or iterative methods [30] are required to compute the optimal argument of (3.16), because a closed-form solution does not exist. To compute the minimum volume ellipsoid in a computationally efficient manner, a fixed-point iteration algorithm is presented.

Proposition 3.3 (Minimum volume composition)**.** *Given an initial guess* $\alpha^{[0]} \in \mathbb{R}^N$ *that satisfies* (3.15)*, the following iteration*

$$\alpha_i^{[k+1]} = \psi_i(\alpha^{[k]}) = \sqrt{\frac{1}{n} trace\big(B(\alpha^{[k]})^{-1} B_i\big)} \tag{3.17}$$

converges to α^{det} in (3.16), where ψ_i is the i-th element of a nonlinear map $\psi : \mathbb{R}^N \to \mathbb{R}^N$, and $\alpha^{[k]}$ represents α at the k-th iteration.

Proof. Note that the cost in (3.16) is convex for all α that satisfy the condition (3.15) as proposed in [31]. Consider an augmented cost function

$$C_a(\alpha, \gamma) = \log\det(B(\alpha)) + \gamma \left(\sum_{j=1}^{N} \alpha_j - 1 \right), \tag{3.18}$$

where $\gamma \in \mathbb{R}$ is the Lagrange multiplier for the equality constraint in (3.15). The first-order optimality condition of (3.18) for α_i is

$$\frac{\partial}{\partial \alpha_i} C_a(\alpha, \gamma) = -\frac{1}{\alpha_i^2} \text{trace}\big(B(\alpha)^{-1} B_i\big) + \gamma = 0, \tag{3.19}$$

since

$$\frac{\partial}{\partial \alpha_i} \log\det(B(\alpha)) = \text{trace}\left(B(\alpha)^{-1} \frac{\partial}{\partial \alpha_i} B(\alpha) \right).$$

Rearranging (3.19) yields

$$\text{trace}\left(B(\alpha)^{-1} \left(\frac{B_i}{\alpha_i} \right) \right) - \gamma \alpha_i = 0. \tag{3.20}$$

The summation of (3.20) from $i = 1$ to $i = N$ determines γ as

$$\begin{aligned} &\text{trace}\left(B(\alpha)^{-1} \left(\sum_{i=1}^{N} \frac{B_i}{\alpha_i} \right) \right) - \gamma \sum_{i=1}^{N} \alpha_i \\ &= \text{trace}\big(B(\alpha)^{-1} B(\alpha)\big) - \gamma \sum_{i=1}^{N} \alpha_i \\ &= n - \gamma = 0. \end{aligned}$$

Consequently, (3.19) can be rewritten as

$$\alpha_i^2 = \frac{1}{n} \text{trace}\big(B(\alpha)^{-1} B_i\big),$$

which concludes the fixed-point iteration $\alpha = \psi(\alpha)$ in (3.17). The convexity of the cost (3.16) supports that the iteration (3.17) converges to the unique and optimal point α^{det}. □

Remark 3.3. *The proposed iteration* (3.17) *updates α for multiple ellipsoids as described in Fig. 3.3. This is the main advantage with respect to the previous works [31, 32] which recursively compose multiple ellipsoids one after another. Also,* (3.17) *can be interpreted as the generalization of [30] where the case of $N = 2$ only is covered.*

Remark 3.4. *Allowing some abuse of notation, the minimum volume bounding ellipsoid from* (3.16) *is denoted as*

$$B(\alpha^{det}) := \bigoplus_{i=1}^{N} B_i. \tag{3.21}$$

Remark 3.5. *The minimum volume composition is invariant under coordinate transformation since*

$$\begin{aligned}\alpha^{det} &= \arg\min\big(\log\det\big(RB(\alpha)R^{\top}\big)\big)\\ &= \arg\min\big(\log\big(\det(R)\det(B(\alpha))\det(R^{\top})\big)\big)\\ &= \arg\min\big(\,\log\det(B(\alpha))\big)\end{aligned}$$

for all $R \in \mathbb{R}^{n\times n}$.

3.4 Ellipsoidal Approximation of FRS of LTV Systems

Based on the minimum volume composition operator proposed in (3.21), the objective of this section is to find ellipsoids $\mathcal{E}(Q_z(t))$ and $\mathcal{E}(Q_x(t))$ that enclose the FRS of the LTV systems in (2.9) and (2.11) such that

$$\mathcal{Z}(t) \subseteq \mathcal{E}(Q_z(t)), \quad \underline{\mathcal{X}}(t) \subseteq \mathcal{E}(Q_x(t)) \quad \forall t \in [t_0, t_N].$$

From (3.14), the shape matrix for the FRS of the state-independent system is computed

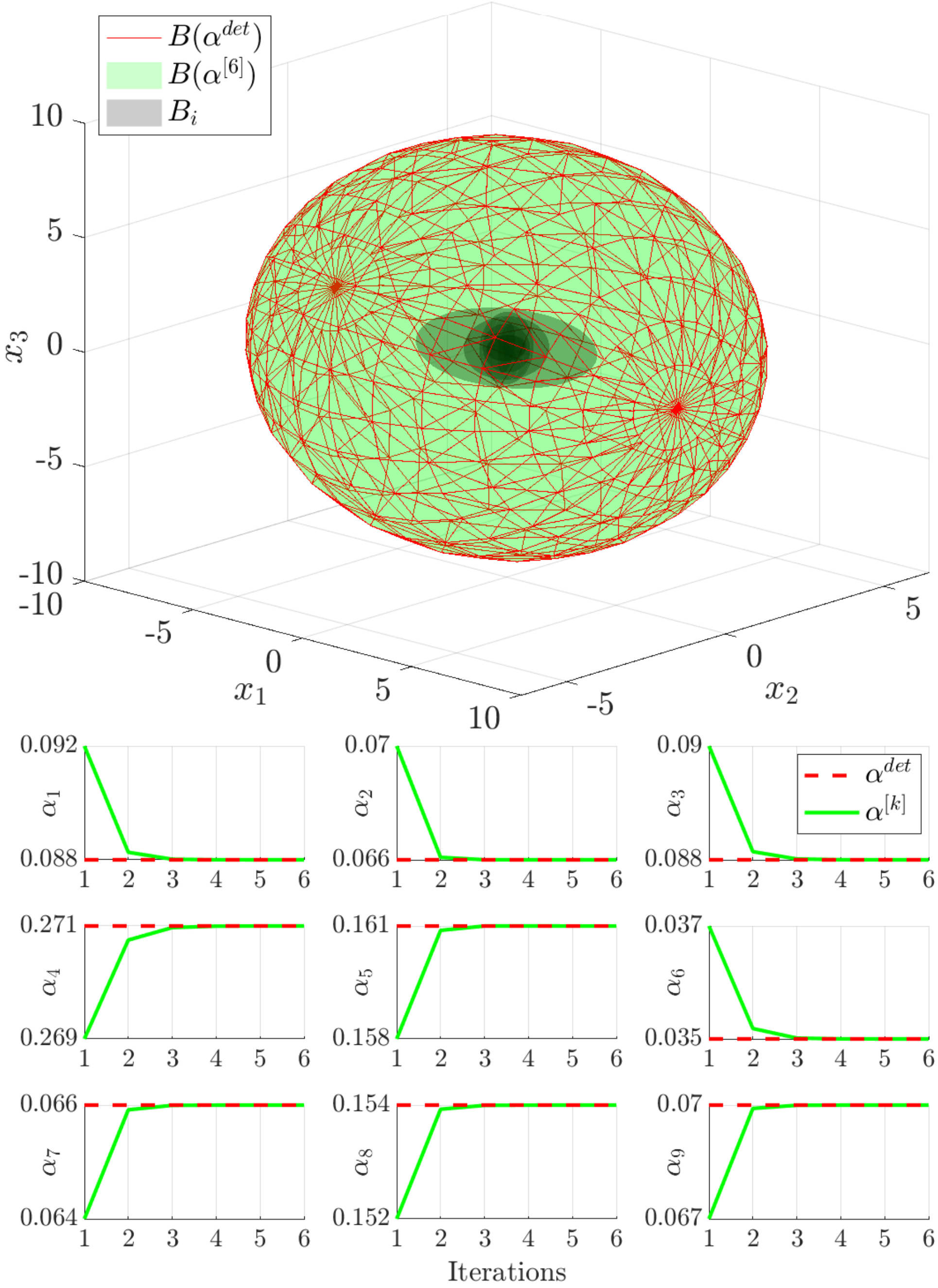


Figure 3.3: Minimum volume composition of 9 ellipsoids in $\mathbb{R}^3$. The wired red ellipsoid is the result of the optimization (3.16), and the shaded green ellipsoid is the result at the 6th iteration of (3.17). The shaded gray regions are the nine randomly generated ellipsoids. The graphs at the bottom show the convergence characteristics of $\alpha^{[k]}$.

as the following recurrence:

$$Q_z(t_0) = Q_0, \quad Q_z(t_j) = Q_z(t_{j-1}) \oplus \bigoplus_{i=1}^{m} Z_i(t_j, t_{j-1}) \quad \forall j \in \{1, \ldots, k\}.$$

As a result, the shape matrix at $t = t_k$ is calculated as

$$Q_z(t_k) = Q_z(t_0) \oplus \bigoplus_{i=1}^{m} \bigoplus_{j=1}^{k} Z_i(t_j, t_{j-1}).$$

Considering the coordinate transform in (2.17), the shape matrix for the FRS of the state-dependent LTV system in (2.9) is

$$Q_x(t_k) = \Phi(t_k, t_0) Q_z(t_k) \Phi(t_k, t_0)^\top.$$

The recurrence for Q_x is given by

$$Q_x(t_0) = Q_0, \quad Q_x(t_j) = \Phi(t_j, t_{j-1}) Q_x(t_{j-1}) \Phi(t_j, t_{j-1})^\top \oplus \bigoplus_{i=1}^{n_w} X_i(t_j, t_{j-1}), \quad \forall j \in \{1, \ldots, k\},$$

where the shape matrix $X_i(t_j, t_{j-1}) \in \mathbb{S}^n_{++}$ is computed as

$$\begin{aligned} X_i(t_j, t_{j-1}) &= \Phi(t_j, t_0) Z_i(t_j, t_{j-1}) \Phi(t_j, t_{j-1})^\top \\ &= (t_j - t_{j-1}) \Phi(t_j, t_{j-1}) N_i(t_j, t_{j-1}) \Phi(t_j, t_{j-1})^\top + \epsilon (t_j - t_{j-1})^2 \Phi(t_j, t_0) \Phi(t_j, t_0)^\top \end{aligned} \tag{3.22}$$

for all $j \in \{1, \ldots, k\}$. Finally, we obtain

$$Q_x(t_k) = \Phi(t_k, t_0) Q_x(t_0) \Phi(t_k, t_0)^\top \oplus \bigoplus_{i=1}^{m} \bigoplus_{j=1}^{k} \Phi(t_k, t_j) X_i(t_j, t_{j-1}) \Phi(t_k, t_j)^\top. \tag{3.23}$$

Note that the composition of ellipsoids must be performed independently at each time step. It means that $Q_x(t_k)$ must be the composition of $1 + m \times k$ shape matrices, and $Q_x(t_{k-1})$ should not be used in the calculation of $Q_x(t_k)$. This is because the proposed minimum volume composition operation is not associative, unlike the Minkowski sum of convex sets as follows:

$$\begin{aligned} \mathcal{E}(Q_1) \oplus \mathcal{E}(Q_2) \oplus \mathcal{E}(Q_3) &= \mathcal{E}(Q_1) \oplus (\mathcal{E}(Q_2) \oplus \mathcal{E}(Q_3)) \,, \\ \mathcal{E}(Q_1 \oplus Q_2 \oplus Q_3) &\subseteq \mathcal{E} \left(Q_1 \oplus (Q_2 \oplus Q_3) \right) , \end{aligned} \tag{3.24}$$

where Q_1, Q_2, and Q_3 are shape matrices of arbitrary ellipsoids. The sequential composition of ellipsoids from $Q_x(t_0)$ to $Q_x(t_k)$ as in (3.24) leads to overly conservative approximation. Algorithm 1 presents the overall process for the computation of $Q_x(t)$, and how to compose the shape matrices over time.

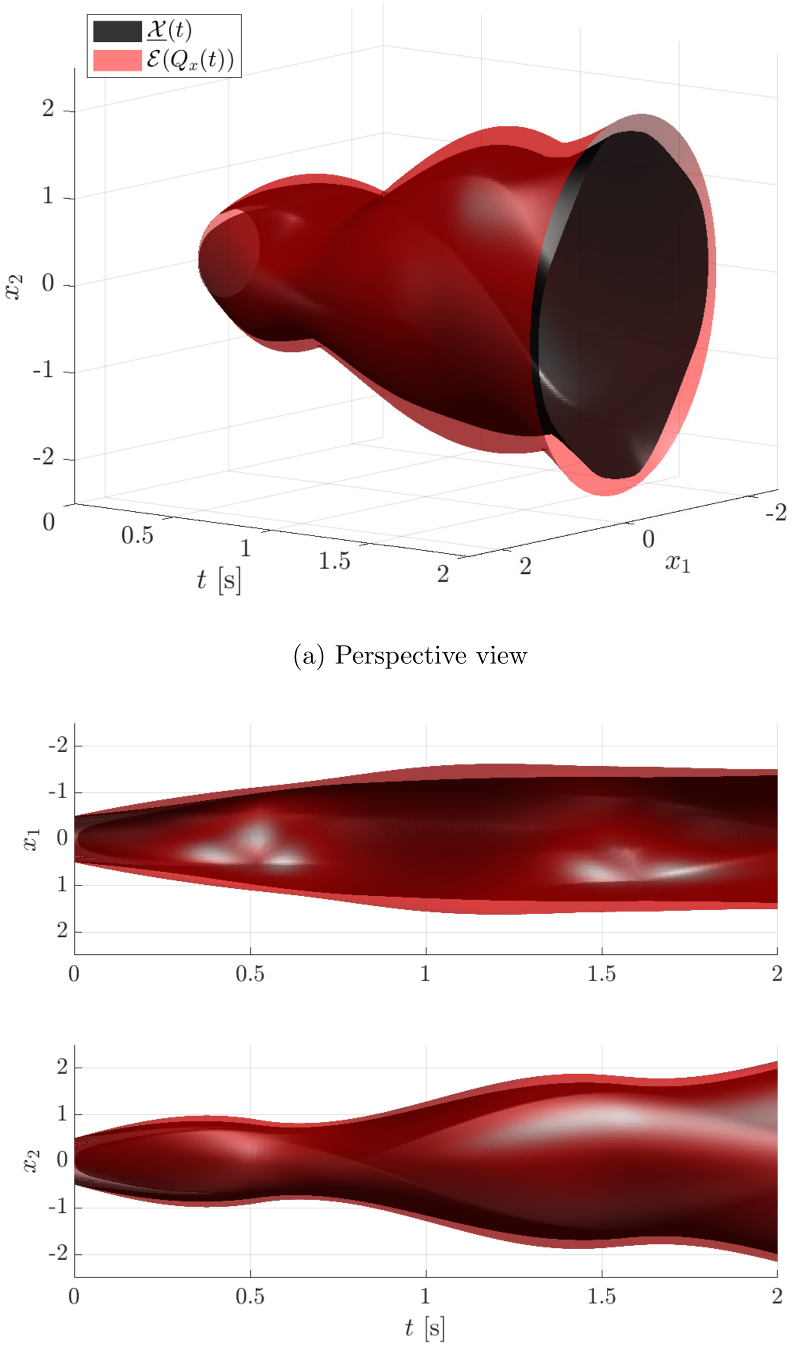
Fig. 3.4 illustrates the comparison between the FRS of a 2-state LTV system, i.e. $\underline{\mathcal{X}}(t)$ in (2.10), and the proposed ellipsoidal approximation $\mathcal{E}(Q_x(t))$ in (3.23). To compute $\underline{\mathcal{X}}(t)$, the solution of the HJB equation associated with the LTV system is computed by the level set toolbox [23]. In the evaluation, $t_0 = 0$ and $t_N = 2$ with 0.01 seconds of time step is used. As expected, the proposed ellipsoidal approximation $\mathcal{E}(Q_x(t))$ certainly encloses the FRS $\underline{\mathcal{X}}(t)$. Fig. 3.5 describes the distribution of the minimum volume composition parameter

Algorithm 1 Computing funnel of LTV systems

```
$Q_x(t_0) = Q_0$
$Q_{basis} \leftarrow \{Q_0\}$
for $k \in \{1, \dots, N-1\}$ do
    Compute system matrices $A_k$ and $D_k$.
    for $i \in \{1, \dots, m\}$ do
        Compute $N_i$ by solving the Lyapunov equation (3.11) using $A_k$ and $D_k$.
        Compute $X_i$ from (3.22) using $N_i$.
        $Q_{basis} \leftarrow Q_{basis} \cup \{X_i\}$
    end for
    $\Phi = \exp(A_k(t_{k+1} - t_k))$
    for $Q \in Q_{basis}$ do
        Transform the shape matrices as $Q = \Phi Q \Phi^\top$.
    end for
    Compute $Q_x(t_{k+1})$ from the minimum volume composition (3.21) using $Q_{basis}$.
end for
Return $\mathcal{F}(t) = \mathcal{E}(Q_x(t))$.
```

α in (3.17). As t_k proceeds, the range of t_j is also stretched, which means an increase in the number of ellipsoids to be composed. The distribution of α depends on the system characteristics. For this particular LTV system, the second channel of disturbance $(i = 2)$ is dominant since the most of the values of α along the second channel are higher than the first channel.

The proposed ellipsoidal approximation of the FRS of LTV systems does not require any computationally burdensome procedure, such as a numerical optimization. So, the proposed method seems to be used for quick computation of the outer approximation of the FRS. However, the ellipsoidal approximation at this stage cannot guarantee the reachability of the nonlinear dynamics since the linearization error is not considered. In the next chapter, methods for addressing the nonlinearity of the system and an algorithm for approximating the FRS of nonlinear systems are presented.



(a) Perspective view

(b) Side views

Figure 3.4: Comparison of the FRS and the corresponding ellipsoidal approximation for a 2-state LTV system: $\dot{x}_1(t) = (-0.8t + 0.5)x_1(t) + \cos(1.5t + 2)x_2(t) + 0.4\cos(t)w_1(t) - 0.4t^2 w_2(t)$, $\dot{x}_2(t) = 0.5t^{2/3}x_1(t) - 2\exp(-0.7t)x_2(t) + 0.08tw_1(t) + 2.8\cos(3t)w_2(t)$ with $Q_0 = \mathrm{diag}(0.25, 0.25)$. The black and the red regions are the FRS of the system and the proposed ellipsoidal approximation respectively.

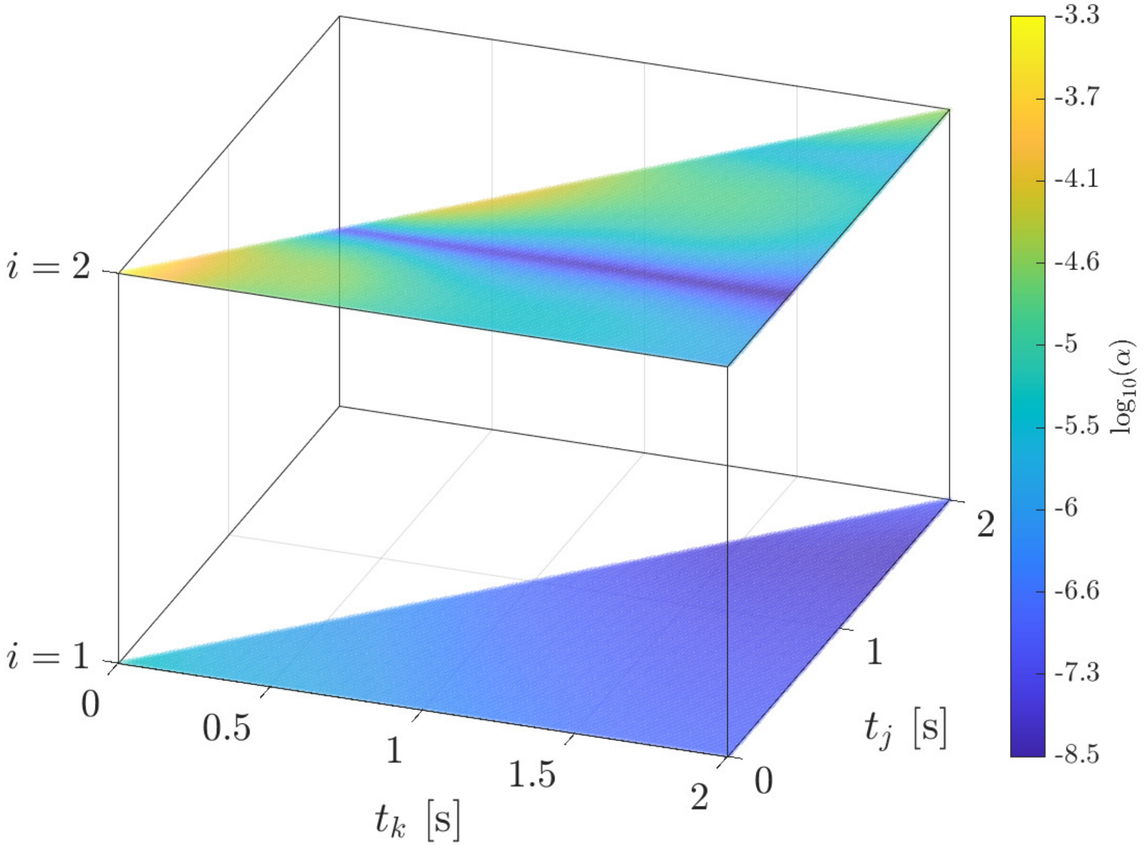


Figure 3.5: Distribution of the coefficients α used in the minimum volume composition (3.23). All computational conditions are identical with the case of Fig. 3.4. The values of α are encoded as color. Each channel of disturbance is separated as the different heights of the colored points.

