

UC Santa Cruz

UC Santa Cruz Electronic Theses and Dissertations

Title

A Geometric Approach for Learning Reach Sets

Permalink

<https://escholarship.org/uc/item/3xc0x966>

Author

Haddad, Shadi

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

A GEOMETRIC APPROACH FOR LEARNING REACH SETS

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

DOCTOR OF PHILOSOPHY

in

APPLIED MATHEMATICS

by

Shadi Haddad

March 2024

The dissertation of Shadi Haddad
is approved:

Associate Professor Abhishek Halder, Chair

Professor Hongyun Wang

Professor Qi Gong

Peter Biehl
Vice Provost and Dean of Graduate Studies

Contents

Abstract	vii
1 Introduction	1
1.1 Reachability Analysis	2
1.2 Novelty w.r.t. Existing Literature	3
1.3 Technical Contributions	6
1.4 Organization	8
1.5 Notations	12
1.6 Preliminaries	14
1.6.1 Full state feedback linearizable systems	14
1.6.2 State transition matrix	19
1.6.3 Support function	19
1.6.4 Polar dual	22
1.6.5 Vector measure	23
1.6.6 Zonotope	23
1.6.7 Variety and ideal	24
1.6.8 Minkowski difference	24
1.6.9 Induced norm	25
1.7 Integrator Reach Set with Compact Input Set \mathcal{U}	25
2 Integrator Reach Sets with Time Invariant Set-Valued Uncertainties	28
2.1 Taxonomy and Boundary	32
2.1.1 $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ is a Zonoid	33
2.1.2 $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ is semialgebraic	37
2.1.3 Implicitization of $\partial\mathcal{X}_t^\square(\{\mathbf{x}_0\})$	38
2.1.4 Polar dual of $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$	41
2.1.5 Summary of taxonomy	42
2.2 Size of $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ with Time Invariant Set-Valued Inputs	45
2.2.1 Volume	45
2.2.2 Diameter	47
2.2.3 Scaling of the volume	50
2.2.4 Scaling of the diameter	52

2.3	Benchmarking Over-approximations of Integrator Reach Sets	54
3	Integrator Reach Sets with Time Varying Set-Valued Uncertainties	59
3.1	Auxiliary Definitions	59
3.1.1	Support function	61
3.2	Size of $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ with Time Varying Set-Valued Inputs	61
3.2.1	Boundary of \mathcal{X}_t^\square with time varying set-valued inputs	62
3.2.2	Volume of $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ with time varying set-valued inputs	64
3.3	Recovering $\partial\mathcal{X}_t^\square$ for Time Invariant \mathcal{U}	65
4	Applications	68
4.1	Exact Computation of LTI Reach Set	68
4.1.1	Brunovsky normal form and LTI reach set	68
4.1.2	Semi-analytical algorithm	70
4.1.3	Input set in Brunovsky coordinates	71
4.1.4	Example for single input LTI reach set	74
4.2	Intersection Detection	76
4.2.1	Certifying $\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=)\emptyset$	77
4.2.2	The case when $\mathcal{U}^A, \mathcal{U}^B$ are norm balls	79
4.2.3	The Case when $\mathcal{U}^A, \mathcal{U}^B$ are hyperrectangles	83
4.2.4	Numerical example for intersection detection	86
4.3	Learning Reach Set for Full State Feedback Linearizable Systems	87
4.3.1	Approximating feedback linearizable reach set via scenario optimization	89
4.3.2	Probabilistic inclusion	93
4.3.3	Example: learning the reach set of a static state feedback linearizable system with $\mathbf{r} = (3, 2)^\top$	94
4.3.4	Example: learning the reach set of a static state feedback linearizable system with $r = d = 4$	97
4.4	Boundary of the Minkowski Sum of \mathcal{X}_t^\square with a Line Segment	99
4.4.1	Example: boundary of the Minkowski sum of $\mathcal{X}_t(\{\mathbf{x}_0\})$ with a line segment embedded in 2D space	102
4.4.2	Example: Boundary of the Minkowski sum of $\mathcal{X}_t(\{\mathbf{x}_0\})$ with a line segment embedded in 3D space	104
4.4.3	Example: learning the reach set of a dynamic state feedback linearizable system	106
4.5	A Note on Monte Carlo vs Feedback Linearization	111
4.6	Parallelization	113
5	The Hausdorff Distance between Norm Balls and their Linear Maps	119
5.1	Hausdorff Distance between Unit Norm Balls	122
5.1.1	Hausdorff distance between polyhedral D -norm balls	124
5.2	Composition with a Linear Map	127