4

Reachability Analysis of Nonlinear Systems

This chapter concentrates on the computation of the funnel $\mathcal{F}(t)$ in (2.4) of nonlinear systems via polynomial approximation of the value function $V(x,t)$ in (2.5). Section 4.1 presents the approximated (polynomial) value function and formulates a constrained optimization problem to find the funnel that tightly encloses the FRS $\mathcal{X}(t)$ in (2.3). Section 4.2 lists some basic operations on polynomials utilized in the subsequent sections. Based on the property of the Bernstein polynomial, Section 4.3 proposes an approximation of the Hamiltonian and the relaxation of the constraint of the problem formulated in Section 4.2. A numerical algorithm for computing the funnel is presented in Section 4.4. Finally, some discussions and simulation results are presented in Section 4.5.

4.1 Minimal Funnel

In this section, an optimization problem to compute the funnel is presented. Let $\hat{V} : \mathbb{R}^n \times [t_0, t_N] \to \mathbb{R}$ be an approximation of the value function $V(x,t)$. The approximated value function $\hat{V}(x,t)$ must always be less than the value function $V(x,t)$ so that the subzero level set of $\hat{V}(x,t)$ certainly contains the subzero level set of $V(x,t)$, expressed as the following:

$$V(x,t) \geq \hat{V}(x,t) \quad \forall (x,t) \in \mathbb{R}^n \times [t_0, t_N].$$

From Theorem 2.1, the condition for the conservative approximation is

$$\begin{aligned} \frac{\partial \hat{V}(x,t)}{\partial t} + H\Big(x, \frac{\partial \hat{V}(x,t)}{\partial x}, t\Big) \leq 0 \quad & \forall (x,t) \in \mathbb{R}^n \times [t_0, t_N], \\ \hat{V}(x,t) = l(x) \quad & \forall (x,t) \in \mathbb{R}^n \times t_0, \end{aligned} \tag{4.1}$$

where l and H are the initial value function and the Hamiltonian defined in the HJB equation (2.6) respectively. As far as the approximated value function satisfies the condition in (4.1), any set that encloses the subzero level set of $\hat{V}(x,t)$ can be a valid candidate for the funnel. At the same time, the funnel should enclose the FRS as compact as possible to prevent unintended over-approximation. Consequently, the objective of this chapter is to solve the following optimization problem.

Problem 4.1 (Minimal funnel)**.** *For the given initial value function l in* (2.6)*, and the Hamiltonian H in* (2.7)*, find the optimal set $\mathcal{F}(t)$ that satisfies*

$$\begin{aligned} \inf_{\mathcal{F}(t)} \quad & \int_{t_0}^{t_N} \mathrm{Vol}(\mathcal{F}(t))\, dt \\ \text{s.t.} \quad & \frac{\partial \hat{V}(x,t)}{\partial t} + H\Big(x, \frac{\partial \hat{V}(x,t)}{\partial x}, t\Big) \leq 0 && \forall (x,t) \in \mathbb{R}^n \times [t_0, t_N], \\ & \hat{V}(x,t) = l(x) && \forall (x,t) \in \mathbb{R}^n \times t_0, \\ & \{x \mid \hat{V}(x,t) \leq 0\} \subseteq \mathcal{F}(t) && \forall t \in [t_0, t_N], \end{aligned} \tag{4.2}$$

where Vol$(\cdot)$ represents the volume of a set.

Problem 4.1 is not appropriate to solve with numerical methods because the optimization requires to search the optimal function of state and time, namely $\hat{V}(x,t)$. For computational efficiency, $\hat{V}(x,t)$ will be parameterized as a multivariate polynomial of states in Section 4.3. The polynomial approximation of the value function significantly reduces the search space of the optimization (4.2), making it manageable to solve with numerical methods. Consequently, the objective of (4.2) will be converted to search the optimal time-varying coefficient of the polynomial $\hat{V}(x,t)$. In the next section, we present some basic operations on polynomials, which are frequently used in the rest of the thesis.

4.2 Multivariate Bernstein Polynomial

This section presents the main property of a polynomial in a specific form, namely the Bernstein polynomial. Using the multi-index notation, a multivariate monomial of degree $\boldsymbol{i}$ is defined as the following:

$$x^{\boldsymbol{i}} \coloneqq \prod_{j=1}^{n} x_j^{i_j} \in \mathbb{R},$$

where x_j is the j-th element of x, $i_j \in \mathbb{N}_+$ is the j-th element of the n-index $\boldsymbol{i} \in \mathbb{N}_+^n$, and $\mathbb{N}_+$ represents a set of nonnegative integers. The $\boldsymbol{N}$-th order multivariate polynomial $p : \mathbb{R}^n \to \mathbb{R}$ is the weighted sum of monomials

$$p(x) = \sum_{\boldsymbol{i} \leq \boldsymbol{N}} c_{(\boldsymbol{i})} x^{\boldsymbol{i}},$$

where $\boldsymbol{N} \in \mathbb{N}_+^n$ is the maximum degree, $c_{(\boldsymbol{i})} \in \mathbb{R}$ is the $\boldsymbol{i}$-th coefficient of the polynomial p, and the inequality $\boldsymbol{i} \leq \boldsymbol{N}$ between n-indices holds if and only if $i_j \leq N_j$ for all $j \in \{1, \ldots, n\}$. The set of degrees of the $\boldsymbol{N}$-th order polynomial is

$$\mathcal{O}_{\boldsymbol{N}} \coloneqq \{\boldsymbol{i} \mid \boldsymbol{i} \leq \boldsymbol{N}\} \subset \mathbb{N}_+^n.$$

The number of coefficients of the $\boldsymbol{N}$-th order polynomial is denoted as $|\boldsymbol{N}| \coloneqq \mathrm{card}(\mathcal{O}_{\boldsymbol{N}}) = \prod_{j=1}^{n}(N_j + 1) \in \mathbb{N}$, thus $c \in \mathbb{R}^{|\boldsymbol{N}|}$. A basis vector $\phi_{\boldsymbol{N}} : \mathbb{R}^n \to \mathbb{R}^{|\boldsymbol{N}|}$ of the $\boldsymbol{N}$-th order polynomial is defined as

$$\phi_{\boldsymbol{N},(\boldsymbol{i})}(x) \coloneqq x^{\boldsymbol{i}} \quad \forall \boldsymbol{i} \leq \boldsymbol{N},$$

where $\phi_{\boldsymbol{N},(\boldsymbol{i})}(x)$ is the $\boldsymbol{i}$-th element of $\phi_{\boldsymbol{N}}(x)$. Thus, the polynomial p can be expressed as

$$p(x) = \phi_{\boldsymbol{N}}(x)^\top c.$$

Lemma 4.1 (Product of polynomials). *The product of $p_1(x) = \phi_{\boldsymbol{N}_1}(x)^\top c_1$ and $p_2(x) = \phi_{\boldsymbol{N}_2}(x)^\top c_2$ can be expressed as $p(x) = \phi_{\boldsymbol{N}}(x)^\top c$, where $\boldsymbol{N} = \boldsymbol{N}_1 + \boldsymbol{N}_2 \in \mathbb{N}_+^n$, the coefficient vector $c \in \mathbb{R}^{|\boldsymbol{N}|}$ is*

$$c = P_{\boldsymbol{N}_1,\boldsymbol{N}_2}(c_1 \otimes c_2), \tag{4.3}$$

and the (i, j) element of the matrix $P_{\boldsymbol{N}_1,\boldsymbol{N}_2} \in \mathbb{R}^{|\boldsymbol{N}|\times|\boldsymbol{N}_1||\boldsymbol{N}_2|}$ is

$$\left(P_{\boldsymbol{N}_1,\boldsymbol{N}_2}\right)_{ij} = \begin{cases} 1 & \text{if } \left(\phi_{\boldsymbol{N}}(x)\right)_i = \left(\phi_{\boldsymbol{N}_1}(x) \otimes \phi_{\boldsymbol{N}_2}(x)\right)_j, \\ 0 & \text{otherwise,} \end{cases}$$

for all $i \in \{1, \dots, |\boldsymbol{N}|\}$ and $j \in \{1, \dots, |\boldsymbol{N}_1||\boldsymbol{N}_2|\}$.

Remark 4.1. *Since the Kronecker product $c_1 \otimes c_2$ is bilinear, c in (4.3) can be expressed in a linear form such that*

$$c = P_{\boldsymbol{N}_1,\boldsymbol{N}_2}(I_{|\boldsymbol{N}_1|} \otimes c_2)c_1 = P_{\boldsymbol{N}_1,\boldsymbol{N}_2}(c_1 \otimes I_{|\boldsymbol{N}_2|})c_2,$$

whenever either c_1 or c_2 is known.

Lemma 4.2 (Derivative of a polynomial). *The derivative of $p(x) = \phi_{\boldsymbol{N}}(x)^\top c$ with respect to x_j is $\phi_{\boldsymbol{N}}(x)^\top \tilde{c}$, where the coefficient vector $\tilde{c} \in \mathbb{R}^{|\boldsymbol{N}|}$ is*

$$\tilde{c} = D_{\boldsymbol{N},j}c,$$

and $D_{\boldsymbol{N},j} \in \mathbb{R}^{|\boldsymbol{N}|\times|\boldsymbol{N}|}$ is the constant matrix that satisfies

$$\frac{\partial \phi_{\boldsymbol{N}}(x)}{\partial x_j} = D_{\boldsymbol{N},j}^\top \phi_{\boldsymbol{N}}(x) \quad \forall j \in \{1, \dots, n\}.$$

The Bernstein polynomial is a multivariate polynomial of the following form:

$$p(x) = \sum_{\boldsymbol{i} \leq \boldsymbol{N}} b_{(\boldsymbol{i})} \beta_{\boldsymbol{N},(\boldsymbol{i})}(x) = \beta_{\boldsymbol{N}}(x)^\top b,$$

where $b \in \mathbb{R}^{|\boldsymbol{N}|}$ is the Bernstein coefficient, $\beta_{\boldsymbol{N}} : \mathbb{R}^n \to \mathbb{R}^{|\boldsymbol{N}|}$ is the Bernstein basis of the $\boldsymbol{N}$-th order polynomial, $b_{(\boldsymbol{i})}$ and $\beta_{\boldsymbol{N},(\boldsymbol{i})}$ are the $\boldsymbol{i}$-th elements of b and $\beta_{\boldsymbol{N}}$, respectively. In this work, the Bernstein basis defined on $[-1, 1]^n$ is considered such that

$$\beta_{\boldsymbol{N},(\boldsymbol{i})}(x) := \prod_{j=1}^{n} \binom{N_j}{i_j} \frac{1}{2^{N_j}} (1 + x_j)^{i_j} (1 - x_j)^{N_j - i_j} \quad \forall \boldsymbol{i} \leq \boldsymbol{N}.$$

Note that all the elements of the Bernstein basis $\beta_{\boldsymbol{N}}(x)$ is also polynomials of x. Consequently, a linear mapping between coefficients b and c can be constructed.

Lemma 4.3 (Bernstein coefficient of a polynomial)**.** *The Bernstein coefficient of a polynomial $p(x) = \phi_{\boldsymbol{N}}(x)^\top c$ is*

$$b = B_{\boldsymbol{N}} c,$$

where $B_{\boldsymbol{N}} \in \mathbb{R}^{|\boldsymbol{N}| \times |\boldsymbol{N}|}$ is the Bernstein transformation matrix of the $\boldsymbol{N}$-th order polynomial that satisfies

$$\phi_{\boldsymbol{N}}(x) = B_{\boldsymbol{N}}^\top \beta_{\boldsymbol{N}}(x).$$

A key feature of a polynomial in Bernstein form is that the Bernstein coefficient provides lower and upper bounds of the polynomial defined on $[-1, 1]^n$.

Theorem 4.1 (Bounds of a Bernstein polynomial)**.** *For all $x \in [-1, 1]^n$, a polynomial $p(x) = \beta_{\boldsymbol{N}}(x)^\top b$ is bounded below and above as*

$$\min_{x \in [-1,1]^n} p(x) \geq \min_{\boldsymbol{i} \leq \boldsymbol{N}} b_{(\boldsymbol{i})}, \qquad \max_{x \in [-1,1]^n} p(x) \leq \max_{\boldsymbol{i} \leq \boldsymbol{N}} b_{(\boldsymbol{i})}.$$

Proof. It is apparent that $\beta_{\boldsymbol{N},(\boldsymbol{i})}(x) \geq 0$ for all $x \in [-1, 1]^n$ and $\boldsymbol{i} \leq \boldsymbol{N}$. By using the

Binomial theorem, the sum of $\beta_{\boldsymbol{N},(\boldsymbol{i})}(x)$ is

$$\begin{aligned}\sum_{\boldsymbol{i}\le\boldsymbol{N}}\beta_{\boldsymbol{N},(\boldsymbol{i})}(x) &= \sum_{\boldsymbol{i}\le\boldsymbol{N}}\prod_{j=1}^{n}\binom{N_j}{i_j}\frac{1}{2^{N_j}}(1+x_j)^{i_j}(1-x_j)^{N_j-i_j}\\ &= \sum_{i_1=0}^{N_1}\cdots\sum_{i_n=0}^{N_n}\prod_{j=1}^{n}\binom{N_j}{i_j}\Big(\frac{1+x_j}{2}\Big)^{i_j}\Big(\frac{1-x_j}{2}\Big)^{N_j-i_j}\\ &= \prod_{j=1}^{n}\Big(\frac{1+x_j+1-x_j}{2}\Big)^{N_j}\\ &= 1.\end{aligned}$$

Therefore, the following

$$\begin{aligned}p(x) &= \sum_{\boldsymbol{i}\le\boldsymbol{N}} b_{(\boldsymbol{i})}\beta_{\boldsymbol{N},(\boldsymbol{i})}(x) \ge \Big(\min_{\boldsymbol{i}\le\boldsymbol{N}} b_{(\boldsymbol{i})}\Big)\sum_{\boldsymbol{i}\le\boldsymbol{N}}\beta_{\boldsymbol{N},(\boldsymbol{i})}(x) = \min_{\boldsymbol{i}\le\boldsymbol{N}} b_{(\boldsymbol{i})}\\ p(x) &= \sum_{\boldsymbol{i}\le\boldsymbol{N}} b_{(\boldsymbol{i})}\beta_{\boldsymbol{N},(\boldsymbol{i})}(x) \le \Big(\max_{\boldsymbol{i}\le\boldsymbol{N}} b_{(\boldsymbol{i})}\Big)\sum_{\boldsymbol{i}\le\boldsymbol{N}}\beta_{\boldsymbol{N},(\boldsymbol{i})}(x) = \max_{\boldsymbol{i}\le\boldsymbol{N}} b_{(\boldsymbol{i})}\end{aligned}$$

holds for all $x \in [-1,1]^n$. □

From Theorem 4.1, a conservative approximation of the minimum and maximum of a polynomial $p(x)$ defined on $[-1,1]^n$ can be obtained. However, not all polynomials are defined on $[-1,1]^n$. The following lemma presents that any polynomial in an arbitrary rectangular domain can be represented in $[-1,1]^n$.

Lemma 4.4 (Domain transformation of a polynomial)**.** *Let $\sigma : \mathbb{R}^n \to \mathbb{R}^n$ be a linear transformation from $v \in [-1,1]^n$ to x such that $x = \sigma(v) = Q^{\frac{1}{2}}v$, where $Q \in \mathbb{S}^n_{++}$ is the given shape matrix of the rectangular region defined as*

$$\mathcal{R}(Q) := \{Q^{\frac{1}{2}}v \mid v \in [-1,1]^n\} \subset \mathbb{R}^n.$$

Then, for all $x \in \mathcal{R}(Q)$, a polynomial $p(x) = \phi_{\boldsymbol{N}}(x)^\top c$ can be expressed as $p(\sigma(v)) =$

$\phi_{\bar{\boldsymbol{N}}}(v)^\top \tilde{c}$, where $\bar{\boldsymbol{N}} \coloneqq (\sum_{j=1}^n N_j)\mathbf{1}^n \in \mathbb{N}_+^n$,

$$\tilde{c} = T_{\boldsymbol{N}}(Q)c \in \mathbb{R}^{|\bar{\boldsymbol{N}}|},$$

and $T_{\boldsymbol{N}} : \mathbb{S}_{++}^n \to \mathbb{R}^{|\bar{\boldsymbol{N}}|\times|\boldsymbol{N}|}$ is the domain transformation matrix that satisfies

$$\phi_{\boldsymbol{N}}(\sigma(v)) = T_{\boldsymbol{N}}(Q)^\top \phi_{\bar{\boldsymbol{N}}}(v).$$

By combining Lemmas 4.3 and 4.4, the lower and upper bounds of a polynomial in a finite domain can be obtained, as a direct consequence of Theorem 4.1.

Corollary 4.1 (Bounds of a polynomial in a finite domain)**.** *For a given shape matrix $Q \in \mathbb{S}_{++}^n$, a polynomial $p(x) = \phi_{\boldsymbol{N}}(x)^\top c$ is bounded below and above in $\mathcal{R}(Q)$ as*

$$\min_{x\in\mathcal{R}(Q)} p(x) \geq \min_{\boldsymbol{i}\leq\bar{\boldsymbol{N}}} b_{(\boldsymbol{i})}, \quad \max_{x\in\mathcal{R}(Q)} p(x) \leq \max_{\boldsymbol{i}\leq\bar{\boldsymbol{N}}} b_{(\boldsymbol{i})},$$

where $b_{(\boldsymbol{i})}$ is the $\boldsymbol{i}$-th element of $b = B_{\bar{\boldsymbol{N}}} T_{\boldsymbol{N}}(Q)c \in \mathbb{R}^{|\bar{\boldsymbol{N}}|}$.

The Bernstein coefficient of a polynomial is useful to handle inequality constraints related to polynomials. Suppose finding an admissible set of coefficients of a polynomial $p(x) = \phi_{\boldsymbol{N}}(x)^\top c$ such that $p(x) \leq 0$ for all $x \in \mathcal{R}(Q)$. From Corollary 4.1, the following implication

$$B_{\bar{\boldsymbol{N}}} T_{\boldsymbol{N}}(Q)c \leq 0 \implies p(x) \leq 0 \quad \forall x \in \mathcal{R}(Q) \tag{4.4}$$

is valid. Consequently, polynomial inequalities on a finite domain can be easily converted to linear inequalities of the corresponding coefficient. This implication will be thoroughly used for the proposed funnel computation algorithm, as discussed in the following section.

4.3 Approximation of Hamiltonian

This section proposes a polynomial approximation of the Hamiltonian H in (2.7) that is the component of the inequality constraint of Problem 4.1. To exploit the useful property of the Bernstein polynomial presented in Section 4.2, the following assumption on the structure of the dynamics is presented.

Assumption 4.1 (Polynomial dynamics)**.** *The dynamics $f : \mathbb{R}^n \times \mathbb{R}^m \times [t_0, t_N] \to \mathbb{R}^n$ can be expressed as polynomials of states and disturbances such that*

$$f_j(x(t), w(t), t) = \phi_{\boldsymbol{N}_w}(w)^\top S_j(t)^\top \phi_{\boldsymbol{N}_x}(x) \quad \forall j \in \{1, \ldots, n\}, \tag{4.5}$$

where f_j is the j-th element of f, $S_j : [t_0, t_N] \to \mathbb{R}^{|\boldsymbol{N}_x| \times |\boldsymbol{N}_w|}$ is the time-varying coefficient matrix of f_j, $\boldsymbol{N}_x \in \mathbb{N}_+^n$ and $\boldsymbol{N}_w \in \mathbb{N}_+^m$ are the maximum degree of f_j in x and w, respectively.

Remark 4.2. *Assumption 1 is not highly restrictive because high-order (usually 3 or greater) Taylor expansion of the dynamics along the nominal state trajectory can appropriately capture the nonlinearities of the system in practice [8]. Also, the small-angle assumption of trigonometric functions, which is widely used for multirotor systems [33], can also simplify the nonlinear dynamics to a polynomial of states.*

Consider an $\boldsymbol{N}$-th order polynomial of states that parameterizes the approximated value function such that

$$\hat{V}(x, t) = \phi_{\boldsymbol{N}}(x)^\top c(t), \tag{4.6}$$

where $c : [t_0, t_N] \to \mathbb{R}^{|\boldsymbol{N}|}$ is the time-varying coefficient of $\hat{V}(x, t)$. Assuming that the initial value function $l(x)$ in (2.6) is a polynomial of states, $c_0 := c(t_0)$ is determined as

$$l(x) = \phi_{\boldsymbol{N}}(x)^\top c_0.$$

Proposition 4.1 (Approximation of the Hamiltonian)**.** *For the given polynomial dynamics f in (4.5) and approximated value function $\hat{V}$ in (4.6), the Hamiltonian in (2.7) is bounded* *above with the following $|\boldsymbol{N}_w|$ linear functions of c such that*

$$H\Big(x, \frac{\partial \hat{V}(x,t)}{\partial x}, t\Big) \leq \phi_{\boldsymbol{N}+\boldsymbol{N}_x}(x)^\top H_{\boldsymbol{m}}(t)c(t) \quad \forall \boldsymbol{m} \leq \boldsymbol{N}_w, \tag{4.7}$$

where $H_{\boldsymbol{m}} : [t_0, t_N] \to \mathbb{R}^{|\boldsymbol{N}+\boldsymbol{N}_x| \times |\boldsymbol{N}|}$ is a time-varying matrix.

Proof. From (2.7), the Hamiltonian is

$$H\Big(x, \frac{\partial \hat{V}(x,t)}{\partial x}, t\Big) = \max_{w \in \mathcal{W}} \Big(\sum_{j=1}^{n} \frac{\partial \hat{V}(x,t)}{\partial x_j} f_j(x,w,t) \Big).$$

By using Lemmas 4.1 and 4.2, the argument of the maximization in the above equation is

$$\frac{\partial \hat{V}(x,t)}{\partial x_j} f_j(x,w,t) = \phi_{\boldsymbol{N}_w}(w)^\top \tilde{c}_j(x,t) \quad \forall j \in \{1, \ldots, n\}, \tag{4.8}$$

where the $\boldsymbol{m}$-th element of $\tilde{c}_j : \mathbb{R}^n \times [t_0, t_N] \to \mathbb{R}^{|\boldsymbol{N}_w|}$ is the polynomial of x such that

$$\begin{aligned}\tilde{c}_{j,(\boldsymbol{m})}(x,t) &= (\phi_{\boldsymbol{N}}(x)^\top D_{\boldsymbol{N},j} c(t))(\phi_{\boldsymbol{N}_x}(x)^\top S_{j,\boldsymbol{m}}(t)) \\ &= \phi_{\boldsymbol{N}+\boldsymbol{N}_x}(x)^\top P_{\boldsymbol{N},\boldsymbol{N}_x} \big(I_{|\boldsymbol{N}|} \otimes S_{j,\boldsymbol{m}}(t)\big) D_{\boldsymbol{N},j} c(t),\end{aligned}$$

and $S_{j,\boldsymbol{m}} : [t_0, t_N] \to \mathbb{R}^{|\boldsymbol{N}_x|}$ is the $\boldsymbol{m}$-th column of S_j. The summation of (4.8) for all $j \in \{1, \ldots, n\}$ yields

$$\frac{\partial \hat{V}(x,t)}{\partial x} \cdot f(x,w,t) = \phi_{\boldsymbol{N}_w}(w)^\top \tilde{c}(x,t), \tag{4.9}$$

where the $\boldsymbol{m}$-th element of $\tilde{c} : \mathbb{R}^n \times [t_0, t_N] \to \mathbb{R}^{|\boldsymbol{N}_w|}$ is

$$\tilde{c}_{(\boldsymbol{m})}(x,t) = \phi_{\boldsymbol{N}+\boldsymbol{N}_x}(x)^\top P_{\boldsymbol{N},\boldsymbol{N}_x} \Big(\sum_{j=1}^{n} \big(I_{|\boldsymbol{N}|} \otimes S_{j,\boldsymbol{m}}(t)\big) D_{\boldsymbol{N},j} \Big) c(t).$$

Using Lemma 4.3, the Bernstein transform of (4.9) is

$$\phi_{\boldsymbol{N}_w}(w)^\top \tilde{c}(x,t) = \beta_{\boldsymbol{N}_w}(w)^\top B_{\boldsymbol{N}_w} \tilde{c}(x,t) = \beta_{\boldsymbol{N}_w}(w)^\top \tilde{b}(x,t),$$

where $\boldsymbol{m}$-th element of $\tilde{b} : \mathbb{R}^n \times [t_0, t_N] \to \mathbb{R}^{|\boldsymbol{N}_w|}$ is

$$\tilde{b}_{(\boldsymbol{m})}(x,t) = \sum_{\boldsymbol{i} \le \boldsymbol{N}_w} (B_{\boldsymbol{N}_w})_{\boldsymbol{m}\boldsymbol{i}}\, \tilde{c}_{(\boldsymbol{i})}(x,t) = \phi_{\boldsymbol{N}+\boldsymbol{N}_x}(x)^\top H_{\boldsymbol{m}}(t) c(t),$$

$(B_{\boldsymbol{N}_w})_{\boldsymbol{m}\boldsymbol{i}}$ is the $(\boldsymbol{m}, \boldsymbol{i})$ element of $B_{\boldsymbol{N}_w}$, and $H_{\boldsymbol{m}}$ is defined as

$$H_{\boldsymbol{m}}(t) := \sum_{\boldsymbol{i} \le \boldsymbol{N}_w} (B_{\boldsymbol{N}_w})_{\boldsymbol{m}\boldsymbol{i}}\, P_{\boldsymbol{N},\boldsymbol{N}_x} \Big(\sum_{j=1}^{n} \big(I_{|\boldsymbol{N}|} \otimes S_{j,\boldsymbol{m}}(t) \big) D_{\boldsymbol{N},j} \Big) \tag{4.10}$$

for all $\boldsymbol{m} \le \boldsymbol{N}_w$. Based on Assumption 3.1 and Theorem 4.1, the Hamiltonian is bounded above with the corresponding Bernstein coefficients such that

$$H\Big(x, \frac{\partial \hat{V}(x,t)}{\partial x}, t\Big) \le \max_{\boldsymbol{m} \le \boldsymbol{N}_w} \tilde{b}_{(\boldsymbol{m})}(x,t),$$

which concludes the proof. □

Remark 4.3. *Since the Hamiltonian is approximated by a conservative manner as in* (4.7)*, the following implication*

$$\begin{aligned} &\frac{\partial \hat{V}(x,t)}{\partial t} + \phi_{\boldsymbol{N}+\boldsymbol{N}_x}(x)^\top H_{\boldsymbol{m}}(t) c(t) \le 0 \quad \forall \boldsymbol{m} \le \boldsymbol{N}_w \\ \Longrightarrow\ &\frac{\partial \hat{V}(x,t)}{\partial t} + H\Big(x, \frac{\partial \hat{V}(x,t)}{\partial x}, t\Big) \le 0 \end{aligned} \tag{4.11}$$

is valid for all $(x,t) \in \mathbb{R}^n \times [t_0, t_N]$.

4.4 Funnel Computation

This section presents an LP-based numerical algorithm to solve Problem 4.1 based on the findings in Sections 4.2 and 4.3. Instead of computing the approximated value function that is valid for the entire domain $\mathbb{R}^n$, this work searches for a locally valid approximation such that

$$\frac{\partial \hat{V}(x,t)}{\partial t} + H\Big(x, \frac{\partial \hat{V}(x,t)}{\partial x}, t\Big) \leq 0 \quad \forall (x,t) \in \mathcal{R}(Q(t)) \times [t_0, t_N], \tag{4.12}$$

where $Q : [t_0, t_N] \to \mathbb{S}^n_{++}$ is the shape matrix of the domain of interest $\mathcal{R}$, respectively. The locally valid approximation in (4.12) is introduced to utilize the implication (4.4) in the subsequent derivation. Note that the FRS is the subzero level set of the value function, so it is enough to approximate the value function on the finite domain, provided that the region $\mathcal{R}$ is sufficiently close to the FRS. Moreover, the value function can further be approximated for the entire state space by using the subdivision method [2], which is not the scope of the current work.

Also, to represent the funnel $\mathcal{F}(t)$ as an ellipsoid, a quadratic function is introduced such that

$$\tilde{V}(x,t) \coloneqq x^\top R(t) x + \rho(t), \tag{4.13}$$

where $R : [t_0, t_N] \to \mathbb{S}^n_{++}$, and $\rho : [t_0, t_N] \to \mathbb{R}$. Then, the subzero level set of $\tilde{V}(x,t)$ can be the funnel $\mathcal{F}(t)$, namely

$$\mathcal{F}(t) = \mathcal{E}(-\rho(t) R(t)^{-1}).$$

Since a quadratic function of states is also a polynomial of states, $\tilde{V}(x,t)$ can also be expressed as $\phi_{\boldsymbol{N}}(x)^\top E_{\boldsymbol{N}} \tilde{c}(t)$, where $\tilde{c} : [t_0, t_N] \to \mathbb{R}^{\frac{n^2+n+2}{2}}$ is the time-varying coefficient vector such that

$$\tilde{c}(t) = [\rho(t), \mathrm{vech}(R(t))^\top]^\top,$$

vech : $\mathbb{S}^n_{++} \to \mathbb{R}^{\frac{n(n+1)}{2}}$ is the half-vectorization operator for symmetric matrices, and $E_{\boldsymbol{N}} \in \mathbb{R}^{|\boldsymbol{N}| \times \frac{n^2+n+2}{2}}$ is the constant matrix that satisfies

$$\phi_{\boldsymbol{N}}(x)^\top E_{\boldsymbol{N}} \tilde{c}(t) = x^\top R(t) x + \rho(t).$$

Consequently, Problem 4.1 is converted to a computationally tractable form as follows.

$$\begin{aligned}
\max_{c(t),\tilde{c}(t)} \quad & \int_{t_0}^{t_N} \Big(\int_{\mathcal{R}(Q(t))} \tilde{V}(x,t)\, dx \Big)\, dt \\
\text{s.t.} \quad & \frac{\partial \hat{V}(x,t)}{\partial t} + H\Big(x, \frac{\partial \hat{V}(x,t)}{\partial x}, t\Big) \leq 0 \qquad \forall (x,t) \in \mathcal{R}(Q(t)) \times [t_0, t_N], \\
& \tilde{V}(x,t) - \hat{V}(x,t) \leq 0 \qquad \forall (x,t) \in \mathcal{R}(Q(t)) \times [t_0, t_N],
\end{aligned} \tag{4.14}$$

where the cost function in (4.2), i.e. the minimization of $\mathrm{Vol}(\mathcal{F}(t))$, is replaced by the maximization of the integral of $\tilde{V}(x,t)$ over $\mathcal{R}(Q(t))$ because the two criteria have similar meaning. For computational efficiency, the following assumption is used.

Assumption 4.2 (Time discretization)**.** *Let $t_k \in [t_0, t_N]$ be one of $N+1$ evenly discretized time indices, where $t_k < t_{k+1}$ for all $k \in \{0, \ldots, N-1\}$. For a sufficiently small interval $[t_k, t_{k+1})$,*

$$\frac{\partial \hat{V}(x,t_k)}{\partial t} = \frac{\hat{V}_{k+1}(x) - \hat{V}_k(x)}{\delta t},$$

where $\hat{V}_k(x) := \hat{V}(x,t_k)$ for all $k \in \{0, \ldots, N\}$, and $\delta t \in \mathbb{R}$ is the discretized time step.

Note that the converted optimization problem (4.14) can be divided into two subsequent problems: the polynomial approximation for determining $c(t)$, and the quadratic approximation for determining $\tilde{c}(t)$ with respect to the given $c(t)$. This is because it is possible to find the optimal $c(t)$ regardless of $\tilde{c}(t)$ by redesigning the cost function of (4.14) with respect to $c(t)$. The details of the optimization procedure are covered in the following subsections.

4.4.1 Polynomial Approximation

Suppose that the initial guess of the domain $\mathcal{R}(Q(t))$ is given for all $t \in [t_0, t_N]$. For visual clarity, $Q_k := Q(t_k) \in \mathbb{S}^n_{++}$ is used for all $k \in \{0, \ldots, N\}$. Considering the given sequence of $\mathcal{R}(Q_k)$ and Lemma 4.4, a cost function of the polynomial approximation problem is designed as the summation of the integral of $\hat{V}_k(x)$ over the domain $\mathcal{R}(Q_k)$ for all $k \in \{1, \dots, N\}$:

$$\begin{aligned}\sum_{k=1}^{N} \int_{\mathcal{R}(Q_k)} \hat{V}_k(x)\; dx &= \sum_{k=1}^{N} \int_{[-1,1]^n} \phi_{\bar{\boldsymbol{N}}}(v)^\top T_{\boldsymbol{N}}(Q_k) c_k \; dv \\ &= h_{\bar{\boldsymbol{N}}}^\top \Big(\sum_{k=1}^{N} T_{\boldsymbol{N}}(Q_k) c_k \Big), \end{aligned} \tag{4.15}$$

where $c_k \coloneqq c(t_k)$, and the $\boldsymbol{i}$-th element of the constant vector $h_{\bar{\boldsymbol{N}}} \in \mathbb{R}^{|\bar{\boldsymbol{N}}|}$ is defined as

$$h_{\bar{\boldsymbol{N}},(\boldsymbol{i})} \coloneqq \int_{[-1,1]^n} \phi_{\bar{\boldsymbol{N}},(\boldsymbol{i})}(v)\; dv \quad \forall \boldsymbol{i} \leq \bar{\boldsymbol{N}}.$$

Now, consider the inequality constraint in (4.12). By using Assumption 4.2 and the implication (4.11) of Proposition 4.1, the sufficient condition of (4.12) is given as

$$\phi_{\boldsymbol{N}+\boldsymbol{N}_x}(x)^\top \big(C_{\boldsymbol{N},\boldsymbol{N}_x}(c_{k+1} - c_k) + \delta t H_{\boldsymbol{m}}(t_k) c_k \big) \leq 0 \tag{4.16}$$

for all $x \in \mathcal{R}(Q_k)$, $\boldsymbol{m} \leq \boldsymbol{N}_w$, and $k \in \{0, \dots, N-1\}$, where $C_{\boldsymbol{N},\boldsymbol{N}_x} \in \mathbb{R}^{|\boldsymbol{N}+\boldsymbol{N}_x| \times |\boldsymbol{N}|}$ is the constant matrix that satisfies

$$\phi_{\boldsymbol{N}}(x) = C_{\boldsymbol{N},\boldsymbol{N}_x}^\top \phi_{\boldsymbol{N}+\boldsymbol{N}_x}(x).$$

From the implication in (4.4), the inequality (4.16) is valid only if

$$B_{\overline{\boldsymbol{N}+\boldsymbol{N}_x}} T_{\boldsymbol{N}+\boldsymbol{N}_x}(Q_k) \big(C_{\boldsymbol{N},\boldsymbol{N}_x}(c_{k+1} - c_k) + \delta t H_{\boldsymbol{m}}(t_k) c_k \big) \leq 0$$

for all $\boldsymbol{m} \leq \boldsymbol{N}_w$ and $k \in \{0, \dots, N-1\}$. Consequently, the following LP is formulated:

Problem 4.2 (Polynomial approximation)**.** *Let f be the polynomial dynamics in* (4.5)*, and $H_{\boldsymbol{m}}$ be the time-varying matrix defined in* (4.10)*. For the given initial coefficients $c_0 \in \mathbb{R}^{|\boldsymbol{N}|}$ and the sequence of the domain $\mathcal{R}(Q_k)$ for all $k \in \{0, \dots, N\}$, find the optimal coefficient*

c_k of the polynomial value function $\hat{V}_k(x) = \phi_{\boldsymbol{N}}(x)^\top c_k$ that satisfies

$$\begin{aligned} \max_{c_1,\ldots,c_N} \quad & h_{\bar{\boldsymbol{N}}}^\top\Big(\sum_{k=1}^{N} T_{\boldsymbol{N}}(Q_k)c_k\Big) \\ \text{s.t.} \quad & B_{\overline{\boldsymbol{N}+\boldsymbol{N}_x}} T_{\boldsymbol{N}+\boldsymbol{N}_x}(Q_k)\big(C_{\boldsymbol{N},\boldsymbol{N}_x}(c_{k+1}-c_k)+\delta t H_{\boldsymbol{m}}(t_k)c_k\big) \le 0 \\ & \qquad\qquad\qquad\qquad \forall \boldsymbol{m} \le \boldsymbol{N}_w \quad \forall k \in \{0,\ldots,N-1\}. \end{aligned} \tag{4.17}$$

4.4.2 Quadratic Approximation

The next step is to find the optimal quadratic approximation $\tilde{V}(x,t)$ in (4.13) with respect to the given $\hat{V}_k(x)$ computed from Problem 4.2 for all $k \in \{1,\ldots,N\}$. As can be seen in (4.14), the constraint related with $\tilde{V}(x,t)$ does not involve time-dependent terms. So, it is possible to compute the optimal $\tilde{c}_k \coloneqq \tilde{c}(t_k)$ for each $k \in \{1,\ldots,N\}$. Similar to (4.15), a cost function of the quadratic approximation for the given domain $\mathcal{R}(Q_k)$ is designed as

$$\int_{\mathcal{R}(Q_k)} \tilde{V}_k(x)\, dx = h_{\bar{\boldsymbol{N}}}^\top T_{\boldsymbol{N}}(Q_k) E_{\boldsymbol{N}} \tilde{c}_k,$$

where $\tilde{V}_k(x) \coloneqq \tilde{V}(x,t_k)$. Also, by using the implication in (4.4), a linear inequality constraint that ensures $\tilde{V}_k(x) - \hat{V}_k(x) \le 0$ for all $x \in \mathcal{R}(Q_k)$ can be constructed as

$$B_{\bar{\boldsymbol{N}}} T_{\boldsymbol{N}}(Q_k)(E_{\boldsymbol{N}}\tilde{c}_k - c_k) \le 0.$$

Consequently, the following simple LP is proposed to find the optimal quadratic function:

Problem 4.3 (Quadratic approximation)**.** *For the given coefficient c_k computed in* (4.17) *and the domain $\mathcal{R}(Q_k)$ for all $k \in \{1,\ldots,N\}$, find the optimal coefficient $\tilde{c}_k$ of the quadratic function $\tilde{V}_k(x) = \phi_{\boldsymbol{N}}(x)^\top E_{\boldsymbol{N}}\tilde{c}_k$ that satisfies*

$$\begin{aligned} \max_{\tilde{c}_k} \quad & h_{\bar{\boldsymbol{N}}}^\top T_{\boldsymbol{N}}(Q_k) E_{\boldsymbol{N}} \tilde{c}_k \\ s.t. \quad & B_{\bar{\boldsymbol{N}}} T_{\boldsymbol{N}}(Q_k)(E_{\boldsymbol{N}}\tilde{c}_k - c_k) \le 0, \end{aligned} \tag{4.18}$$

for each $k \in \{1, \ldots, N\}$.

After optimizing $\tilde{c}_k$ from Problem 4.3 for all $k \in \{1, \ldots, N\}$, the following ellipsoid can be computed as the subzero level set of the quadratic value function $\tilde{V}_k(x)$:

$$\{x \mid \tilde{V}_k(x) \leq 0\} = \mathcal{E}\big(-\rho_k R_k^{-1}\big),$$

where $R_k \coloneqq R(t_k)$, and $\rho_k \coloneqq \rho(t_k)$. The previously given domain $\mathcal{R}(Q_k)$ for all $k \in \{1, \ldots, N\}$ can be updated as

$$Q_k = -\rho_k R_k^{-1}. \tag{4.19}$$

As the domain of interest $\mathcal{R}(Q_k)$ for all $k \in \{1, \ldots, N\}$ is renewed, Problems 4.2 and 4.3 are solved again. Those processes are iterated until the sum of the volume of ellipsoids, i.e. $\sum_{k=1}^{N} \log(\det(Q_k))$, converges. After the iteration finishes, the resulting ellipsoid $\mathcal{E}(Q_k)$ is used as the funnel $\mathcal{F}_k \coloneqq \mathcal{F}(t_k)$ for all $k \in \{1, \ldots, N\}$. Algorithm 2 presents the overall process of the proposed funnel computation procedure.

Algorithm 2 Computing funnel of nonlinear systems

```
Initialize the domain $\mathcal{R}(Q_k)$.
$converged = false$
$cost_{prev} = \sum_{k=1}^{N} \log(\det(Q_k))$
while $\neg converged$ do
    Compute $c_k$ by solving LP (4.17) using $Q_k$.
    Compute $\tilde{c}_k$ by solving LP (4.18) using $c_k$, and $Q_k$.
    Update $Q_k$ from (4.19).
    $cost = \sum_{k=1}^{N} \log(\det(Q_k))$
    $dcost = (cost_{prev} - cost)/cost_{prev}$
    if $|dcost| < \epsilon$ then
        $converged = true$
    end if
    $cost_{prev} = cost$
end while
Return $\mathcal{F}_k = \mathcal{E}(Q_k)$.
```

4.4.3 Initial Guess of the Domain

Until now, the initial guess of the domain of interest $\mathcal{R}(Q(t))$ for all $t \in [t_0, t_N]$ has been assumed to be given in advance. However, the quality of the initial guess of the domain is crucial for how fast the iterative computation converges. Suppose that the size of $\mathcal{R}(Q(t))$ is much larger than the FRS of the system. It means that the polynomial value function $\hat{V}(x,t)$ at the initial step covers the unnecessary region that will be contracted as the iteration proceeds. Contrary, when the size of $\mathcal{R}(Q(t))$ is much smaller than the FRS of the system, the polynomial value function $\hat{V}(x,t)$ at the initial step cannot fully capture the FRS of the system. So, it takes extra iterations until the polynomial value function $\hat{V}(x,t)$ covers the FRS of the system. In this work, the ellipsoidal approximation of the FRS of the linearized system, i,e, $\mathcal{E}(Q_x(t))$ in (3.23), is utilized as the initial guess of the domain $\mathcal{R}(Q(t))$. Although the ellipsoidal approximation of the FRS of LTV systems cannot enclose the FRS of nonlinear systems, it is sufficiently close to the FRS of nonlinear systems, and thus appropriate for the initial guess of $Q(t)$.

4.5 Simulation Results and Discussion

This section presents the simulation results of the funnel computation for various examples. The proposed algorithm is implemented in MATLAB R2020b, and CVX [34] is used for solving the LP. We also implement the proposed LP formulation in C++ with CPLEX running on Ubuntu 18.04. For comparison, we compute the funnel by SOS program, following the formulation presented in [8]. We use SOSTOOLS [35] to convert the SOS program to a semi-definite program (SDP), and SeDuMi [36] is used for solving the SDP. We also solve the HJB equation with a numerical solver [23] to compare the funnels and the FRS of the system. The comparison of computation times is summarized in Table 4.1. All the computation times in Table 4.1 are measured in a machine with eight processors, 2.3 GHz of CPU, and 16 GB of RAM.