5.2.1	Estimates for arbitrary T	128
5.2.2	Estimates for random T	132
5.3	Integral Version and Application	133
6	Support Function Representation Learning of Compact Sets	138
6.1	Learning Algorithms	139
6.1.1	Data generation	139
6.1.2	QP algorithm	140
6.1.3	ISNN algorithm	141
6.2	Case Studies	143
6.2.1	p -Sum of ellipsoids	144
6.2.2	Reach sets of linear time-varying (LTV) systems	145
6.2.3	Collision detection	149
6.2.4	Reach set of deep neural network	151
6.2.5	ROA of nonlinear systems	157
6.2.6	Control invariant set	158
7	Summary and Future Work	161
7.1	Summary	161
7.2	Future Work	163
A	Proofs for Chapter 1	165
A.1	Proof of Lemma 1.1	165
A.2	Proof of Lemma 1.2	165
A.3	Proof of Lemma 1.3	165
A.4	Proof of Proposition 1.1	166
B	Proofs for Chapter 2	167
B.1	Proof of Theorem 2.1	167
B.2	Proof of Theorem 2.2	167
B.3	Proof of Proposition 2.3	168
B.4	Proof of Corollary 2.4	169
B.5	Proof of Theorem 2.5	170
B.6	Proof of Theorem 2.6	171
B.7	Proof of Theorem 2.7	174
C	Proofs for Chapter 3	176
C.1	Proof of Theorem 3.1	176
C.2	Proof of Theorem 3.2	176
D	Proofs for Chapter 4	177
D.1	Proof of Lemma 4.1	177
D.2	Proof of Theorem 4.1	178

D.3	Proof of Theorem 4.2	179
D.4	Proof of Theorem 4.3	180
D.5	Proof of Theorem 4.5	180
D.6	Proof of Theorem 4.6	181
D.7	Proof of Theorem 4.7	182
E	Proofs for Chapter 5	184
E.1	Proof of Proposition 5.2	184
E.2	Proof of Proposition 5.3	185
E.3	Proof of Proposition 5.5	185
E.4	Proof of Theorem 5.2	186
E.5	Proof of Corollary 5.3	186
F	Proofs for Chapter 6	187
F.1	Proof of Theorem 6.1	187
F.2	Proof of Theorem 6.2	187
	Bibliography	202

Abstract

A Geometric Approach for Learning Reach Sets

Shadi Haddad

Reachability analysis is a method to guarantee the performance of safety-critical applications such as automated driving and robotics against dynamic uncertainties. The main object of study is the reach set, defined as the set of states that a controlled dynamical system may reach at a future time, depending on a set-valued evolution of uncertainties. We develop the theory and algorithms for learning the reach sets of full state feedback linearizable systems—an important class of nonlinear control systems, common in vehicular applications such as automobiles and drones. These reach sets, in very general settings, are compact but nonconvex. The new idea we propose is to compute these reach sets in the associated Brunovsky normal coordinates, and then transform the sets back to the original coordinates via known diffeomorphisms. Our algorithms exploit learning-theoretic ideas to provide probabilistic guarantees on the computed sets.

As a by-product of our analysis, we uncover the exact geometry of the integrator reach set with compact set-valued inputs. These exact results include the closed-form parametric and implicit formulae for the boundaries, volumes, and widths of the integrator reach sets. The exact parametric formula for the boundary admits an integral representation involving the boundary of the compact input sets. The exact implicit formula is given by the vanishing of certain Hankel determinants. These results on integrators should be of independent interest, serving as benchmarks for quantifying the conservatism in reach set computation algorithms.

Our geometric analysis also helps clarify a taxonomy, i.e., what kind of compact convex sets can the integrator reach sets be. We show that the integrator reach sets

resulting from arbitrary, time-invariant, compact input sets are zonoids and semialgebraic, but not spectrahedra. The integrator reach sets resulting from arbitrary, time-varying, compact input sets are shown to be zonoids but not semialgebraic in general.

We detail how these geometric results enable the semi-analytical computation of the reach set of any controllable linear time invariant system, as well as the reach sets of full state feedback linearizable systems.

Leveraging an Isomorphism between compact sets and their support functions, we also propose a data-driven method for learning any general compact set.

This is useful for learning compact sets such as reach sets, maximal control invariant sets, region-of-attraction that are related to an underlying nonlinear dynamical system but an analytic model for the dynamical nonlinearities are unavailable. Our results show that the proposed geometric learning ideas can be efficient when we only have access to simulated or experimentally observed data. We demonstrate computational learning of these compact sets by carrying out regression analyses on their support functions using finite data sets. Finally, we outline the directions for future research.

1 | Introduction

In the context of systems control, there exists a substantial literature on estimating specialized categories of sets that hold computational significance, such as the reach set [1–6], the region-of-attraction (ROA) [7–9], and the maximal controlled invariant set (MCIS) [10–14]. A *reach set* is defined as the set of states a controlled dynamical system may reach at a future time—subject to set-valued uncertainties in initial condition, input and disturbance. A *region-of-attraction* describes the set of initial states leading to a stable equilibrium point. In addition, the *maximal controlled invariant set* (MCIS), is also known colloquially as a *viability kernel*: A set $\mathcal{C} \subset \mathcal{R}^d$ is a *control-invariant* (CI) for (6.29), if for any initial condition $\mathbf{x}_0 \in \mathcal{C}$, there exist a control trajectory $\{\mathbf{u}(t)\}_{t=0}^{\infty}$ such that dynamic (6.29) stays inside \mathcal{C} for all $t \geq 0$. The set-valued framework allows robustness against uncertainty in decision making.

Due to limitations faced by classical methods for computing these sets, recent research endeavors have shifted towards learning-based approaches. Such approaches aim to approximate these sets through machine learning algorithms and data-driven techniques [15–20]. Learning a set is of importance in several data-driven decision making applications, both in and outside systems-control; see e.g., [21–28].

The dissertation at hand aims to contribute to this evolving landscape by developing a geometric approach for learning compact sets. The initial stage of this research centers on devising a novel geometric approach for learning the reach set of dynamical systems under set-valued input uncertainties. This foundational work serves as a stepping stone for the subsequent phase, where the scope broadens to incorporate learning any general compact set. In particular, we learn a compact set up to the closure of its convexification, from finite data.

1.1 Reachability Analysis

Consider a deterministic, finite dimensional, nonlinear control system given by a controlled ODE initial value problem

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, \mathbf{v}), \quad \mathbf{z}(t=0) \in \mathbb{R}^{d_z}, \quad \mathbf{v} \in \mathcal{V} \subset \mathbb{R}^m, \quad (1.1)$$

with d_z states and m inputs, where the initial state $\mathbf{z}(t=0)$ is given, and the known input set \mathcal{V} is compact. The (*forward*) *reach set* \mathcal{Z}_t of (1.1) is defined as the set of states the system (1.1) may reach at time t . Specifically,

$$\mathcal{Z}_t := \bigcup_{\text{measurable } \mathbf{v}(\cdot) \in \mathcal{V}} \{ \mathbf{z}(t) \in \mathbb{R}^{d_z} \mid \dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, \mathbf{v}), \mathbf{z}(t=0) \in \mathbb{R}^{d_z}, \mathbf{v}(\tau) \in \mathcal{V} \text{ for all } 0 \leq \tau \leq t \}. \quad (1.2)$$

The research objective of this work is to explore the geometry of the *reach set* \mathcal{Z}_t for certain special classes of the controlled vector field \mathbf{f} .

Reach set computation finds broad applicability in safety critical applications, from biomedical to engineering. Loosely speaking, reachability analysis is performed to ensure that the states of a controlled system remain inside certain regions of the state space that are guaranteed to be safe or to reach a desired target. Motion planning in biomedical applications such as needle steering [29, 30] and multilumen transoral lung access [31] uses reachability analysis to ensure that operations stay within a safety margin during medical interventions. Reachability analysis has also been applied extensively to both autonomous driving [32] and air traffic control [33] to avoid collisions without sacrificing efficiency. In chemical engineering, reachability analysis is critical for determining the achievable end-point products of complex processes such as emulsion polymerizations (crystallization, precipitation, and granulation) [34].

The research perspective and the geometric approach we pursue break away from

existing literature, which is based on a philosophy of generalization, i.e., the design and analysis of reach set computation algorithms for the most general class of possible systems. In this work, we instead focus on specificity, i.e., we investigate the geometry of the true reach set \mathcal{Z}_t for specific classes of systems. Our research vision is that this change in disposition, from the generic to the specific, can uncover a new genre of theory and algorithms for reach sets.

1.2 Novelty w.r.t. Existing Literature

The existing reach set literature, broadly speaking, either provides very generic statements (e.g., some class of reach sets are convex and/or compact) or gives specific numerical algorithms of different types, viz. *parametric algorithms*, such as ellipsoidal or zonotopic over-approximation [35, 36]; *nonparametric algorithms*, such as level set computation [37]; or *semiparametric algorithms*, such as sample-based statistical learning [38, 39].

In a *parametric algorithm*, the idea is to fix a simple geometric shape primitive (e.g., “ellipsoid”, “zonotope”, etc.) beforehand and to over-approximate the reach set within that class. To guarantee safety, one would like to ensure that the true reach set is always contained in the approximated set, i.e., the computed set should be an over-approximation. At the same time, it is desired to minimize the conservatism, i.e., the over-approximation must be “tight”. This naturally leads to an optimization problem whose decision variables are the parameters of the shape primitive, and the objective is to minimize a suitable functional (e.g., volume) of the over-approximating set subject to the constraint that the true reach set is contained in the approximating set. Two specific implementations of such parametric algorithms are provided by the ellipsoidal toolbox [35] and the CORA toolbox [36]: The first uses “ellipsoid” while

the latter uses “zonotope” as the over-approximating shape primitive.