4.5.1 Lotka-Volterra

The proposed funnel computation algorithm is validated for the following nonlinear system:

$$\begin{aligned}\dot{x}_1(t) &= (0.1w(t)+3)x_1(t)(1-x_2(t)),\\ \dot{x}_2(t) &= (0.1w(t)+3)x_2(t)(x_1(t)-1).\end{aligned} \tag{4.20}$$

The system (4.20) is polynomial of states and disturbance, with $\boldsymbol{N}_x = [1,1]^\top$ and $\boldsymbol{N}_w = 1$. For the polynomial approximation of the value function, $\boldsymbol{N} = [3,3]^\top$ is used. The funnel is computed for 1 second with evenly discretized 50 time segments, i.e. $N = 50$ and $\delta t = 0.02$ seconds. The initial value function is $l(x) = x^\top Q(t_0)^{-1}x - 1$, where $Q(t_0) = \text{diag}(0.05, 0.05)^2$. Also, to solve the HJB equation, each state space is gridded by 401 points and thus a total of 160801 grid points are used for the computation of the FRS.

Fig. 4.1 presents the comparison between the funnels computed by the proposed method and SOS program. Both of the funnels are good outer approximations of the FRS, as can be seen in Fig. 4.1 where the green ellipsoids and the shaded red surface enclose the shaded gray surface. The proposed method and the SOS program-based method yield fairly similar results; however, the proposed method is much faster than the SOS program-based method, as supported by Table 4.1.

Cases	Level set [23]	SOS program[8]	Proposed	
			MATLAB	C++
Lotka-Volterra	391.2246 s	108.9980 s	2.1813 s	0.0615 s
Unicycle	1821.6029 s	295.5570 s	9.5814 s	0.4571 s

Table 4.1: Comparison of the funnel computation time

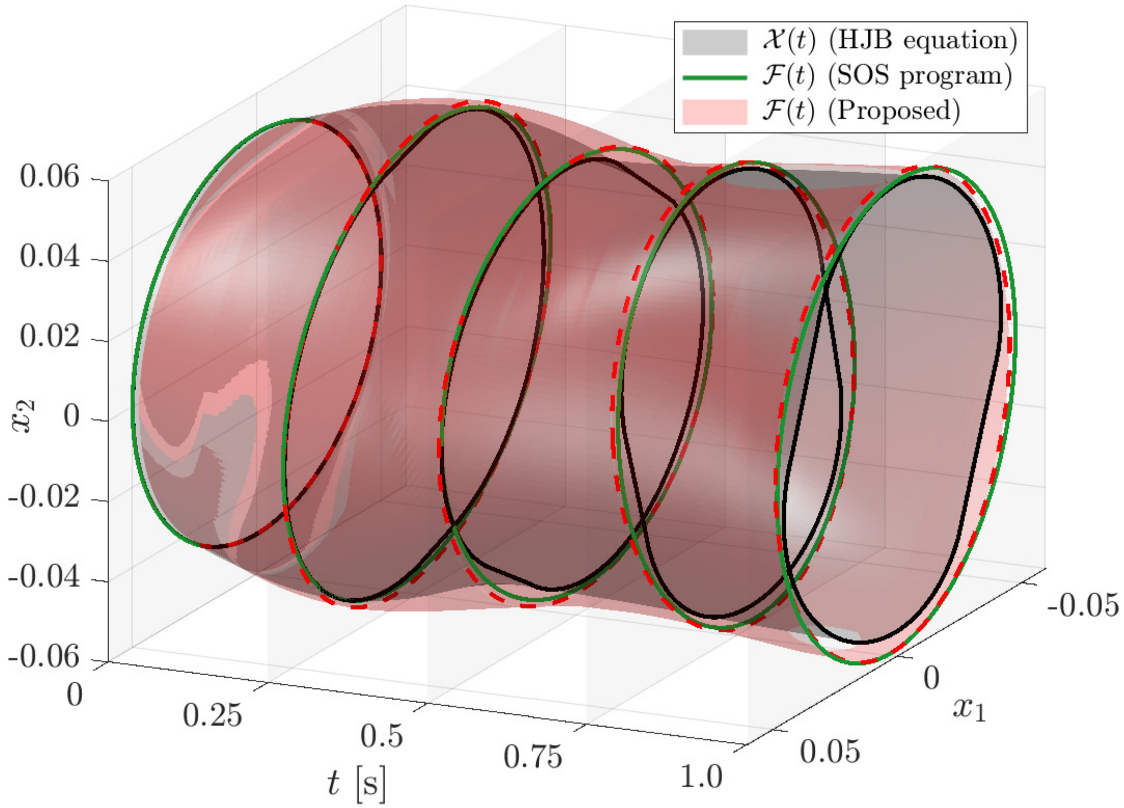


Figure 4.1: Comparison of the funnels of the system (4.20) computed by the proposed method and SOS program [8]. The shaded gray surface and the black line represent the FRS $\mathfrak{X}(t)$ from the HJB equation [23]. The shaded red surface and the dashed red line represent the result of the proposed method. The funnel computed by SOS program is depicted as green ellipsoids. Note that the sets are shifted by the nominal trajectory for a better comparison.

The proposed algorithm computes the funnel via two different parameterizations of the value function: the polynomial approximation $\hat{V}$, and the quadratic approximation $\tilde{V}$. Since a quadratic function of states is also a polynomial of states, a question may arise as to whether the two-step procedure (the polynomial and quadratic approximation) of the proposed algorithm makes sense. Note that the property of the Bernstein polynomial used in the proposed algorithm holds only for a finite domain of interest. Considering that it is difficult to compute the subzero level set of arbitrary order of a polynomial function, the quadratic approximation of the value function is required to characterize the domain of interest as an ellipsoid. However, approximating the value function to a quadratic function can be overly conservative, because the quadratic function does not have enough variables to imitate the value function. From this point of view, the polynomial approximation of the value function alleviates the problem of the lack of decision variables of a quadratic function. Fig. 4.2 represents the funnels computed by two different methods: the quadratic approximation, and the combination of quadratic and polynomial approximation. The funnel computed from only the quadratic approximation is more conservative than the funnel computed by the proposed two-step procedure as can be seen in Fig. 4.2 where the red ellipsoids are enclosed by the blue ellipsoids.

A similar tendency can be observed depending on the order of the polynomial approximation. Fig. 4.3 describes the funnels of the system (4.20) computed by 2 different orders of the polynomial value function. In this simulation, the funnel computed by the high-order polynomial, i.e. $\boldsymbol{N} = [5, 5]^\top$, is compared to the funnel computed by the low-order polynomial, i.e. $\boldsymbol{N} = [2, 2]^\top$. Both of the funnels certainly enclose the FRS as can be seen in Fig. 4.3 where the shaded gray region is located in the shaded red and green regions. However, the funnel computed by the high-order polynomial is less conservative compared to the funnel computed by the low-order polynomial as can be seen in Fig. 4.3 where the shaded red region encloses the shaded green region. This is because the high-order polynomial involves quite more decision variables (coefficients) compared to the low-order polynomial. The high-order polynomial can better approximate the value function, resulting in a less

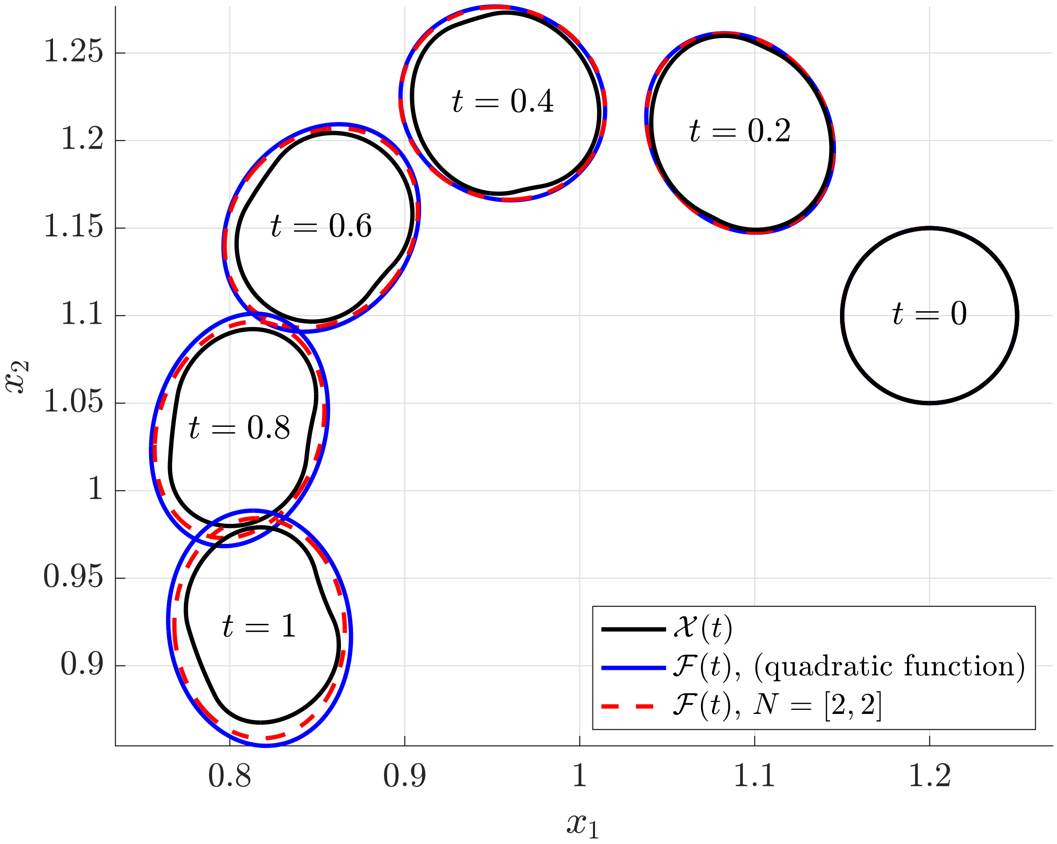


Figure 4.2: Comparison of the funnels of the system (4.20) computed by two different approaches. The solid blue lines and the dashed red lines represent the funnels computed by the quadratic approximation and the combination of the quadratic and polynomial approximation of the value function, respectively. The solid black lines represent the FRS of the system (4.20). For the polynomial approximation of the value function, $\boldsymbol{N} = [2, 2]^\top$ is used for this simulation. Other computational conditions are identical with Fig. 4.1.

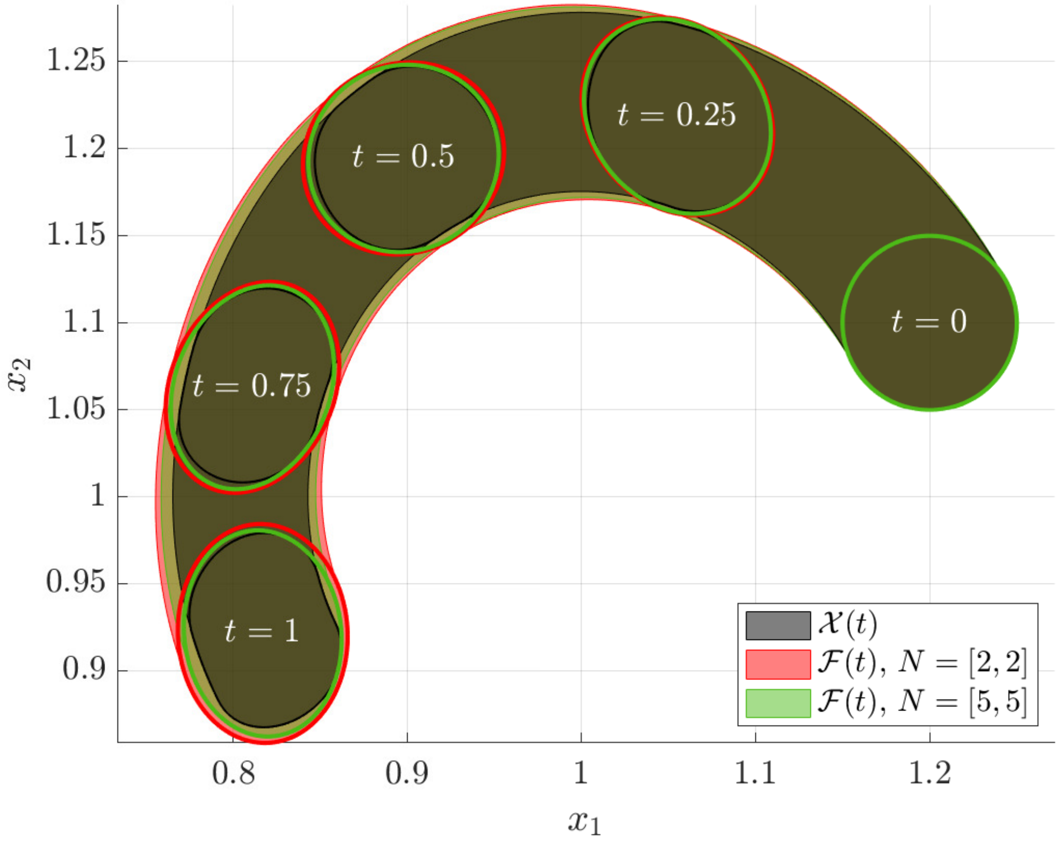


Figure 4.3: Comparison of the funnels of the system (4.20) computed by different orders of polynomials. The shaded red and the green regions are the funnels computed by the low ($\boldsymbol{N} = [2, 2]^\top$) and high-order ($\boldsymbol{N} = [5, 5]^\top$) polynomials, respectively. The shaded gray region in the colored regions represents the FRS of the system (4.20). Other computational conditions are identical with Fig. 4.1.

conservative approximation of the FRS.

Fig. 4.4 describes the tradeoff between the computation time and the conservativeness of the funnel. The higher the order of the polynomial, the tighter approximation of the FRS is possible as supported by Fig. 4.3. At the same time, the computation time of the funnel with the high-order polynomial takes longer than the low-order polynomial due to the difference in the number of variables. In Fig. 4.4, the computation time is measured as the total time spent in 3 iterations of Algorithm 2. Also, the conservativeness of the funnel is evaluated by the ratio of the volume of the funnel to the FRS, i.e.

$$\text{Conservativeness} = \int_{t_0}^{t_N} \frac{\text{Vol}(\mathcal{F}(t))}{\text{Vol}(\mathcal{X}(t))}\, dt.$$

As the order of the approximation increases, the computation time also increases, but the conservatism tends to decrease.

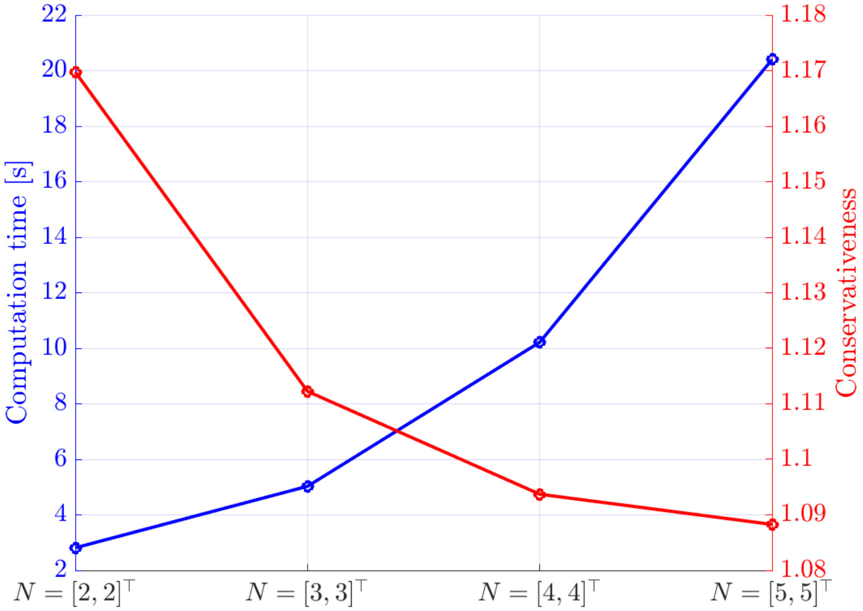


Figure 4.4: The tradeoff between the computation time and the conservativeness of the funnel. The horizontal axis of the graph represents the order of the polynomial. The blue and red lines represent the computation time and the conservativeness of the funnel, respectively. Other computational conditions are identical with Fig. 4.1.

4.5.2 Unicycle

The proposed algorithm has been tested for the following unicycle dynamics:

$$\begin{aligned}\dot{x}_1(t) &= u_1(t)\cos(x_3(t)),\\ \dot{x}_2(t) &= u_1(t)\sin(x_3(t)),\\ \dot{x}_3(t) &= u_2(t) + 0.05w(t),\end{aligned} \tag{4.21}$$

where $u = [u_1, u_2]^\top \in \mathbb{R}^2$ is the input of the system. The trigonometric functions are replaced by the 3rd order Taylor expansion to make the dynamics (4.21) polynomial of states. In this simulation, evenly discretized time segments from $t_0 = 0$ to $t_N = 1$ with $N = 50$ are used. For the polynomial approximation of the value function, $\boldsymbol{N} = [2, 2, 2]^\top$ is used. The funnels of (4.21) are computed for 5 different inputs such that $u_2(t) = 3\rho\sin(2\pi t)$, where $\rho \in \{-1, -0.5, 0, 0.5, 1\}$. Also, $u_1(t) = 1$ and $Q(t_0) = \mathrm{diag}(0.03, 0.03, 0.03)^2$ for all cases. For the computation of the FRS, each state space is gridded by 151 points, and thus a total of 3442951 grid points are used.

Similar to the results of the 2-dimensional case, the computed funnels certainly enclose the FRSs as illustrated in Fig. 4.5. Also, as illustrated in Fig. 4.6, 500 states in the initial set are sampled and propagated with the worst-case disturbance. As expected, the computed funnels certainly encompass all the states affected by disturbances. The computation time of the proposed method is 0.4571 seconds, and thus the proposed method is a suitable solution for the runtime safety verification of the system (4.21).

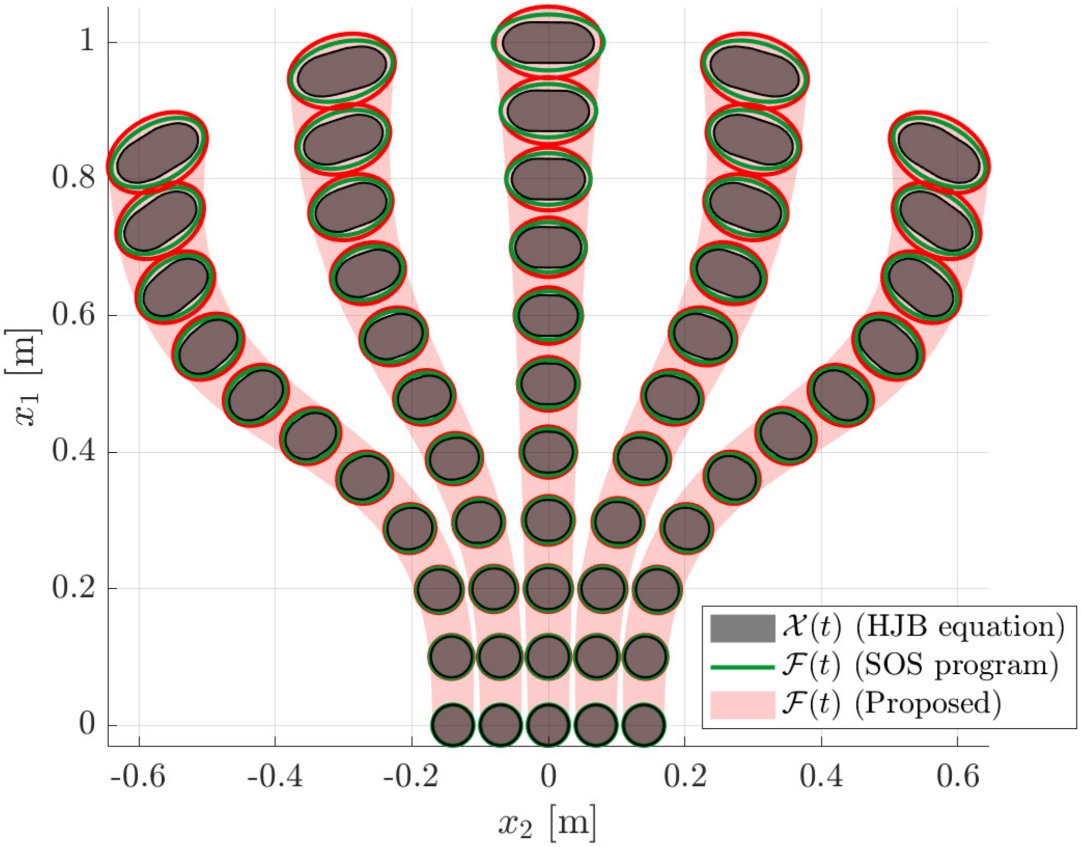


Figure 4.5: Comparison of the funnels of the system (4.20) computed by the proposed method and SOS program [8]. The shaded gray surface and the black line represent the FRS $\mathfrak{X}(t)$ from the HJB equation [23]. The shaded red surface and the dashed red line represent the result of the proposed method. The funnel computed by the SOS program is depicted as green ellipsoids. Note that the sets are projected onto x_1–x_2 plane for the visualization.