An example of a *nonparametric algorithm* is the level set toolbox [37], which exploits the fact that the (forward) reach set \mathcal{Z}_t in (1.2) can be seen as the zero sublevel set of the viscosity solution $V(t, \mathbf{z})$ of certain Hamilton-Jacobi-Bellman (HJB) partial differential equation associated with the controlled dynamics (1.1). Specifically, given the HJB equation

$$\frac{\partial V}{\partial t} + \max_{\mathbf{v} \in \mathcal{V}} \left\langle \frac{\partial V}{\partial \mathbf{z}}, \mathbf{f}(\mathbf{z}, \mathbf{v}) \right\rangle = 0,$$

such that $V(t=0, \mathbf{z}) = \|\mathbf{z} - \mathbf{z}(t=0)\|_2^2,$ (1.3)

the forward reach set (1.2) satisfies

$$\mathcal{Z}_t = \left\{ \mathbf{z} \in \mathbb{R}^{d_z} \mid \underbrace{V(t, \mathbf{z}) \leq 0}_{\text{zero sublevel sets of the solution of (1.3)}} \right\}. \quad (1.4)$$

Semiparametric algorithms such as [38–40] estimate the reach set from the samples of the trajectory with statistical performance guarantees. The advantage here is that no explicit dynamical model is needed; only a trajectory generation oracle is assumed to be available. The disadvantage, however, is that one can only hope to provide guarantees about the quality of approximation in batch learning, i.e., in a probably approximately correct (PAC) sense. In particular, the approximating set may no longer be an over-approximation of the true reach set, i.e., one trades off the number of samples against a probability of violating safety.

None of the methods mentioned above give any specific algebraic or topological results about the ground truth, i.e., the geometry of the true reach set that we are numerically approximating. For instance, even for linear time invariant systems, the existing literature does not provide the exact gap between the volume of the true reach set and the volume of the set returned by the approximation oracles.

In the absence of ground truth, given two different over-approximation algorithms,

we cannot quantify the comparative performance of one algorithm against the other. Neither the graphical visualizations such as low dimensional projection plots for the reach set, nor the Monte Carlo algorithms can resolve this issue.

Standing on this research landscape, we investigate the geometry of the reach sets of full state feedback linearizable systems—a subclass of nonlinear control systems common in engineering applications. If a system is full state feedback linearizable, then it is possible to diffeomorphically map the system into a canonical linear control system known as the *Brunovsky normal form*. The Brunovsky normal form is a chain of integrators, where different integrators may have different degrees. We sometimes refer to the Brunovsky normal form as the “integrator dynamics”.

Since the Brunovsky normal form, a.k.a. the integrator dynamics, is a special linear time invariant system, a natural question is whether it is possible to get the exact geometry of the reach set in normal coordinates, and then transform that set back to the original state coordinates via known diffeomorphism. This is the first technical approach we have pursued.

Motivated by our geometric approach, we then aim to broaden our focus to understand the reach set of any general compact set where the underlying dynamics are either unknown or analytically complex to manage. We introduce a novel representation for the learning of compact sets, up to the closure of their convexification. We posit that the key to learn a compact set lies in learning its support function. The main idea behind the proposed approach is rooted in an isomorphism between the space of sublinear functions and the space of finite dimensional compact convex sets, which allows transcribing set operations of system-control interest to exact functional operations on the corresponding support functions—the latter being computationally amenable via support function calculus. Given finite data collected from numerical simulation or from experimental measurements, we outline two algorithms to learn

the associated support functions via convex and nonconvex programming, respectively.

1.3 Technical Contributions

Following the novel geometric approach outlined above for full state feedback linearizable systems, we make the following contributions.

- i We derive the closed form parametric as well as the implicit formula for the boundary of an integrator reach set.
- ii We deduce closed form formula for the volume and diameter of an integrator reach set.
- iii We derive scaling laws for the behavior of diameter and volume of the integrator reach set with respect to time and state space dimension.
- iv We establish that the integrator reach set is a zonoid, semialgebraic, and not a spectrahedron.
- v We derive explicit formulas for the support functions and boundaries of the integrator reach sets subject to time-varying compact set-valued uncertainties in the inputs. Such uncertainties appear in dynamic state feedback linearizable systems even when such systems are subjected to time-variant set-valued input uncertainties.
- vi We provide algorithms to detect intersections between multiple integrator reach sets, thereby allowing intersection detection between the reach sets of the corresponding feedback linearizable systems.

- vii We present a semi-analytical method for exact computation of the boundary of the reach set of a single-input controllable linear time invariant (LTI) system with given bounds on its input range.
- viii We propose a probabilistic learning approach to estimate the reach sets of feedback linearizable systems.
- ix We consider the problem of computing the (two-sided) Hausdorff distance between the unit ℓ_{p_1} and ℓ_{p_2} norm balls in finite dimensional Euclidean space. We then establish a connection between this particular problem and the problem of computing the Hausdorff distance between the reach sets of a linear dynamical system with different unit norm ball-valued input uncertainties.
- x We deduce a closed-form formula for the Hausdorff distance between unit ℓ_{p_1} and ℓ_{p_2} norm balls in \mathbb{R}^d for $1 \leq p_1 < p_2 \leq \infty$, i.e., a formula for $\delta(\mathbb{B}_{p_1}^d, \mathbb{B}_{p_2}^d)$, and provide details on the landscape of the corresponding nonconvex optimization objective.
- xi We derive upper bound for Hausdorff distance between the common linear transforms of the ℓ_p and ℓ_q norm balls. We point out a class of linear maps for which the aforesaid closed-form formula for the Hausdorff distance is recovered, thereby broadening the applicability of the formula.
- xii Bringing together results from the random matrix theory literature, we provide upper bounds for the expected Hausdorff distance when the linear map is random with independent mean-zero entries for two cases: when the entries have magnitude less than unity, and when the entries are standard Gaussian.
- xiii We provide certain generalization of the aforesaid formulation by considering the Hausdorff distance between two set-valued integrals.

- xiv We propose learning a compact set by learning its support function from the (possibly noisy) elements of that set available as finite data and discuss how our approach is computationally more beneficial other set representation alternatives.
- xv We present two algorithms to learn the support function via sublinear regression: convex quadratic programming, and training an input sublinear neural network that involves nonconvex programming.

1.4 Organization

This dissertation is structured as follows. In Ch. 1.6.1, we provide the requisite backgrounds on the full state feedback linearizable control systems. We consider both static and dynamics state feedbacks. We also clarify their connections with the differentially flat systems. Within Ch. 1.6, we provide an overview of the fundamental mathematical concepts that play a role in guiding the entirety of our research. In Ch. 1.7, we briefly introduce the reach set of the integrator dynamics risen from any general compact set-valued input uncertainty.

In Ch. 2, we undertake an investigation of the geometry of the integrator reach sets with time invariant set-valued inputs. After reviewing some preliminary concepts, the results on the taxonomy and the boundary of these reach sets are provided in Ch. 2.1.

In Ch. 2.1.1, guided by the support function structure, we show that the integrator reach set is a zonoid, defined as the Hausdorff limiting set of a sequence of zonotopes. Then, in Ch. 2.1.1, we derive the closed form equations for the boundary. We do so by exploiting that the Legendre-Fenchel conjugate of the support function, yielding the indicator function of the set. Exploiting the boundary formula, we show in Ch. 2.1.2 that the reach set is semialgebraic.

The results on the size of the reach set are collected in Ch. 2.2. We obtain the closed form formulas for the volume and the diameter of the set in Ch. 2.2.1 and Ch. 2.2.2, respectively. In Sections 2.2.3 and 2.2.4, we discuss the scaling laws governing the behavior of the volume and the diameter with respect to time and state space dimension. The application of these results for benchmarking the reach set over-approximation algorithms are discussed in Ch. 2.3.

In Chapter 3, we generalize our developments in Chapter 2. In Ch. 2.1, we derive the explicit formula for the support function and boundaries of the integrator reach set resulting from *time-varying* compact set-valued uncertainties in the inputs. These results, then, become prime movers to enable the computational applications in Chapter. 4, which includes intersections detection between multiple integrator reach sets (Ch. 4.2), semi-analytical computation of LTI reach sets (Ch. 4.1) and learning the reachset of the feedback linearizable systems (4.3). Illustrative numerical examples are interspersed within all 3 sections.

Specifically, in Ch. 4.1, we introduce a semi-analytical approach that enables the precise computation of the boundary of the reach set for a single-input, controllable linear time-invariant (LTI) system, taking into account the constraints on its input range.

Then, in Ch. 4.2, we discuss how building upon the geometric results in Chapter 3, the intersection between differentially flat systems can be certified or falsified by detecting intersection of their corresponding Brunovsky normal forms. Ch. 4.3 explores the computation of the forward reach sets for differentially flat nonlinear systems via estimating the reach set in the corresponding integrator coordinates achieved using known diffeomorphism.

In Ch. 4.6, we perform a complexity analysis for the learning method introduced in Ch. 4.3.

In the interest of computational tractability, in most part of this research we assume that the input uncertainty for the integrator reach are box-valued, characterized by the ℓ_∞ norm ball. This is despite the possibility that the true nature of these sets could adhere to an ℓ_p norm ball, where $0 < p < \infty$. Such approximations in defining the input uncertainty sets result in an over-estimation of the reach sets, as elaborated in Chapter 2. In this context, measuring the degree of over-approximation is tantamount to calculating the Hausdorff distance between the respective reach sets. Driven by this understanding, Chapter 5 focuses on estimating the Hausdorff distance between the reach set of linear time-invariant systems, which include the integrator, emerging from different norm-valued input uncertainties. The rest of this paragraph sheds light on the novel contributions stemming from this particular avenue of our research.

First, we consider the problem of computing the (two-sided) Hausdorff distance between the unit ℓ_{p_1} and ℓ_{p_2} norm balls in finite dimensional Euclidean space for $1 \leq p_1 < p_2 \leq \infty$, and derive a closed-form formula for the same (Ch. 5.1). We also derive a closed-form formula for the Hausdorff distance between the k_1 and k_2 unit D -norm balls, which are certain polyhedral norm balls in d dimensions for $1 \leq k_1 < k_2 \leq d$ (Ch. 5.1.1). When two different ℓ_p norm balls are transformed via a common linear map, we obtain several estimates for the Hausdorff distance between the resulting convex sets. These estimates upper bound the Hausdorff distance or its expectation, depending on whether the linear map is arbitrary or random (Ch. 5.2). We then generalize the developments for the Hausdorff distance between two set-valued integrals obtained by applying a parametric family of linear maps to different ℓ_p unit norm balls, and then taking the Minkowski sums of the resulting sets in a limiting sense. To illustrate an application, we show that the problem of computing the Hausdorff distance between the reach sets of a linear dynamical system with different unit norm ball-