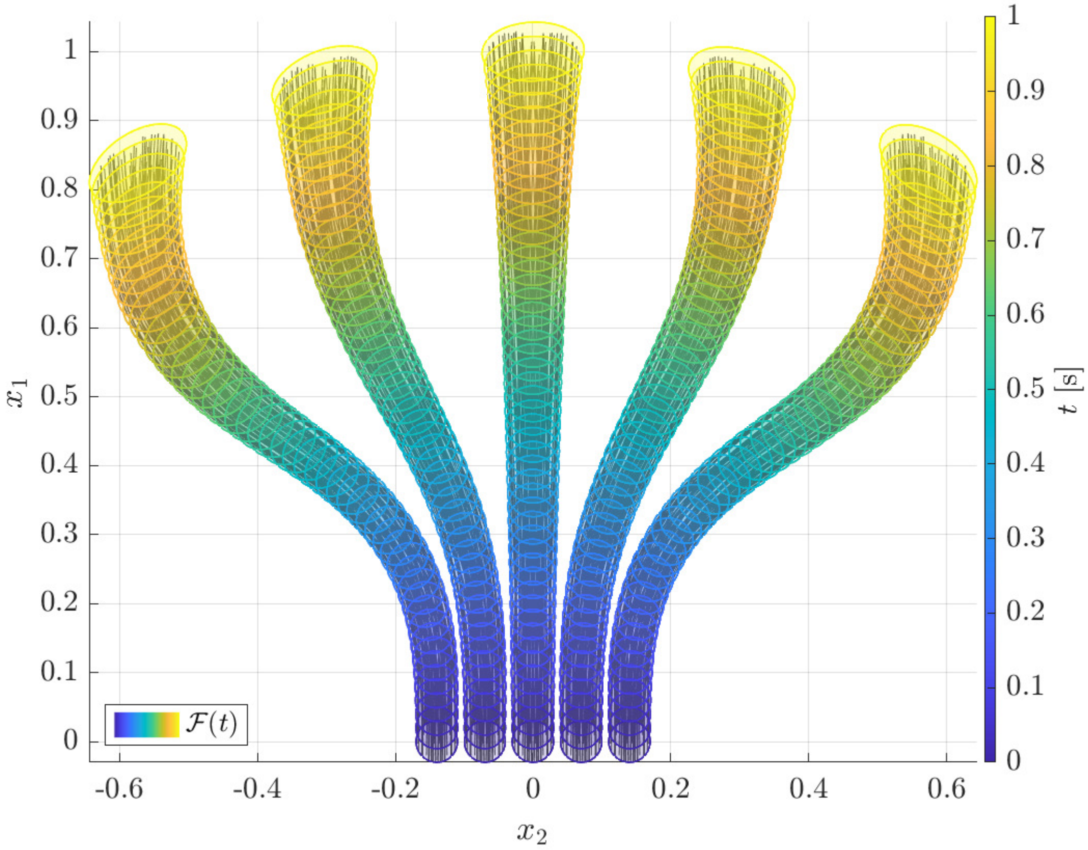


Figure 4.6: Comparison of the funnels of the system (4.21) and state trajectories interrupted by the worst-case disturbances. The colored regions are the funnels that start with blue and end with yellow. The gray lines in the colored regions are the state trajectories starting from the randomly sampled initial states.

4.5.3 Multirotor

The proposed algorithm is further applied to the following 8-state mass-normalized multirotor dynamics:

$$\begin{aligned}
&\dot{x}_1(t) = x_4(t), \quad \dot{x}_2(t) = x_5(t), \quad \dot{x}_3 = x_6(t), \\
&\dot{x}_4(t) = u_1(t)\cos(x_7(t))\sin(x_8(t)) + w_1(t)\bar{w}_1, \\
&\dot{x}_5(t) = -u_1(t)\sin(x_7(t)) + w_2(t)\bar{w}_2, \\
&\dot{x}_6(t) = u_1(t)\cos(x_7(t))\cos(x_8(t)) - g + w_3(t)\bar{w}_3, \\
&\dot{x}_7(t) = u_2(t), \quad \dot{x}_8(t) = u_3(t),
\end{aligned} \tag{4.22}$$

where $u \in \mathbb{R}^3$ is consist of the normalized thrust, and the desired roll/pitch rates, $w \in \mathbb{R}^3$ represents the external acceleration disturbances, and $g \in \mathbb{R}$ is the gravitational acceleration. For the known control sequence, the FRS of the system (4.22) can be reconstructed from FRSs of the following subsystems:

$$\dot{\xi}_i(t) = s_i(\xi_i(t), w_i(t), u(t)) \quad \forall i \in \{1, 2, 3\},$$

where s_1, s_2, and s_3 are the corresponding dynamics from (4.22), and the states of the subsystems are defined as

$$\begin{aligned}
\xi_1(t) &\coloneqq [x_1(t), x_4(t), x_7(t), x_8(t)]^\top \in \mathbb{R}^4, \\
\xi_2(t) &\coloneqq [x_2(t), x_5(t), x_7(t), x_8(t)]^\top \in \mathbb{R}^4, \\
\xi_3(t) &\coloneqq [x_3(t), x_6(t), x_7(t), x_8(t)]^\top \in \mathbb{R}^4.
\end{aligned}$$

The FRS of the system (4.22) is contained in the intersection of the reprojections of the FRSs of the subsystems [21]. Instead of directly applying the proposed algorithm to the system (4.22), the funnels of the subsystems are evaluated and then reprojected to the entire state space to compute a set containing the FRS, which is a valid funnel of the system (4.22).

Fig. 4.7 presents the funnels of the system (4.22) for some selected control sequences. The funnels are computed for the finite duration $t \in [0, 1]$ with 25 segments, and $\bar{w} = [0.01, 0.01, 0.01]^\top$ is used. The trigonometric functions in (4.22) are replaced with the 3rd order Taylor expansion. The average funnel computation time for one subsystem is measured as 1.0891 seconds with the implementation in C++. Note that the funnel of each subsystem can be computed in parallel, and thus it took about 1 second to generate the funnel of the system (4.22) for a single control sequence. Consequently, the proposed method has the potential to react to changes regarding additional disturbances at runtime and thus can be used for online safety verification.

Through Chapters 3 and 4, how to compute the funnel of the nonlinear system is presented. Based on the conservative approximation of the FRS of the LTV system and the value function of the HJB equation, the funnel computed by the proposed method tightly encompasses the FRS without compromising the theoretical guarantee. In the following chapter, the proposed funnel is used for planning the safety-guaranteed reference trajectory.

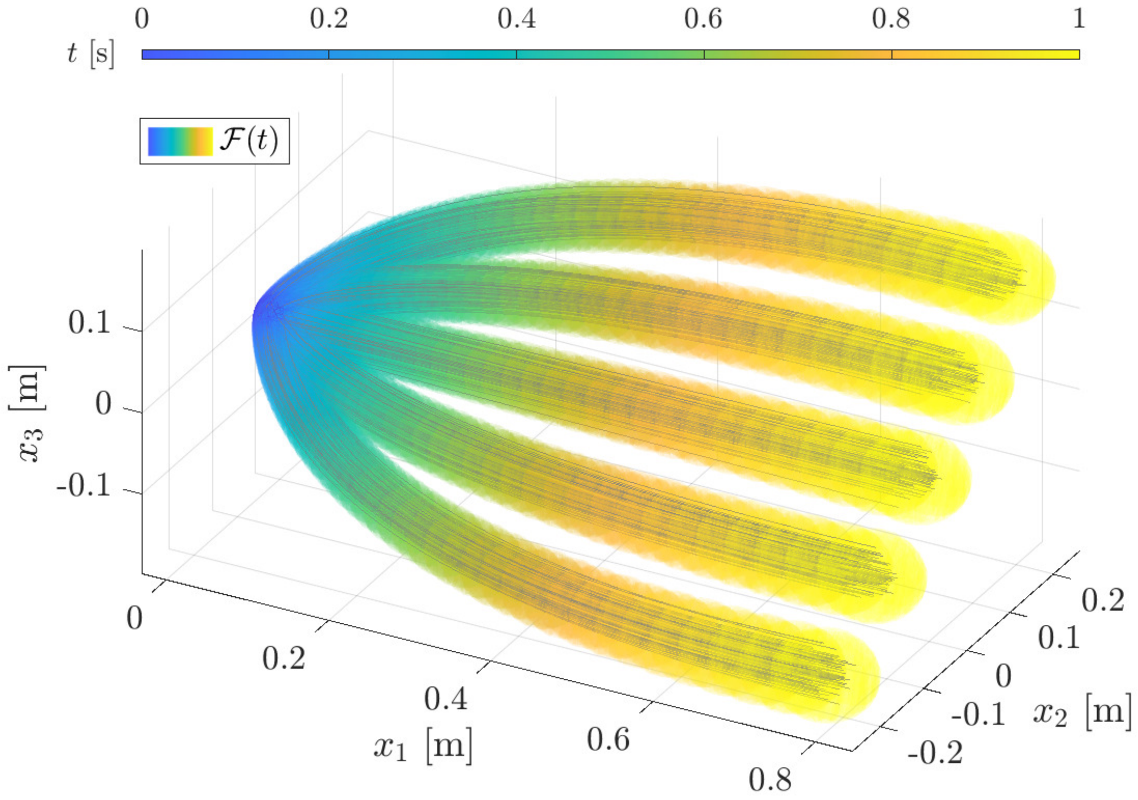


Figure 4.7: Comparison of the funnels and state trajectories of the multirotor in (4.22) interrupted by the worst-case disturbances for some selected control sequences. The colored regions are the funnels that start with blue and end with yellow. The gray lines in the colored regions are the state trajectories starting from the randomly sampled initial states. The funnels certainly enclose all the states affected by disturbances.

5

Safety-guaranteed Trajectory Planning

This section details the generation of the safety-guaranteed reference trajectory. Section 5.1 states the safety-guaranteed trajectory planning problem regarding the closed-loop dynamics of nonlinear systems. In Section 5.2, the generation of the global reference trajectory that is used as the initial guess of the local replanning is briefly explained. Section 5.3 presents the local replanning of the reference trajectory with a safety guarantee. Simulation results and discussion are provided in Section 5.4.

5.1 Problem Statement

Consider an open-loop nonlinear system $f : \mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathcal{W} \to \mathbb{R}^n$ as

$$\dot{x}(t) = f\big(x(t), u(t), w(t)\big),$$

where $u(t) \in \mathbb{R}^{n_u}$ denotes the input of the system, and $n_u \in \mathbb{R}$ is the dimension of the input. Let the reference trajectory be denoted as $r(t) \in \mathbb{R}^{n_r}$, where $n_r \in \mathbb{R}$ is the dimension of the reference state. In this work, a nonlinear state-feedback reference-tracking controller including some disturbances such as errors in state measurements, i.e. $u(t) = \mu(x(t), r(t), w(t))$, is considered. The closed-loop system is expressed as

$$\begin{aligned} \dot{x}(t) &= f\big(x(t), \mu(x(t), r(t), w(t)), w(t)\big) \\ &= g\big(x(t), w(t), r(t)\big) \\ &\coloneqq g_r\big(x(t), w(t), t\big), \end{aligned} \tag{5.1}$$

where the subscript r of g represents that the closed-dynamics depends on the reference trajectory. In the generation of the reference trajectory, the following dynamic system $s : \mathbb{R}^{n_r} \times \mathbb{R}^{n_a} \to \mathbb{R}^{n_r}$ for the reference state $r(t)$ is considered:

$$\dot{r}(t) = s(r(t), a(t)), \tag{5.2}$$

where $a(t) \in \mathbb{R}^{n_a}$ is the input of (5.2) and $n_a \in \mathbb{R}$ is the dimension of $a(t)$. As an example, if $r(t)$ is consist of the desired position and velocity of the system, the corresponding input $a(t)$ can be the desired acceleration of the system.

The planning objective is to guarantee the safety of the system described in (5.1) by modifying the reference trajectory. Accordingly, the reference trajectory $r(t)$ should be planned such that the FRS of the system (5.1) and unsafe regions do not intersect.

Problem 5.1 (Safety-guaranteed trajectory planning)**.** *For a given preplanned reference $r^d : [0, t_f] \to \mathbb{R}^{n_r}$, find the local reference during a finite horizon $t \in [t_0, t_N] \subset [0, t_f]$ that satisfies*

$$\begin{aligned} \min_{r,\, a} \quad & \int_{t_0}^{t_N} \big\|r(t) - r^d(t)\big\|^2 + \kappa\big\|a(t)\big\|^2 \, dt \\ \text{s.t} \quad & \dot{r}(t) = s(r(t), a(t)) \quad \forall t \in [t_0, t_N], \\ & \mathcal{X}(t) \cap \mathcal{X}_{unsafe} = \varnothing \quad \forall t \in [t_0, t_N], \end{aligned} \tag{5.3}$$

where $\kappa \in \mathbb{R}$ is a positive weight between tracking and regulation, $s : \mathbb{R}^{n_r} \times \mathbb{R}^{n_a} \to \mathbb{R}^{n_r}$ is the dynamics of reference state in (5.2)*, $\mathcal{X}(t)$ is the FRS of the system in* (5.1)*, and $\mathcal{X}_{unsafe} \subset \mathbb{R}^n$ is the set of unsafe states.*

In Chapters 3 and 4, reachability analysis has been performed on the dynamics whose nominal trajectory is centered at the origin, as mentioned in (2.2). Accordingly, the closed-loop dynamics in (5.1) is also shifted to the origin in order to utilize the proposed method. Given the reference trajectory $r(t)$, the nominal trajectory $q(t) \in \mathbb{R}^n$ is defined as the solution of the disturbance-free dynamics:

$$\dot{q}(t) = g_r(q(t), 0, t) \tag{5.4}$$

with the initial condition $q(t_0) = x(t_0)$. By introducing the deviation of state $\tilde{x}(t) \coloneqq x(t) - q(t) \in \mathbb{R}^n$, the time-derivative of $\tilde{x}(t)$ is

$$\begin{aligned} \dot{\tilde{x}}(t) &= g_r(\tilde{x}(t) + q(t), w(t), t) - g_r(q(t), 0, t) \\ &\coloneqq \tilde{g}_r(\tilde{x}(t), w(t), t), \end{aligned} \tag{5.5}$$

and it is apparent that the dynamics in (5.5) is centered at the origin. Let $\tilde{\mathcal{F}}(t) \subset \mathbb{R}^n$ be the funnel of the shifted dynamics in (5.5). Consequently, the funnel $\mathcal{F}(t)$ of the dynamics in (5.1) can be represented as

$$\mathcal{F}(t) = q(t) + \tilde{\mathcal{F}}(t) = \mathcal{E}(q(t), Q(t)) \quad \forall t \in [t_0, t_N], \tag{5.6}$$

where $Q(t) \in \mathbb{S}^n_{++}$ is the shape matrix computed from Algorithm 2 associated with the shifted dynamics in (5.5), and $\mathcal{E} : \mathbb{R}^n \times \mathbb{S}^n_{++} \to \mathcal{P}(\mathbb{R}^n)$ denotes an ellipsoid centered at the first argument vector.

5.2 Global Reference Planning

As proposed in Problem 5.1, the objective of planning is the local modification of the current reference $r(t)$ for $t \in [t_0, t_N]$ considering the global reference $r^d(t)$ for $t \in [0, t_f]$. The global reference trajectory r^d is planned only once in the offline phase, and the local reference trajectory r is generated to follow the preplanned reference r^d in runtime. This preplanning process on r^d is necessary as a top-level command of the system to make the system towards the goal. In this work, smooth piecewise polynomials of time t [37] that connect some predefined waypoints between the start and goal points are used. Instead of the piecewise polynomials, a sampling-based [38] or grid-based method [39] can also be used for the preplanning. Note that safety constraints such as collision avoidance are not considered for the preplanned reference r^d, since guaranteeing safety is the role of the local reference optimization as detailed in the following subsection.

5.3 Local Replanning with Safety Guarantee

Given the preplanned reference r^d, this subsection aims to solve Problem 5.1 using the funnel $\mathcal{F}(t)$ in (5.6). Since it is proven that the FRS $\mathcal{X}(t)$ in (2.3) is contained in the funnel, the following implication for the constraint of (5.3) holds:

$$\mathcal{E}(q(t), Q(t)) \cap \mathcal{X}_{unsafe} = \varnothing \quad \Longrightarrow \quad \mathcal{X}(t) \cap \mathcal{X}_{unsafe} = \varnothing.$$

The unsafe set is considered as the following ellipsoid

$$\mathcal{X}_{unsafe} = \mathcal{E}(q_{unsafe}, Q_{unsafe}),$$

where $q_{unsafe} \in \mathbb{R}^n$ and $Q_{unsafe} \in \mathbb{S}^n_{++}$ are the center and shape matrix of the unsafe set respectively. In order to make the intersection of two ellipsoids empty, the distance between two ellipsoids must be positive. Consequently, the following inequality is used as the constraint of (5.3), instead of $\mathcal{X}(t) \cap \mathcal{X}_{unsafe} = \varnothing$:

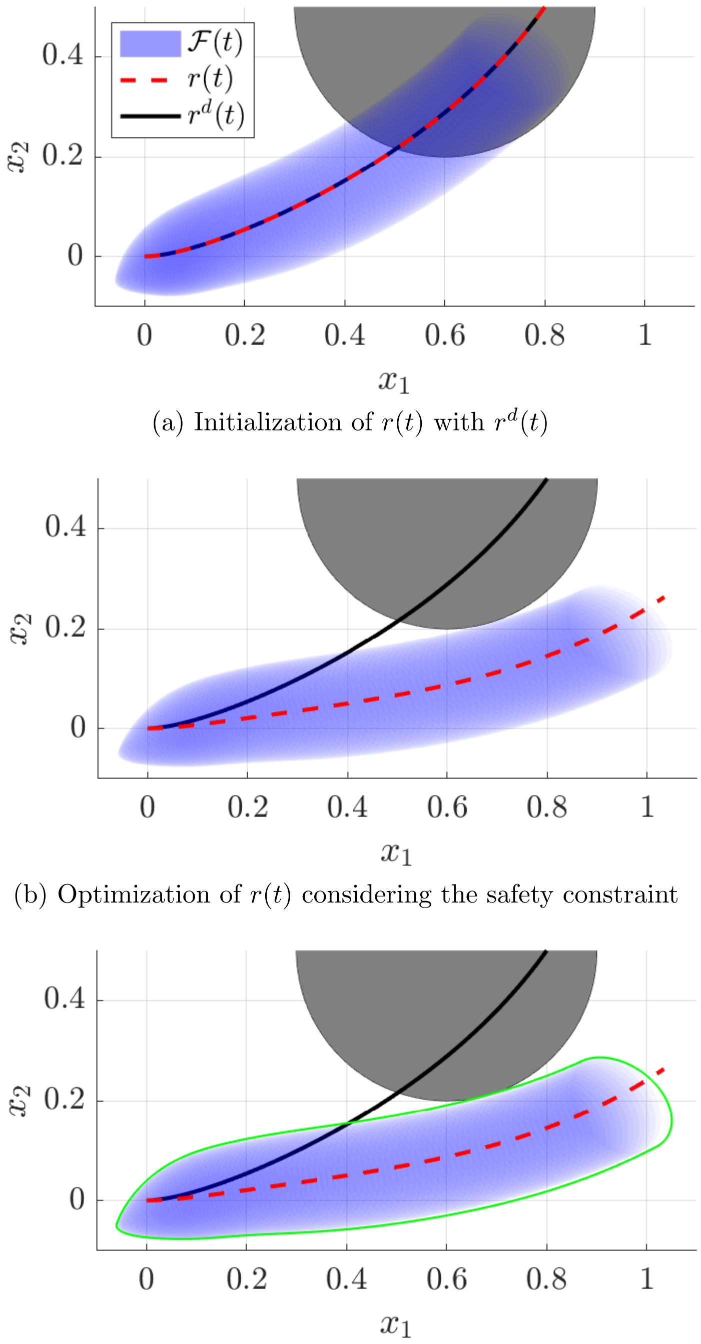
$$(q(t) - q_{unsafe})^\top \big(Q_x(t) \oplus Q_{unsafe}\big)^{-1} (q(t) - q_{unsafe}) > 1,$$

where the operation $\oplus$ between positive definite matrices is defined in (3.21).

Fig. 5.1 illustrates how the local reference trajectory $r(t)$ is optimized. First, $r(t)$ is initialized with $r^d(t)$ for $t \in [t_0, t_N]$, and the nominal trajectory $q(t)$ is computed according to $r(t)$. Then, the funnel $\mathcal{F}(t)$ is computed along the nominal trajectory. As can be seen in Fig. 5.1a, the funnel collides with the obstacle at the initial step. By solving Problem 5.1, the reference trajectory is updated with a new one which is provably collision-free as well as closest to the predefined reference as described in Fig. 5.1b. Note that the shape of the funnel is fixed during the optimization, whereas the nominal trajectory is renewed according to the reference trajectory. After updating the local reference as in Fig. 5.1c, the funnel is reevaluated according to the updated reference. Those processes (the funnel computation and trajectory optimization) are repeated until the local reference trajectory converges. When the iteration is over, the resultant local reference trajectory $r(t)$ for $t \in [t_0, t_N]$ is commanded to the closed-loop dynamics of the system.

Since the funnel can be computed independently of the trajectory optimization procedure, any optimization algorithm which can handle nonlinear constraints can be employed. In this work, the constrained version of differential dynamic programming (DDP) [40] is used.

The safety-guaranteed reference trajectory $r(t)$ should be generated repeatedly as time progresses, considering unexpected disturbances during runtime. Let us denote $\Delta t > 0$ as the planning cycle which represents the timing between the previous and current trajectory generation. Suppose a new reference trajectory $r(t)$ for a new horizon $t \in [t_k, t_{k+N}]$ is planned, where $[t_k, t_{k+N}] = \Delta t + [t_0, t_N]$. The result of the previous horizon can be used, i.e. a warm-start strategy, for the current reference generation. The current value is initialized as $a(t) = a_{prev}(t)$ for $t \in [t_k, t_N]$ and $a(t) = a_{prev}(t_N)$ for $t \in [t_N, t_{k+N}]$, where the subscript *prev* represents the corresponding value in the previous prediction horizon. Then, the initial conditions of the reference, center and shape matrix of the funnel, are updated as $r(t_k) = r_{prev}(t_k)$, $q(t_k) = q_{prev}(t_k)$, and $Q(t_k) = Q_{prev}(t_k)$, respectively. From (5.2), (5.4), and Algorithm 2, the terms $r(t)$, $q(t)$, and $Q(t)$ for $t \in [t_k, t_{k+N}]$ are computed by the warm-started $a(t)$, and used as the initial guess of the current prediction horizon.