valued input uncertainties, reduces to this set-valued integral setting (Ch. 5.3). The organization is as follows. In Ch. 5.1, we consider the Hausdorff distance between unit norm balls for two cases: ℓ_p norm balls for different p , and D -norm balls parameterized by different parameter k . We discuss the landscape of the corresponding nonconvex optimization problem and derive closed-form formula for the Hausdorff distance. Ch. 5.2 considers the Hausdorff distance between the common linear transformation of different ℓ_p norm balls, and bounds the same when the linear map is either arbitrary or random. In Ch. 5.3, we consider an integral version of the problem considered in Ch. 5.2 and illustrate one application in controlled linear dynamical systems with set-valued input uncertainties where this structure appears. These results could be of independent interest.

In chapter (6), we extend our focus from the reach set of LTI systems, to any general compact set. Utilizing the isomorphism between the space of sublinear functions and the space of finite dimensional compact convex sets and pose that learning a compact set (up to the closure of its convexification) it's equivalent to learning its support function. We present two algorithms (Ch. 6.1) to learn the support function via sublinear regression: convex quadratic programming (Ch. 6.1.2), and training an input sublinear neural network that involves nonconvex programming (Ch. 6.1.3). We demonstrate the comparative performance of these algorithms via various numerical examples In Ch. 6.2.

All proofs are deferred to the Appendix. The summary of our research and the future work directions are provided in Chapter 7.

1.5 Notations

Most notations are introduced in situ. We use $\llbracket n \rrbracket := \{1, 2, \dots, n\}$ to denote the set of natural numbers from 1 to n . Boldfaced lowercase and boldfaced uppercase letters are used to denote the vectors and matrices, respectively. The symbol \mathbb{E} denotes the mathematical expectation, $\text{card}(\cdot)$ denotes the cardinality of a set, the superscript \top denotes matrix transpose, and the superscript \dagger denotes the appropriate pseudo-inverse. For a column vector $\boldsymbol{x} \in \mathbb{R}^d$ whose components are differentiable with respect to (w.r.t.) a scalar parameter t , the symbol $\dot{\boldsymbol{x}}$ denotes componentwise derivative of \boldsymbol{x} w.r.t. t . The notation $\lfloor \cdot \rfloor$ stands for the floor function that returns the greatest integer less than or equal to its real argument. The function $\exp(\cdot)$ with matrix argument denotes the matrix exponential. The inequality \geq is to be understood in Löwner sense; e.g., saying \boldsymbol{S} is a symmetric positive semidefinite matrix is equivalent to stating $\boldsymbol{S} \geq \mathbf{0}$. Please also refer to table 1 for the list of some commonly used notations.

Notation	Definition	First appeared
z	state vector of the original system in \mathbb{R}^{d_z}	(1.5) and (4.1)
v	control vector of the original system	(1.5) and (4.1)
x	state vector of the integrator system \mathbb{R}^d	(1.6)
r	relative degree vector	(1.7b)
ρ	augmented states	(1.9)
ρ	augmented states	(1.9)
w	compensator states in \mathbb{R}^{d_w}	(1.9)
ρ	augmented states	(1.9)
u	control vector of the integrator system	(1.6)
A	Brunovsky normal form state coefficient matrix	(1.6)
B	Brunovsky normal form input coefficient matrix	(1.6)
τ	state diffeomorphism $z \rightarrow x$	(1.9)
τ_u	input homeomorphism $v \rightarrow u$	(1.9)
Π_z	Projection operator into z coordinates	(1.9)
e_k^ℓ	k th basis vector of in \mathbb{R}^ℓ	(1.9)
C, d	Nonlinear functions describing the input mapping	(1.8)
Φ	integrator's state transition matrix	(1.8)
$h_{\mathcal{X}}(\mathbf{y})$	support function of set \mathcal{X} in direction \mathbf{y}	(1.23)

1.6 Preliminaries

In the following, we summarize some preliminaries which will be useful in the main body and in the Appendix.

1.6.1 Full state feedback linearizable systems

Consider a continuous time nonlinear control system

$$\dot{z} = \mathbf{f}(z, \mathbf{v}), \quad z \in \mathbb{R}^{d_z}, \quad \mathbf{v} \in \mathcal{V} \subset \mathbb{R}^m, \quad (1.5)$$

where z is the state vector, and \mathbf{v} is the control input vector. System (1.5) is differentially flat or equivalently *feedback linearizable* [41] if there exists a “flat output” vector $\mathbf{y} \in \mathbb{R}^m$ such that the state $z \in \mathbb{R}^{d_z}$ and the input $\mathbf{v}(t) \in \mathbb{R}^m$ can be derived from the flat output \mathbf{y} without any integration. A differentially flat control system is *dynamic feedback linearizable* on an open dense set [41].