(a) Initialization of $r(t)$ with $r^d(t)$

(b) Optimization of $r(t)$ considering the safety constraint

(c) Recomputation of $\mathcal{F}(t)$ with the updated reference $r(t)$

Figure 5.1: The procedure of the proposed trajectory planning. The shaded blue areas represent the funnel $\mathcal{F}(t)$ and the dashed red lines describe the corresponding reference trajectory $r(t)$. The black lines and the gray regions are the predefined reference $r^d(t)$ and an obstacle, respectively. The green line in (c) represents the boundary of the funnel in (b), which is slightly different from the blue region of (c).

5.4 Simulation Results and Discussion

This subsection presents the validation of the proposed planning algorithm with the 4-state unicycle described in Appendix A. The predefined reference $r^d(t)$ is generated with piecewise polynomials, which connects $r^d(0) = [0, 0, 0.2, 0]^\top$ and $r^d(t_f) = [2, 2, 0.2, 0]^\top$ with $t_f = 6$. Also, an additional disturbance that is assumed to be unknown, but can be estimated when the unicycle enters a disturbed region is considered. In the disturbed region, an unknown friction force that acts along the driving direction is increased. So, $\bar{w}_1$ is enlarged when the unicycle enters the region with the disturbance. The local reference trajectory $r(t)$ is planned for 1 second of prediction horizon, i.e. $t_N - t_0 = 1$. The optimization in Problem 5.1 is solved every 0.1 seconds in the simulation time. Detailed descriptions of the parameters used in the simulation are listed in Table 5.1.

Fig. 5.2 illustrates the trajectory optimization results and the corresponding funnels at some selected instances. Starting from the initial condition $r(0) = r^d(0)$, the local reference $r(t)$ is computed such that the funnels avoid obstacles while following the predefined reference $r^d(t)$. Fig. 5.2a describes the moment when the unicycle enters the disturbed region. According to the updated disturbance bound, the funnels predicted from the current time t_0 are also enlarged, as shown in the difference between the light blue and the dark blue regions. This size increase makes the local reference $r(t)$ predicted from the current time

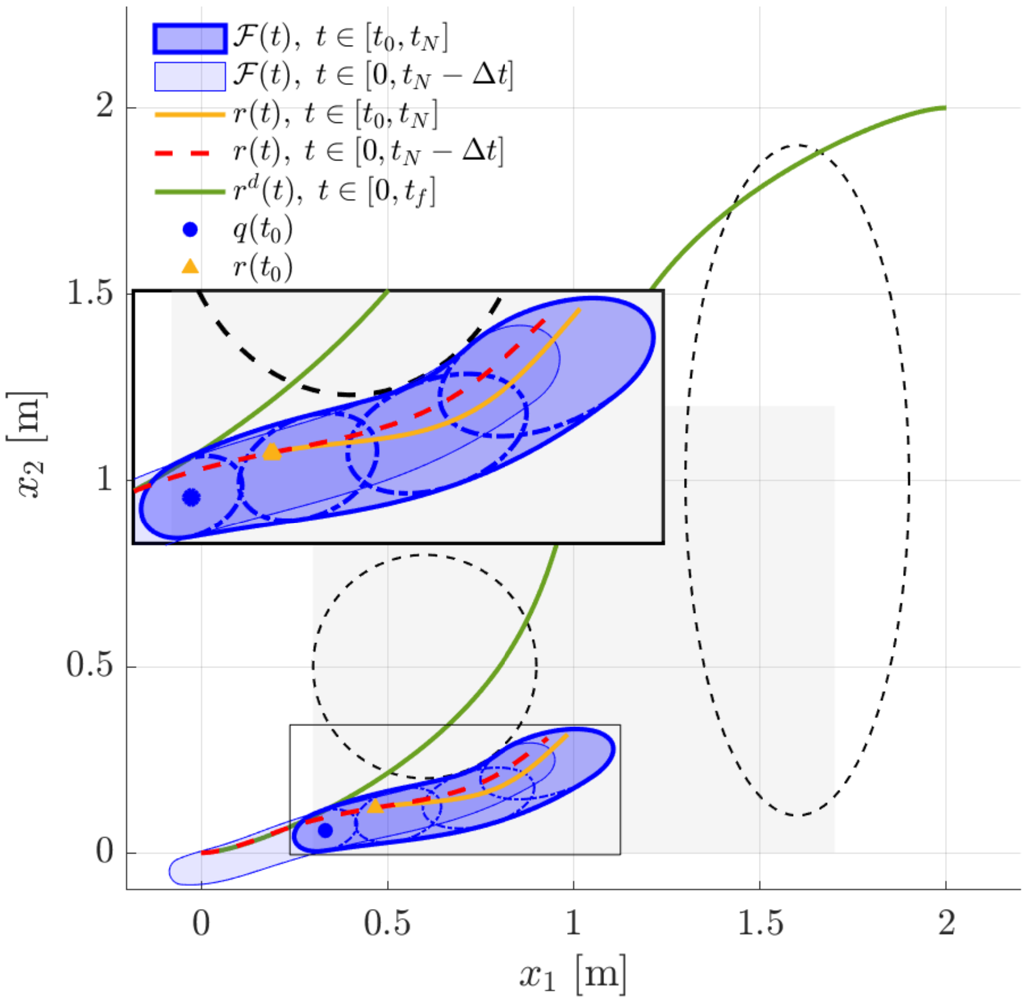
Description	Value
Initial nominal state	$q(0) = [-0.05\text{m}, -0.05\text{m}, 0.2\text{m/s}, 0.1745\text{rad}]^\top$
Initial shape matrix	$Q(0) = \text{diag}([0.0354\text{m}, 0.0354\text{m}, 0.0354\text{m/s}, 0.0354\text{rad}])^2$
Disturbance bound	$\bar{w} = [0.05\text{m/s}^2, 0.05\text{rad/s}^2, 0.01\text{m}, 0.01\text{m}, 0.01\text{m/s}, 0.01\text{rad}]^\top$
Disturbance bound (in disturb. region)	$\bar{w} = [0.4\text{m/s}^2, 0.05\text{rad/s}^2, 0.01\text{m}, 0.01\text{m}, 0.01\text{m/s}, 0.01\text{rad}]^\top$
Discretization step	$t_k - t_{k-1} = 0.01\,\text{s}$
Prediction horizon	$t_N - t_0 = 1\,\text{s}$
Planning cycle	$\Delta t = 0.1\,\text{s}$

Table 5.1: Parameters used in the simulation.

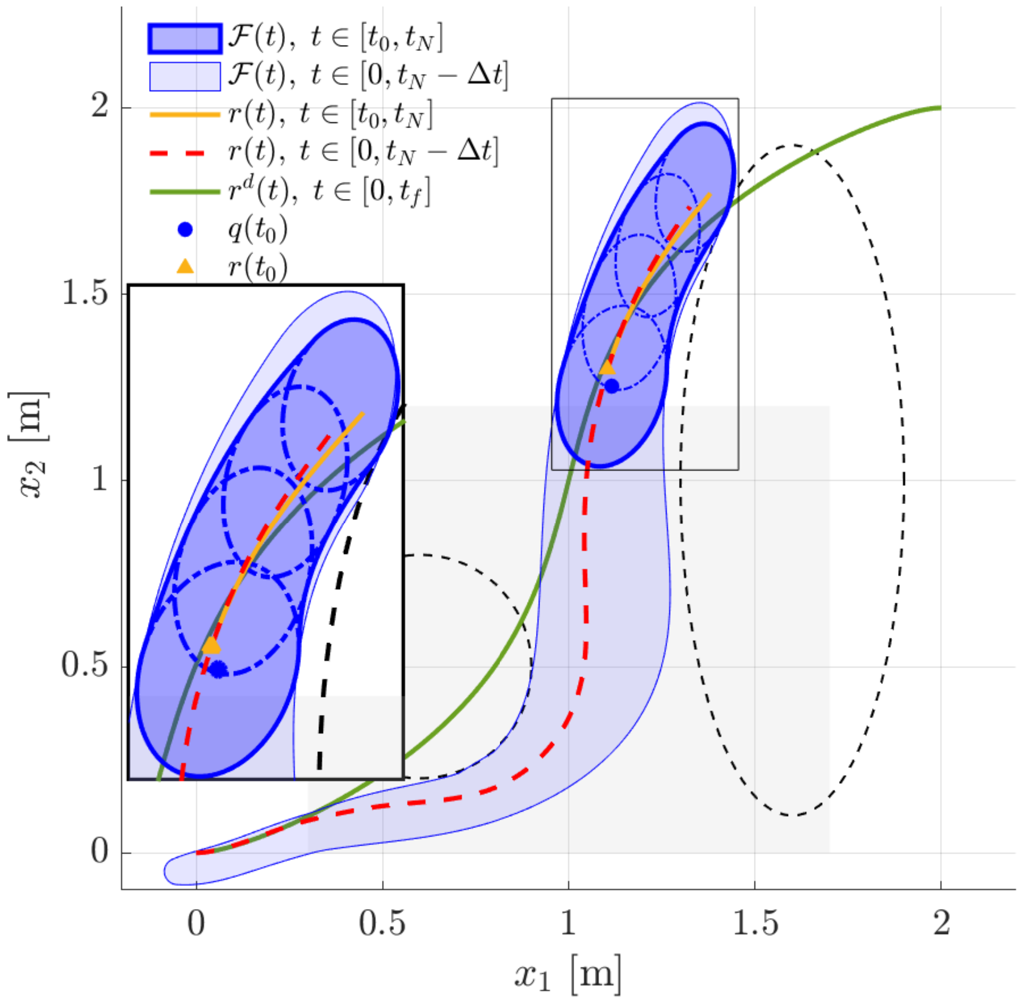
(the orange line) detour the obstacle more securely compared to the previously predicted local reference (the dashed red line). Similarly, Fig. 5.2b shows the trajectory optimization results when the unicycle escapes from the disturbed area. Since the increased disturbance bound is reverted to the original one at this moment, the size of the funnels predicted from the current time slightly decreased compared to the previous predictions. Also, the local reference $r(t)$ predicted from the current time is modified to follow the predefined reference $r^d(t)$ (the green line) a little more compared to the previously optimized local reference. The safety guarantee of the optimized local reference $r(t)$ is validated in Fig. 5.2c. 20,000 states are randomly sampled in the initial set $\mathcal{E}(q(0), Q(0))$ and the worst-case trajectory for each sampled initial state are computed using the method of characteristics [41]. As expected, the funnels enclose the worst-case trajectories for the entire duration of the mission. Consequently, the optimized local reference $r(t)$ can drive all of the states in the initial set to the goal point with guaranteed safety regardless of the unexpected disturbances.

The proposed trajectory optimization algorithm generates the *locally* optimal reference $r(t)$ for receding horizon. Thus, the quality of the solution depends on the preplanned reference trajectory $r^d(t)$. Moreover, a feasible reference may not be found depending on the initial guess. To alleviate this inherent limitation of the local planner, a search-based method such as A* or RRT* can be used in some regions adjacent to the current state to obtain multiple hypotheses. Then, a good initial guess for trajectory optimization can be selected. Also, similar to the concept of [8], it is possible to precompute the funnels for some local references $r(t)$ during the offline phase and choose the best reference to escape from the local minima in runtime. To summarize, the local minima issue can be addressed using various simple technical ideas.

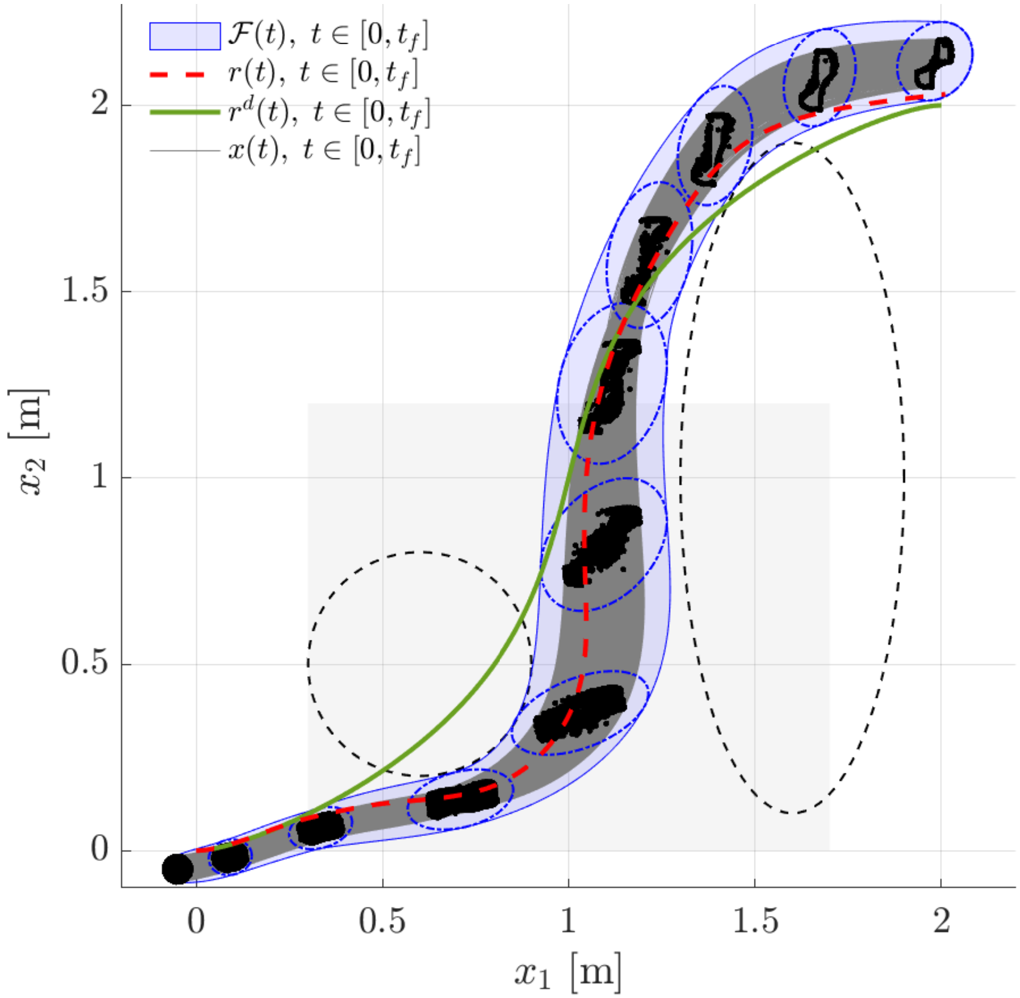
Along with the local minima issue mentioned above, the optimization may become infeasible in complex environments. The main reason for the infeasibility is the conservativeness of the proposed reachability analysis. The conservativeness of the funnel may hinder the planner from finding a collision-free trajectory when the obstacles are densely distributed. Since the funnel computed by the proposed method become conservative as the prediction



(a) Results when the nominal state enters the disturbed region



(b) Results when the nominal state exits the disturbed region



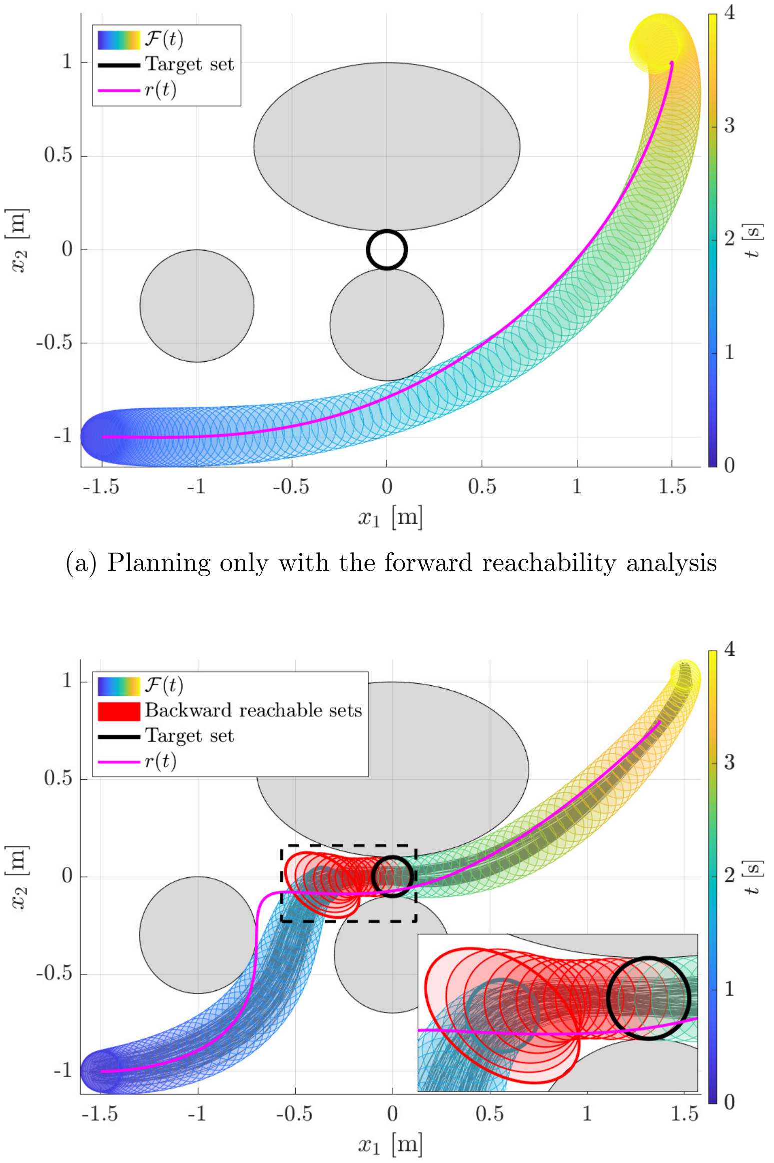
(c) Results for the entire duration

Figure 5.2: Results of the robust trajectory optimization for some selected instances. The regions with light blue and dark blue represent the funnel $\mathcal{F}(t)$ for the previous and current predictions, respectively. The dashed red lines describe the previously optimized local reference $r(t)$, and the orange lines are the local reference $r(t)$ computed at the current instance. The green lines are the predefined reference $r^d(t)$ generated in the offline phase. The dashed black lines and the shaded gray regions are the unsafe and disturbed regions, respectively. In (c), the gray lines are the worst-case trajectories computed from [41] for some sampled initial states in the initial set. The black dots and the blue ellipsoids represent the states on the worst-case trajectories and the funnels at some selected instances.

time horizon gets longer, the safety margin of the collision avoidance constraints can be reduced for the latter part of the prediction horizon if it is needed. Also, when the size of the funnel becomes unintentionally large, the initial set can be reset as a region near the current state in order to remove the accumulated conservativeness.

Incorporation with the backward reachable sets from a target set (e.g. a narrow gap between obstacles) of states can also be considered to improve the feasibility of the planner [42]. The backward reachable set represents a set of states which is guaranteed to reach the target set considering all possible disturbances. As far as the reference trajectory is planned such that the funnel and the backward reachable set are connected, the system is guaranteed to safely pass through the obstacles even in complicated environments. An illustration of the planning in the complicated environment combined with the backward reachability analysis can be seen in Fig. 5.3.

As for dynamic environments where the obstacles move over time, the safety guarantee of the proposed planning algorithm depends on whether the movement of the obstacles is known or not. Since the proposed trajectory planning algorithm runs sufficiently fast as validated by flight experiments in the next section, it is possible to avoid the moving obstacles by rapidly replanning the reference trajectory provided that the movement of the obstacles is known or can be predicted. When the movement of the obstacles cannot be predicted, the unknown movement of the obstacles can be treated as additional bounded disturbances. Similar to the concept of [4], the reachability analysis can be performed on the relative dynamics (i.e. the difference between the system and the obstacles), and then the reference trajectory can be planned so that the relative distance between the system and the obstacles becomes larger than the safety margin.



(a) Planning only with the forward reachability analysis

(b) Planning with the backward reachability analysis

Figure 5.3: Illustrative simulation for the system passing through between obstacles. The colored regions are the funnel $\mathcal{F}(t)$, which start with blue and end with yellow. The magenta line is the planned reference trajectory $r(t)$. The thick black line represents the target set and the red regions are the corresponding backward reachable sets. The shaded gray regions are obstacles. (a) Planning that only considers the forward reachability analysis fails to find the reference trajectory which passes through the target region. (b) Planning with the backward reachable sets generates the reference trajectory that can traverse the complex region without collision. The gray trajectories inside of the colored regions are the reference tracking results starting from the randomly sampled initial states.

6

Experimental Validation

This section presents the results of the proposed trajectory planning algorithm through indoor flight experiments with a multirotor. In Section 6.1, the experiment scenario and environment are briefly described. Setups for the experiment are followed in Section 6.2, and the results and discussion are presented in Section 6.3.

6.1 Scenario

The objective of the experiments is to avoid obstacles in an environment disturbed by the wind. Before the experiment, the global reference trajectory $r^d(t)$ is preplanned with piecewise polynomials that connect some designated waypoints, as described in Fig. 6.1. Two electric fans that generate gusts up to 5 m/s are installed to exert additional disturbances unknown to the multirotor, as depicted in Fig. 6.2. While the multirotor follows the global reference three times, the electric fans are moved so that the region affected by gust is changed for each flight cycle.