A system is called dynamic feedback linearizable if there exist compensator states $\mathbf{w} \in \mathbb{R}^{d_w}$ such that the dynamics in the augmented state vector $\boldsymbol{\rho} := (z, \mathbf{w})$ comprising of the original states and the compensator states, is full state feedback linearizable. That is to say, there exists a diffeomorphism τ that maps the augmented state-control pair $\boldsymbol{\rho} \in \mathbb{R}^{d_z+d_w}$ to *Brunovsky normal coordinates*, $\mathbf{x} \in \mathbb{R}^d$, known as the *integrator dynamics*, given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(t) \in \mathbb{R}^d, \quad \mathbf{u}(t) \in \mathcal{U}(t) \subset \mathbb{R}^m, \quad (1.6)$$

where the set $\mathcal{U}(t)$ is compact, and

$$\mathbf{A} := \text{blkdiag}(\mathbf{A}_1, \dots, \mathbf{A}_m), \quad \mathbf{B} := \text{blkdiag}(\mathbf{b}_1, \dots, \mathbf{b}_m), \quad (1.7a)$$

$$\mathbf{A}_j := \left(\mathbf{0}_{r_j \times 1} \mid \mathbf{e}_1^{r_j} \mid \mathbf{e}_2^{r_j} \mid \dots \mid \mathbf{e}_{r_j-1}^{r_j} \right), \quad \mathbf{b}_j := \mathbf{e}_{r_j}^{r_j}, \quad \text{for } j \in [m] := \{1, \dots, m\}. \quad (1.7b)$$

The relative degree vector $\mathbf{r} = (r_1, r_2, \dots, r_m)^\top \in \mathbb{Z}_+^m$ is a vector of positive integers

satisfying $r_1 + r_2 + \dots + r_m = d$. In (1.7a), the symbol $\text{blkdiag}(\cdot)$ denotes a block diagonal matrix whose arguments constitute its diagonal blocks. In (1.7b), the notation $\mathbf{0}_{r_j \times 1}$ stands for the $r_j \times 1$ column vector of zeros, and e_k^ℓ denotes the k th basis (column) vector in \mathbb{R}^ℓ for $k \leq \ell$.

The control mapping $\tau_u : (\mathbf{v}, \mathbf{z}, \mathbf{w}, \dot{\mathbf{w}}, \ddot{\mathbf{w}}, \dots) \mapsto \mathbf{u}$ is guaranteed to be a homeomorphism, and can be written explicitly as

$$\mathbf{u}(t) = \mathbf{C}(\mathbf{z}(t), \mathbf{w}(t), \dot{\mathbf{w}}, \ddot{\mathbf{w}}, \dots) \mathbf{v}(t) + \mathbf{d}(\mathbf{z}(t), \mathbf{w}(t), \dot{\mathbf{w}}, \ddot{\mathbf{w}}, \dots), \quad (1.8a)$$

$$\dot{\mathbf{w}}(t) = (\mathbf{z}, \mathbf{w}, \mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, \dots), \quad \forall t \geq 0. \quad (1.8b)$$

Note that the inverse mapping from the integrator coordinates \mathbf{x} to the original coordinate \mathbf{z} requires a projection operator Π_z that projects the augmented state $\boldsymbol{\rho}$ to original states \mathbf{z} , i.e.,

$$\mathbf{x} \xrightarrow{\tau^{-1}} \boldsymbol{\rho} \xrightarrow{\Pi_z} \mathbf{z}, \quad \text{while} \quad \mathbf{u} \xrightarrow{\tau_u^{-1}} \mathbf{v}. \quad (1.9)$$

Static feedback linearizable systems are a special case of the above where the original dynamics is full state feedback linearizable without the need for a compensator. In other words, there exists a diffeomorphism $\tau(\mathbf{z}) : \mathbb{R}^{d_z} \mapsto \mathbb{R}^d$, $d = d_z$, that takes (1.5) to integrator dynamics (1.6), wherein the input mapping $\tau_u : (\mathbf{z}, \mathbf{v}) \mapsto \mathbf{u}$ follows

$$\mathbf{u}(t) = \mathbf{C}(\mathbf{z}(t)) \mathbf{v}(t) + \mathbf{d}(\mathbf{z}(t)). \quad (1.10)$$

Later we will show how the feedback linearization property can be used in estimating the support function of reach set resulting from (1.5). In this thesis, we have only considered the class of dynamic state feedback linearizable systems for which the compensator dynamics is affine in control.

Next we will provide some examples to illustrate the ideas outlined so far.