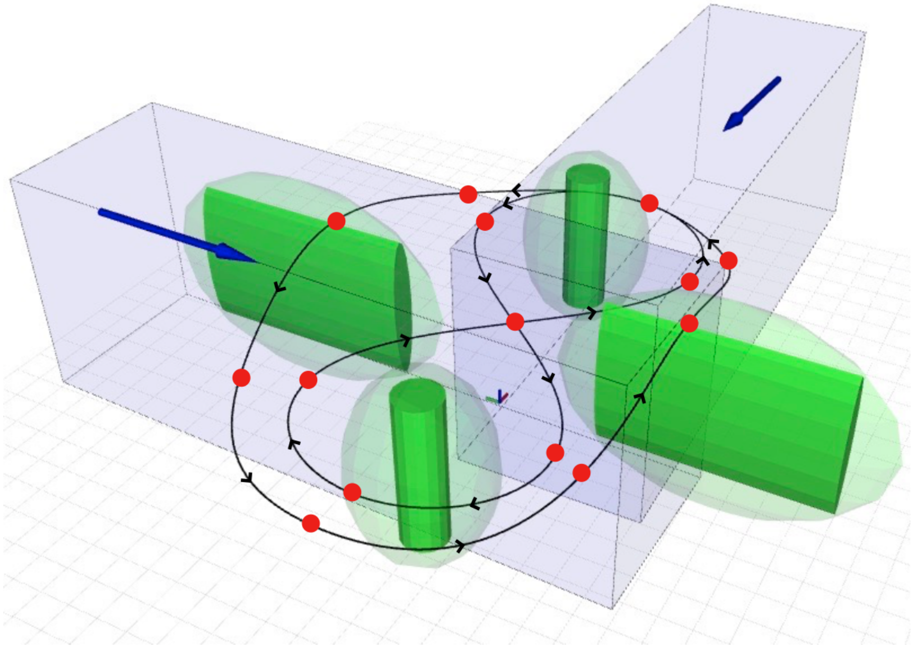


Figure 6.1: The environment of the experiment. The black line is the predefined reference $r^d(t)$ generated from the waypoints described as red dots. The green objects are the obstacles modeled as ellipsoids. The blue boxes and arrows are the expected regions affected by gusts and the direction of gusts, respectively. The arrowheads indicate the direction of the reference.

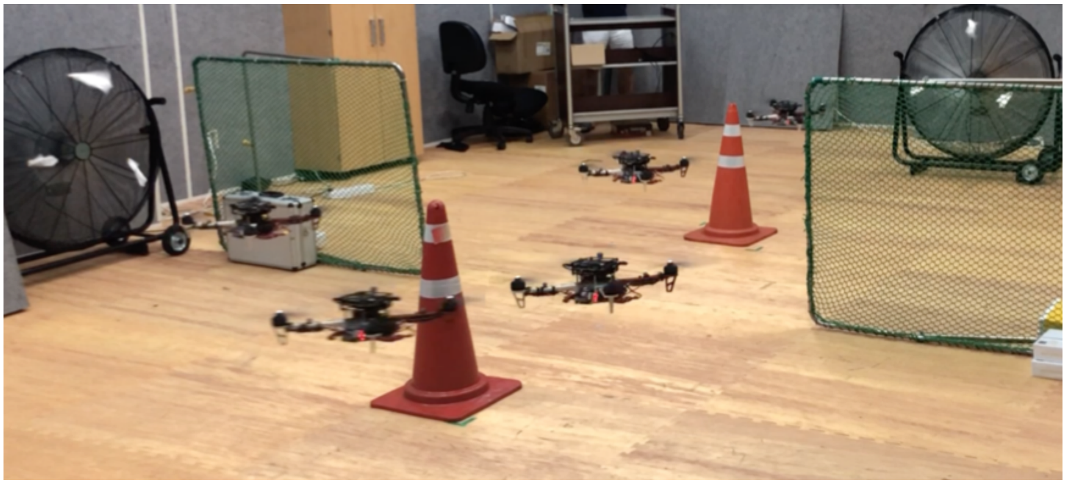


Figure 6.2: Snapshot of the experiment. Two green nets and two red cones are used as obstacles. Also, two black fans generate gusts in the environment. The multirotor avoids the obstacles against gust.

6.2 Setups

The platform is a customized F450 multirotor from DJI, as described in Fig. 6.3. E800 motors with 620S electronic speed controllers are used as a propulsion system. A flight control unit (Pixhawk4) is installed at the center of the multirotor and connected to an onboard computer via USB. For the onboard computer, intel NUC (8 GB RAM with quad-core i7-7567U@3.5 GHz CPU) running Ubuntu 18.04 is used. A MAVROS package is used for the communication between the flight control unit and the onboard computer. The position and velocity of the multirotor are measured with VICON and sent to the onboard computer via Crazyradio 2.4 GHz radio telemetry. The transmitted state measurements are used for onboard control and planning algorithms. Geometric controller [43] is implemented to generate the desired thrust and angular velocity inputs. Those commands are sent to the autopilot, and it is assumed that the flight control unit can track the desired thrust and angular velocity commands sufficiently well. The detailed dynamics of the multirotor system is described in Appendix B.

For the constrained optimization of the local reference $r(t)$, a constrained version of DDP algorithm [40] is implemented with C++. The local reference is generated for 2 seconds of the prediction horizon with 0.01 seconds of the time step. The optimization problem is solved for every 0.1 seconds so that the overall frequency of trajectory planning becomes

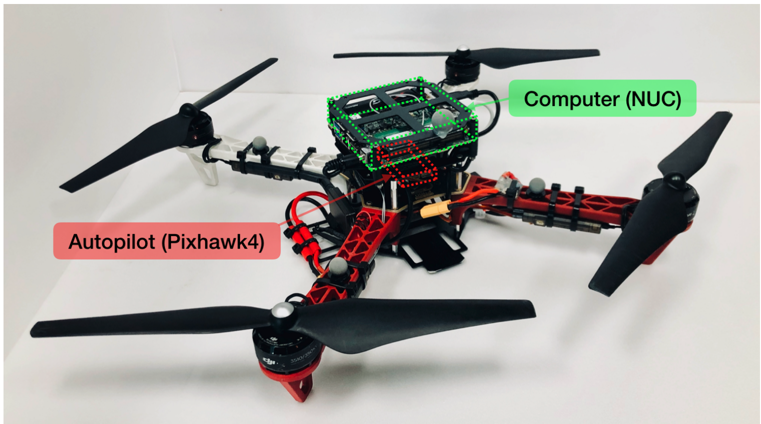


Figure 6.3: The platform used in the experiment

about 10 Hz. It is assumed that the position and size of the obstacles are known, and the obstacles are considered as ellipsoids. A disturbance estimation algorithm [44] is also implemented so that the estimated bounds of the external disturbances are automatically adjusted. Let $\hat{\Delta} \in \mathbb{R}^3$ denotes the estimated disturbance. The bound of the external disturbance is updated as $\bar{\Delta} = \Delta_0 + |\hat{\Delta}|$, where $\Delta_0 \in \mathbb{R}^3$ represents the nominal magnitude of the external disturbances. The other disturbance bounds (the maximum state measurement errors) are constant during the entire experiment. Parameters used in the experiment are listed in Table 6.1. The overall structure of the system, as well as the proposed algorithm, is described in Fig. 6.4.

6.3 Results and Discussion

Fig. 6.5 describes the trajectories and the corresponding funnel of the multirotor per cycle. During the entire experiment, the multirotor successfully avoids the obstacles and follows the 8-shaped trajectory despite the external disturbance from unexpected gusts. As the multirotor approaches the electric fans, the size of the funnel is enlarged according to the estimated magnitude of external disturbances. The reference trajectory $r(t)$ is adjusted in a safer direction to bypass the obstacles more certainly. Thanks to the real-time computation of the funnel, those kinds of evasive maneuvers can happen whenever the system perceives additional disturbances. Table 6.2 validates the real-time compatibility of the proposed planning algorithm.

As can be seen in Fig. 6.6, the funnels properly contain the states at most times. Some sources of the occasional violations are listed as follows: First, the inaccuracy of the estimated external disturbance can degenerate the guarantee. The disturbance estimator may not converge fast enough to the true value when the multirotor suddenly faces gusts and maneuvers in the turbulent windy area formed by two electric fans. The funnel cannot capture the deviation of states appropriately because the actual magnitude of disturbances exceeds the expected bounds. Second, the actual multirotor system may have unknown

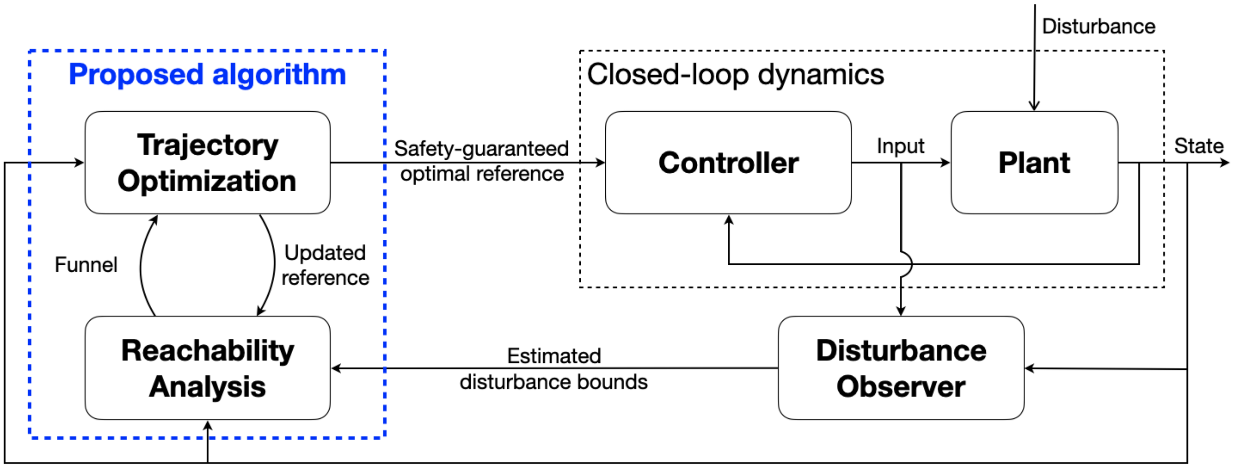


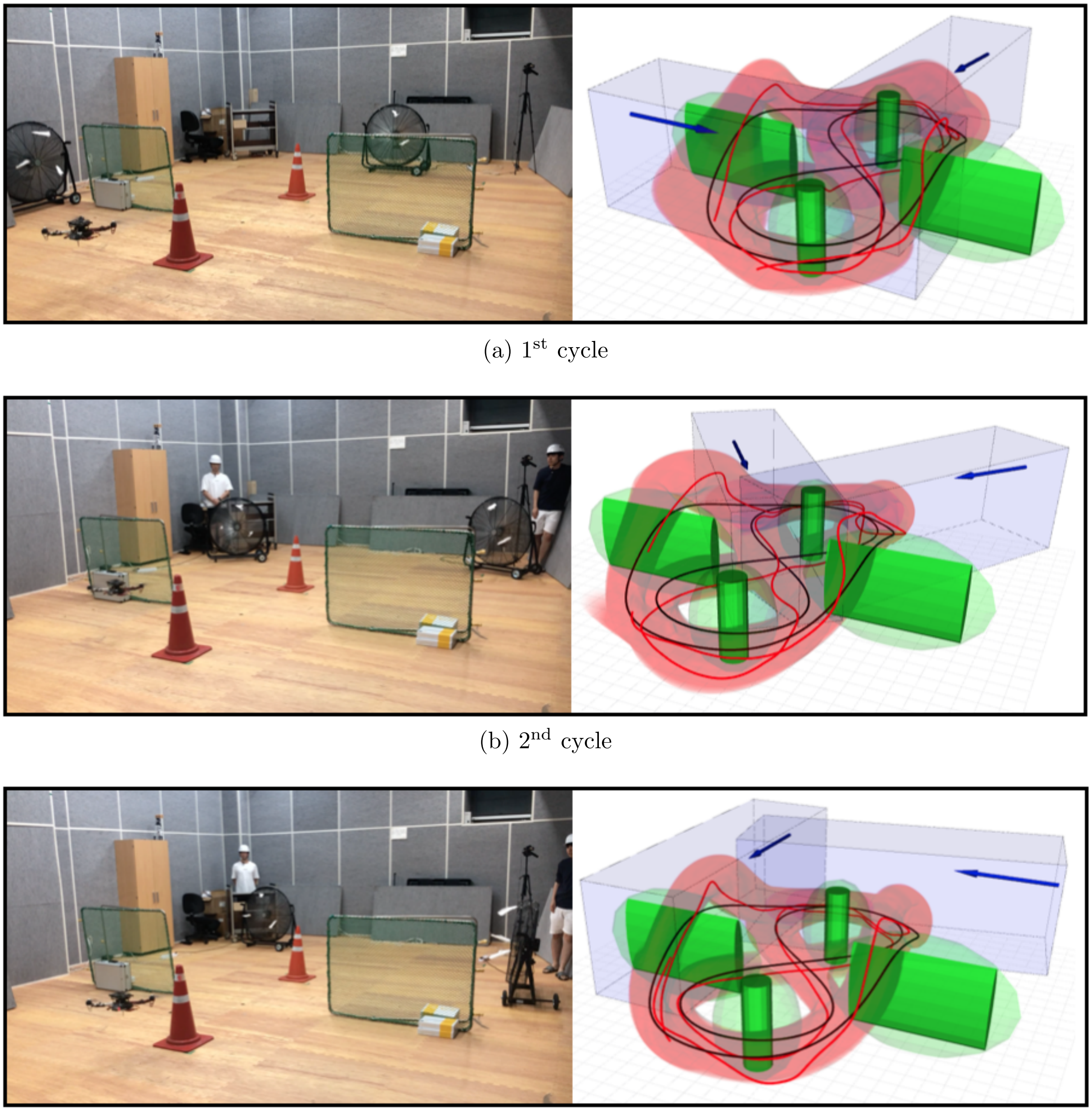
Figure 6.4: Structure of modules used in the experiment

Description	Value
Initial nominal state	$q(0)$ = state of the multirotor at the initial
Initial shape matrix	$Q(0) = \text{diag}([0.1\,\text{m}, 0.1\,\text{m}, 0.1\,\text{m},$ $0.1\,\text{m/s}, 0.1\,\text{m/s}, 0.1\,\text{m/s},$ $0.01\,\text{rad}, 0.01\,\text{rad}, 0.05\,\text{rad}])^2$
Position measurement bound	$\bar{\delta}\mathbf{p} = [0.05\,\text{m}, 0.05\,\text{m}, 0.05\,\text{m}]^\top$
Velocity measurement bound	$\bar{\delta}\mathbf{v} = [0.02\,\text{m/s}, 0.02\,\text{m/s}, 0.02\,\text{m/s}]^\top$
Attitude measurement bound	$\bar{\delta}\mathbf{q} = [0.005\,\text{rad}, 0.005\,\text{rad}, 0.005\,\text{rad}]^\top$
Nominal external disturbance	$\Delta_0 = [0.15\,\text{m/s}^2, 0.15\,\text{m/s}^2, 0.15\,\text{m/s}^2]^\top$
Discretization step	$t_k - t_{k-1} = 0.01\,\text{s}$
Prediction horizon	$t_N - t_0 = 2\,\text{s}$
Planning cycle	$\Delta t = 0.1\,\text{s}$

Table 6.1: Parameters used in the experiment

dynamics that are not reflected in the closed-loop dynamics. The gust directly fronted to the propellers of the multirotor complicates aerodynamics, and it makes the overall dynamics considerably different from the nominal one. Also, the low-level controller implemented in the autopilot may not perfectly track the commands from the high-level controller.

To validate the importance of the modification of the disturbance bound during runtime, comparative flight experiments are performed as can be seen in Fig. 6.7. Two flight experiments are qualitatively compared. In the first experiment of Fig. 6.7, the disturbance bound is updated as the multirotor enters the disturbed region. Considering the modified funnel at the time the multirotor enters the risky area, the reference trajectory is computed in a safer direction, resulting in the successful obstacle avoidance in the disturbed environment as can be seen in the first three columns of Fig. 6.5. In the second experiment of Fig. 6.7, the disturbance bound is not renewed, and the nominal value of the disturbance bound is used for computing the funnel. Without modifying the disturbance bound, the multirotor collides with the obstacle because the funnel no longer guarantees safety when the multirotor is affected by gusts, as can be seen in the last column of Fig. 6.5.



(a) 1^{st} cycle

(b) 2^{nd} cycle

(c) 3^{rd} cycle

Figure 6.5: Results of the experiment. Figures in the left columns are snapshots of the environment, and the right columns are the corresponding visualizations. The solid red lines represent the actual trajectory of the multirotor. The red regions describe the funnels, and the black lines are the preplanned global references. The green objects are the obstacles modeled as ellipsoids, and the shaded blue boxes and arrows are the expected regions affected by gusts and the direction of gusts, respectively.

Procedure		**Process time**
Reachability Analysis	Analysis on the linearized dynamics	5.7 ms
	Analysis on the nonlinear dynamics	22.3 ms
	Etc.	0.2 ms
Trajectory optimization	Forward pass	11.8 ms
	Backward pass	52.4 ms
	Etc.	0.1 ms
Total		92.5 ms

Table 6.2: Mean computation time spent to generate one reference trajectory. The funnel of the linearized LTV system is used as the initial guess of the proposed algorithm that computes the funnel of the nonlinear system. The forward and backward passes in the trajectory optimization section represent routines of the DDP algorithm [45]. The cost of the trajectory planning problem (5.3) for the given $r(t)$ and $a(t)$ is computed during the forward pass. In the backward pass, a new sequence of $a(t)$ that decreases the cost of the trajectory planning problem (5.3) is updated. The forward and backward passes are iterated until the locally optimal sequence of $r(t)$ and $a(t)$ is obtained.

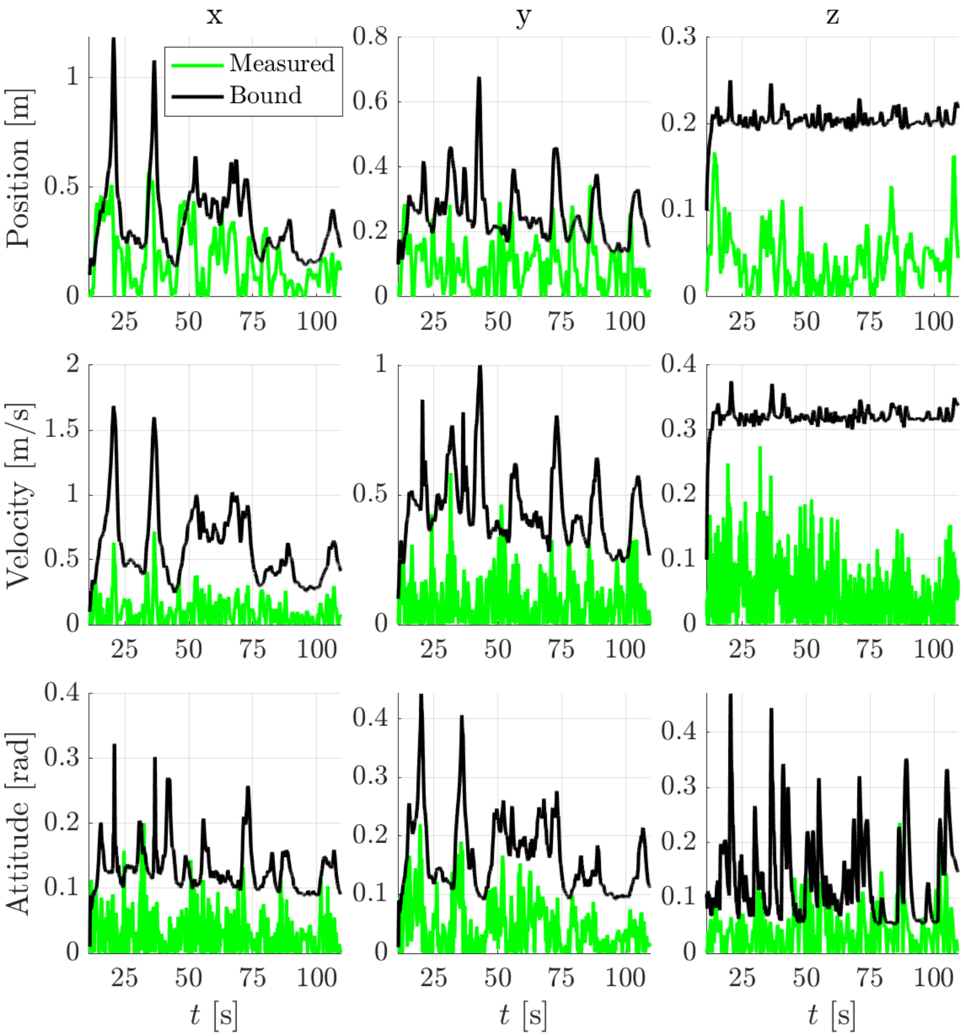


Figure 6.6: Deviation of states from the nominal trajectory. The green lines are the absolute value of the deviation, i.e. $|x_i(t) - q_i(t)|$ for all $i \in \{1, \dots, 9\}$. The black lines are the corresponding bounds from $Q(t)$. Each row represents the position, velocity, and attitude, respectively. Each column describes the corresponding state along x, y, and z axes, respectively.

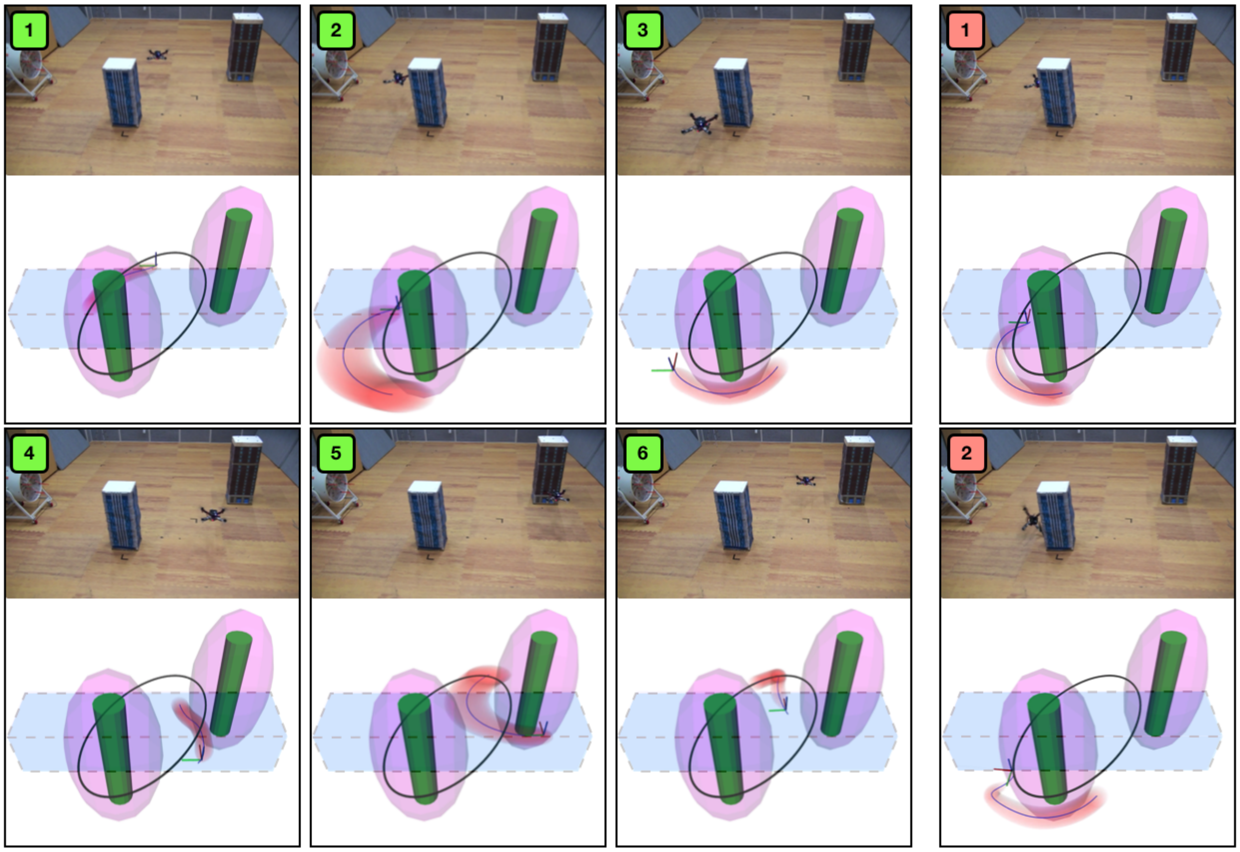


Figure 6.7: Comparison of flight experiments with and without updating the disturbance bound. The shaded blue region is the area affected by wind, and the green cylinders and magenta ellipsoids are the actual and inflated shape of obstacles. The blue line represents the optimized reference trajectory and the colored axes on the trajectory denote the current pose of the multirotor. The red ellipsoids represent the funnel of the multirotor. The thick black solid line denotes the predefined reference trajectory. The first three columns (numbered as 1–6 in green) represent the success case, and the last column (numbered as 1 and 2 in red) represents the failure case.

7

Conclusion

This dissertation presents a real-time safety-guaranteed trajectory planning method. The safeness of the planned trajectory is guaranteed through the reachability analysis. Given that unexpected disturbances can drive the system to risky regions instantaneously, this work concentrates on fast computation of the funnel so that the system can evaluate the danger of the current maneuver plan and swiftly modify the maneuver plan considering the funnel and unsafe regions.

For a real-time reachability analysis, the generalized Hopf formula is employed. Since the formula provides an explicit solution to the value function of the reachability problem, the sizable computational benefit can be derived from the formula compared to the grid-based methods such as the level-set method. The contribution of this dissertation in utilizing the generalized Hopf formula is the characterization of the shape of the FRS, which is proven to be the Minkowski sum of the initial set and the set due to disturbances. To reduce the computational burden regarding the Minkowski sum of sets, the conservative ellipsoidal approximation of the set due to disturbances is proposed. Furthermore, this dissertation presents the minimum volume composition algorithm to prevent overly conservative ellipsoidal approximation of the Minkowski sum of ellipsoids.

The main limitation of the generalized Hopf formula in computing the FRS is that the formula applies only to LTV systems. The FRS computed for the linearized dynamics cannot guarantee the reachability of the system due to the ignored nonlinearity. This dissertation presents the polynomial approximation of the value function of the HJB equation. To guarantee that the funnel encloses the FRS, the conservativeness condition of the polynomial value function is constructed. By using the property of the Bernstein polynomial, the conservativeness condition is converted to linear inequalities of the coefficients of the polynomial value function. The proposed funnel computation method is consist of multiple LPs, which requires much less computational burden compared to SOS program.

After computing the funnel, the locally optimal reference trajectory is found under the safety constraints. The constraints associated with the funnel enforce the system not to fall back into the unsafe area. Thanks to the dedicated computation of the funnel, the trajectory optimization problem can also be solved in real-time. The successful flight experiments in an environment disturbed by gusts validate that real-time safety-guaranteed planning is available through the proposed algorithm.

The possible future works of this study will be the integration with the backward reachability analysis. The backward reachable set, a set of states which is guaranteed to arrive at the desired set under all possible disturbances, will be useful to find trajectory candidates in environments where obstacles are densely distributed. It is expected that the conservativeness of the funnel will be reduced by combining the forward and backward reachable sets and this will improve the feasibility of the trajectory planning even in complicated environments.

Appendix

A. Dynamics of Unicycle

The state variables of the unicycle consist of the position in the horizontal plane (x_1 and x_2), speed (x_3), and heading angle (x_4). The inputs of this system are the acceleration (u_1) and angular velocity (u_2). The total six disturbances are considered, i.e. $m = 6$. The first two (w_1 and w_2) are the external disturbances in acceleration and angular velocity, respectively. The others (w_3, w_4, w_5, and w_6) are the disturbances in state measurement. The reference state of this system consists of the desired position in the horizontal plane (r_1 and r_2) and its time derivatives (r_3 and r_4).

The open-loop dynamics of the unicycle $f : \mathbb{R}^4 \times \mathbb{R}^2 \times \mathbb{R}^6 \to \mathbb{R}^4$ is

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

$$\begin{cases} \dot{x}_1 = x_3 \cos x_4, \\ \dot{x}_2 = x_3 \sin x_4, \\ \dot{x}_3 = u_1 + w_1, \\ \dot{x}_4 = u_2 + w_2. \end{cases}$$

To construct a reference-tracking controller, the error states are defined as

$$\hat{e} := \begin{bmatrix} x_1 + w_3 \\ x_2 + w_4 \end{bmatrix} - \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \in \mathbb{R}^2,$$

$$\dot{\hat{e}} = \begin{bmatrix} (x_3 + w_5)\cos(x_4 + w_6) \\ (x_3 + w_5)\sin(x_4 + w_6) \end{bmatrix} - \begin{bmatrix} r_3 \\ r_4 \end{bmatrix} \in \mathbb{R}^2.$$

The reference-tracking controller is formulated as

$$\begin{aligned} u(t) &= \mu(x(t), r(t), w(t)) \\ &= \frac{1}{v_0} G(x(t), w(t))(-K_p \hat{e}(t) - K_d \dot{\hat{e}}(t)), \end{aligned}$$

where the matrix $G(x, w) \in \mathbb{R}^{2\times 2}$ is

$$G(x, w) = \begin{bmatrix} (x_3 + w_5)\cos(x_4 + w_6) & (x_3 + x_5)\sin(x_4 + w_6) \\ -\sin(x_4 + w_6) & \cos(x_4 + w_6) \end{bmatrix},$$

$K_p = \mathrm{diag}(2, 2) \in \mathbb{R}^{2\times 2}$, $K_d = \mathrm{diag}(3, 3) \in \mathbb{R}^{2\times 2}$ are feedback gain matrices, and $v_0 = 0.5$ is used in the simulation. The dynamics of the reference state is

$$\dot{r}(t) = s(r(t), a(t)),$$

$$\dot{r}_1 = r_3, \quad \dot{r}_2 = r_4, \quad \dot{r}_3 = a_1, \quad \dot{r}_4 = a_2.$$

B. Dynamics of Multirotor

The state variables of the multirotor consist of the position $\mathbf{p} \in \mathbb{R}^3$, velocity $\mathbf{v} \in \mathbb{R}^3$, and Euler angles $\mathbf{q} \in \mathbb{R}^3$. The inputs are the thrust $F \in \mathbb{R}$ and angular velocity $\omega \in \mathbb{R}^3$. The disturbances of this system are $w = [\delta\mathbf{p}^\top, \delta\mathbf{v}^\top, \delta\mathbf{q}^\top, \Delta^\top]^\top \in \mathbb{R}^{12}$, where $\delta\mathbf{p} \in \mathbb{R}^3$, $\delta\mathbf{v} \in \mathbb{R}^3$ and $\delta\mathbf{q} \in \mathbb{R}^3$ are the errors in position, velocity, and attitude measurements, respectively. $\Delta \in \mathbb{R}^3$ is the external disturbance acting through the acceleration channel. The reference for this system consists of the desired position (r_1, r_2, and r_3) and velocity (r_4, r_5, and r_6).

The open-loop dynamics of the multirotor $f : \mathbb{R}^9 \times \mathbb{R}^4 \times \mathbb{R}^{12} \to \mathbb{R}^9$ is:

$$\dot{x}(t) = f\big(x(t), u(t), w(t)\big),$$

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{v}, \\ \dot{\mathbf{v}} &= \dfrac{F}{m} R(\mathbf{q}) e_3 + \mathbf{g} + \Delta, \\ \dot{\mathbf{q}} &= G(\mathbf{q}) \omega, \end{cases}$$

where $R : \mathbb{R}^3 \to \mathrm{SO}(3)$ is the rotation matrix from the Euler angles, $G : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ maps the rates of the Euler angles to the angular velocity, $\mathbf{g} \in \mathbb{R}^3$ is the gravitational acceleration, $m \in \mathbb{R}$ is the total mass of the multirotor, and $e_3 = [0,\ 0,\ 1]^\top$. A geometric controller [43] is used to generate the thrust and angular velocity commands. Similar to the unicycle case, the controller computes the inputs using the state perturbed by the measurement error. The controller can be parameterized with feedback gains as

$$u(t) = \mu(x(t), r(t), w(t); K),$$

where $K = [K_p^\top,\ K_v^\top,\ K_R^\top]^\top \in \mathbb{R}^9$ is the stack of position, velocity, and rotation feedback gains. $K_p = [6,\ 6,\ 10]^\top \in \mathbb{R}^3$, $K_v = [4,\ 4,\ 8]^\top \in \mathbb{R}^3$, and $K_R = [10,\ 10,\ 6]^\top \in \mathbb{R}^3$ are used in the experiment. The dynamics of the reference state is

$$\dot{r}(t) = s(r(t), a(t)),$$

$$\dot{r}_1 = r_4, \quad \dot{r}_2 = r_5, \quad \dot{r}_3 = r_6,$$

$$\dot{r}_4 = a_1, \quad \dot{r}_5 = a_2, \quad \dot{r}_6 = a_3.$$

References

[1] J. Darbon and S. Osher, “Algorithms for overcoming the curse of dimensionality for certain hamilton–jacobi equations arising in control theory and elsewhere,” *Research in the Mathematical Sciences*, vol. 3, no. 1, p. 19, 2016.

[2] C. Muñoz and A. Narkawicz, “Formalization of bernstein polynomials and applications to global optimization,” *Journal of Automated Reasoning*, vol. 51, no. 2, pp. 151–196, 2013.

[3] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, “Hamilton-jacobi reachability: A brief overview and recent advances,” in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 2242–2253.

[4] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, “Fastrack: a modular framework for fast and guaranteed safe motion planning,” in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 1517–1522.

[5] D. Fridovich-Keil, S. L. Herbert, J. F. Fisac, S. Deglurkar, and C. J. Tomlin, “Planning, fast and slow: A framework for adaptive real-time safe trajectory planning,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 387–394.

[6] D. Fridovich-Keil, J. F. Fisac, and C. J. Tomlin, “Safe and complete real-time planning and exploration in unknown environments,” *arXiv preprint arXiv:1811.07834*, 2018.

[7] M. M. Tobenkin, I. R. Manchester, and R. Tedrake, “Invariant funnels around trajectories using sum-of-squares programming,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 9218–9223, 2011.

[8] A. Majumdar and R. Tedrake, “Funnel libraries for real-time robust feedback motion planning,” *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[9] S. Kim, D. Falanga, and D. Scaramuzza, “Computing the forward reachable set for a multirotor under first-order aerodynamic effects,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2934–2941, 2018.

[10] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan, “Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots,” *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1419–1469, 2020.

[11] S. Kousik, B. Zhang, P. Zhao, and R. Vasudevan, “Safe, optimal, real-time trajectory planning with a parallel constrained bernstein algorithm,” *arXiv preprint arXiv:2003.01758*, 2020.

[12] M. E. Villanueva, B. Houska, and B. Chachuat, “Unified framework for the propagation of continuous-time enclosures for parametric nonlinear odes,” *Journal of Global Optimization*, vol. 62, no. 3, pp. 575–613, 2015.

[13] M. E. Villanueva, R. Quirynen, M. Diehl, B. Chachuat, and B. Houska, “Robust mpc via min–max differential inequalities,” *Automatica*, vol. 77, pp. 311–321, 2017.

[14] G. Garimella, M. Sheckells, J. L. Moore, and M. Kobilarov, “Robust obstacle avoidance using tube nmpc.” in *Robotics: Science and Systems*, 2018.

[15] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, “A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games,” *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[16] L. C. Evans, *Partial differential equations.* American Mathematical Society, 2010.

[17] E. Hopf, “Generalized solutions of non-linear equations of first order,” *Journal of Mathematics and Mechanics*, vol. 14, no. 6, pp. 951–973, 1965.

[18] M. R. Kirchner, R. Mar, G. Hewer, J. Darbon, S. Osher, and Y. T. Chow, “Time-optimal collaborative guidance using the generalized hopf formula,” *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 201–206, 2017.

[19] D. Lee and C. J. Tomlin, “Iterative method using the generalized Hopf formula: avoiding spatial discretization for computing solutions of Hamilton-Jacobi equations for nonlinear systems,” in *Decision and Control (CDC), 2016 IEEE 55th Conference on.* IEEE, 2019.

[20] D. Fridovich-Keil, J. F. Fisac, and C. J. Tomlin, “Safely probabilistically complete real-time planning and exploration in unknown environments,” in *2019 International Conference on Robotics and Automation (ICRA).* IEEE, 2019, pp. 7470–7476.

[21] M. Chen, S. L. Herbert, M. S. Vashishtha, S. Bansal, and C. J. Tomlin, “Decomposition of reachable sets and tubes for a class of nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3675–3688, 2018.

[22] S. L. Herbert, S. Bansal, S. Ghosh, and C. J. Tomlin, “Reachability-based safety guarantees using efficient initializations,” in *2019 IEEE 58th Conference on Decision and Control (CDC).* IEEE, 2019, pp. 4810–4816.

[23] I. M. Mitchell, “The flexible, extensible and efficient toolbox of level set methods,” *Journal of Scientific Computing*, vol. 35, no. 2-3, pp. 300–329, 2008.

[24] P.-L. Lions and J.-C. Rochet, “Hopf formula and multitime hamilton-jacobi equations,” *Proceedings of the American Mathematical Society*, vol. 96, no. 1, pp. 79–84, 1986.

[25] J. Darbon, “On convex finite-dimensional variational methods in imaging sciences and hamilton–jacobi equations,” *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2268–2293, 2015.

[26] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[27] K. D. Ikramov, "Conditions for unique solvability of the matrix equation ax+ x t b= c," in *Doklady Mathematics*, vol. 81, no. 1. Springer, 2010, pp. 63–65.

[28] J. Y. Ishihara and M. H. Terra, "On the lyapunov theorem for singular systems," *IEEE transactions on Automatic Control*, vol. 47, no. 11, pp. 1926–1930, 2002.

[29] A. Kurzhanski and T. Filippova, "On the theory of trajectory tubes—a mathematical formalism for uncertain dynamics, viability and control," in *Advances in nonlinear dynamics and control: a report from Russia.* Springer, 1993, pp. 122–188.

[30] A. Halder, "On the parameterized computation of minimum volume outer ellipsoid of minkowski sum of ellipsoids," in *2018 IEEE Conference on Decision and Control (CDC).* IEEE, 2018, pp. 4040–4045.

[31] C. Durieu, E. Walter, and B. Polyak, "Multi-input multi-output ellipsoidal state bounding," *Journal of optimization theory and applications*, vol. 111, no. 2, pp. 273–303, 2001.

[32] C. Durieu, B. T. Polyak, and E. Walter, "Trace versus determinant in ellipsoidal outer-bounding, with application to state estimation," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 3975–3980, 1996.

[33] S. Kim, H. Seo, S. Choi, and H. J. Kim, "Vision-guided aerial manipulation using a multirotor with a robotic arm," *IEEE/ASME Transactions On Mechatronics*, vol. 21, no. 4, pp. 1912–1923, 2016.

[34] M. Grant and S. Boyd, "Cvx: Matlab software for disciplined convex programming, version 2.1," 2014.

[35] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, “Introducing sostools: A general purpose sum of squares programming solver,” in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 1. IEEE, 2002, pp. 741–746.

[36] J. F. Sturm, “Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones,” *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[37] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *2011 IEEE international conference on robotics and automation.* IEEE, 2011, pp. 2520–2525.

[38] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[39] M. Stolle and C. G. Atkeson, “Policies based on trajectory libraries,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 3344–3349.

[40] B. Plancher, Z. Manchester, and S. Kuindersma, “Constrained unscented dynamic programming,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2017, pp. 5674–5680.

[41] N. Subbotina, “The method of characteristics for hamilton—jacobi equations and applications to dynamical optimization,” *Journal of mathematical sciences*, vol. 135, no. 3, pp. 2955–3091, 2006.

[42] H. Seo, C. Y. Son, D. Lee, and H. J. Kim, “Trajectory planning with safety guaranty for a multirotor based on the forward and backward reachability analysis,” in *2020 IEEE international conference on robotics and automation (ICRA).* IEEE, 2020, p. to appear.

[43] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in *Decision and Control (CDC), 2010 49th IEEE Conference on.* IEEE, 2010, pp. 5420–5425.

[44] S. Kim, S. Choi, H. Kim, J. Shin, H. Shim, and H. J. Kim, "Robust control of an equipment-added multirotor using disturbance observer," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1524–1531, 2018.

[45] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2012, pp. 4906–4913.

국문초록

로봇의 안정적인 운용을 위해서는 안전성이 보장된 경로를 계획하는 것이 중요하다. 하지만, 실제 환경에서는 예상치 못한 외란이 시스템에 영향을 주게 되므로 사전에 계획된 안전해 보이는 경로가 실제 런타임에서 안전한지 운용 전에 판단하는 것은 어렵다. 본 논문에서는 이러한 어려움을 극복하기 위한 도달 가능성 분석 기법과 이를 기반으로 안전성이 보장된 경로 계획 기법을 제안한다.

시스템의 안전 여부는 해밀턴-자코비 도달 가능성 분석을 통해 계산된 도달 가능 영역을 통해 보장될 수 있다. 실제 환경에서의 예상치 못한 외란이 시스템을 즉시 위험한 상황으로 유도할 수 있다는 점을 고려하여, 본 논문은 도달 가능성 분석의 실시간화를 목표로 한다. 본 연구에서 제안하는 도달 가능성 분석 알고리즘은 선형 시변 시스템과 비선형 시스템에 대한 두 종류의 기법들로 구성되어 있다. 선형 시변 시스템에 대한 도달 가능 영역에 대해서 본 연구는 일반화된 Hopf formula를 기반으로 하는 타원 근사를 제시한다. 해당 공식은 해밀턴-자코비-벨만 방정식의 해를 제공하기 때문에, 이를 기반으로 한 본 연구의 타원 근사는 선형 시변 시스템의 도달 가능 영역을 포함한다는 것이 보장된다. 선형 시변 시스템의 도달 가능 영역 근사 과정에서 고려되지 않은 비선형성과의 통합을 위해 본 연구는 해밀턴-자코비-벨만 방정식의 해(value function)를 다항 함수로 매개변수화하여 비선형 시스템의 도달 가능 영역을 근사화하는 알고리즘을 추가적으로 제안한다. 본 연구에서는 Bernstein 다항 함수의 특성을 활용함으로써 해당 value function의 보수성 조건이 선형 부등식 제약조건으로 변환될 수 있음을 보인다. 본 연구에서 제안하는 비선형 시스템의 도달 가능 영역에 대한 근사화 알고리즘은 복수의 선형 프로그램으로 구성되어 있으므로 기존의 도달 가능성 분석 기법들에 비해 보다 적은 계산 시간으로 도달 가능 영역을 근사화 할 수 있다.

마지막으로, 본 연구는 제안된 도달 가능성 분석 결과를 기반으로 안전성이 보장된 경로 계획에 대한 최적화 문제를 설정한다. 이 때, 시스템의 안전은 근사화된 도달 가능 영역과 안전하지 않은 영역이 겹쳐지지 않도록 하는 비선형 제약조건으로 고려된다. 본 연구에서 제안한 도달 가능 영역의 근사 결과는 실제 도달 가능 영역을 포함하기 때문에, 해당 제약조건을

만족하는 경로를 따라 운용되는 시스템은 외란에 관계없이 안전이 보장된다. 또한, 제안된 도달 가능 영역 분석 알고리즘은 실제 도달 가능 영역을 빠르게 근사화 할 수 있으므로 안전성이 보장된 경로의 실시간 최적화를 가능하게 한다. 결과적으로, 본 연구가 제안하는 도달 가능성 분석 및 경로 계획 알고리즘을 통해 런타임 중 시스템에 예상치 못한 외란이 작용되는 경우에도 시스템의 안전이 보장되도록 하는 실시간 경로 수정이 가능하다. 바람이 많이 부는 환경에 대한 드론의 장애물 회피 비행 실험들을 통해 본 연구에서 제안한 알고리즘의 타당성 및 실시간성을 검증한다.

주요어: 안전 보장, 경로 계획, 최적화 및 최적 제어, 도달 가능성 분석

학 번: 2016-30185