Example: static state feedback linearizable system with $\mathbf{r} = (3, 2)^\top$

Consider the differentially flat system [42, p. 257]

$$\begin{aligned} \dot{z}_1 &= z_2 + z_2^2 + v_1, & \dot{z}_2 &= z_3 - z_1 z_4 + z_4 z_5 \\ \dot{z}_3 &= z_2 z_4 + z_1 z_5 - z_5^2 + \cos(z_1 - z_5)v_1 + v_2, & \dot{z}_4 &= z_5, & \dot{z}_5 &= z_2^2 + v_2, \end{aligned} \quad (1.11)$$

where $\mathbf{z} \in \mathbb{R}^5$ and $\mathbf{v} \in \mathcal{V} \subset \mathbb{R}^2$. System (1.11) is full state static feedback linearizable and the flat outputs are given by

$$y_1 = z_1 - z_5, \quad y_2 = z_4. \quad (1.12)$$

Setting $(x_1, x_4) = (y_1, y_2)$ and differentiating over time t , we find

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (1.13)$$

The state mapping $\boldsymbol{\tau} : \mathbf{z} \rightarrow \mathbf{x}$, its inverse, $\boldsymbol{\tau}^{-1} : \mathbf{x} \rightarrow \mathbf{z}$, and the input mapping, $\boldsymbol{\tau}_u : \mathbf{v} \rightarrow \mathbf{u}$ are as follows

$$\mathbf{x} = \boldsymbol{\tau}(\mathbf{z}) = \begin{bmatrix} z_1 - z_5 \\ z_2 \\ z_3 - z_1 z_4 + z_4 z_5 \\ z_4 \\ z_5 \end{bmatrix}, \quad \mathbf{z} = \boldsymbol{\tau}^{-1}(\mathbf{x}) = \begin{bmatrix} x_1 + x_5 \\ x_2 \\ x_3 + (x_1 + x_5)x_4 - x_4 x_5 \\ x_4 \\ x_5 \end{bmatrix}, \quad (1.14a)$$

$$\mathbf{u} = \boldsymbol{\tau}_u(\mathbf{z}, \mathbf{v}) = \begin{bmatrix} \cos(z_1 - z_5)v_1 + v_2 \\ z_2^2 + v_2 \end{bmatrix}. \quad (1.14b)$$

Example: static state feedback linearizable system with $r = d = 4$

Consider the differentially flat system [43, Example 13.14]

$$\dot{z}_1 = z_2, \quad \dot{z}_2 = -\sin(z_1) - (z_1 - z_3), \quad \dot{z}_3 = z_4, \quad \dot{z}_4 = (z_1 - z_3) + v \quad (1.15)$$

where $\mathbf{z} \in \mathbb{R}^4$ and $v \in \mathcal{V} \subset \mathbb{R}$. This is a model of single link manipulator with flexible joints and negligible damping. System (1.15) is full state static feedback linearizable with the flat output $y = z_1$. Setting $x_1 = y_1$ and differentiating over time t , we find

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u. \quad (1.16)$$

The state mapping $\boldsymbol{\tau} : \mathbf{z} \rightarrow \mathbf{x}$, its inverse, $\boldsymbol{\tau}^{-1} : \mathbf{x} \rightarrow \mathbf{z}$, and the input mapping $\boldsymbol{\tau}_u : \mathbf{v} \rightarrow \mathbf{u}$ are given by

$$\mathbf{x} = \boldsymbol{\tau}(\mathbf{z}) = \begin{bmatrix} z_1 \\ z_2 \\ -\sin(z_1) - (z_1 - z_3) \\ -z_2 \cos(z_1) - (z_2 - z_4) \end{bmatrix}, \quad \mathbf{z} = \boldsymbol{\tau}^{-1}(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 + \sin(x_1) + x_1 \\ x_4 + x_2 \cos(x_1) + x_2 \end{bmatrix}, \quad (1.17a)$$

$$\mathbf{u} = \boldsymbol{\tau}_u(\mathbf{z}, \mathbf{v}) = -(\cos(z_1) + 2)(-\sin(z_1) + z_3 - z_1) + (z_2^2 - 1)\sin(z_1) + v. \quad (1.17b)$$

Example: dynamic state feedback linearizable system

Consider the differentially flat system

$$\dot{z}_1 = z_2 - v_1, \quad \dot{z}_2 = z_4 v_1, \quad \dot{z}_3 = v_1, \quad \dot{z}_4 = v_2 \quad (1.18)$$

where $\mathbf{z} \in \mathbb{R}^4$ and $\mathbf{v} \in \mathcal{V} \subset \mathbb{R}^2$. System (1.18) is dynamic feedback linearizable. It can be shown that this system is not static state feedback linearizable. We find the compensator variable $w = z_2 - v_1$, and the flat outputs

$$y_1 = z_1, \quad y_2 = z_1 + z_3. \quad (1.19)$$

Now having $(x_1, x_3) = (y_1, y_2)$ and differentiating over time t , we recover the integrator dynamics $\mathbf{x} \in \mathbb{R}^5$,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (1.20)$$

The state mapping $\tau : \mathbf{z} \rightarrow \mathbf{x}$, its inverse, $\tau^{-1} : \mathbf{x} \rightarrow \mathbf{z}$, and the input mapping, $\tau_u : \mathbf{v} \rightarrow \mathbf{u}$ are as follows

$$\mathbf{x} = \tau(\mathbf{z}, w) = \begin{bmatrix} z_1 \\ w \\ z_1 + z_3 \\ z_2 \\ z_4(z_2 - w) \end{bmatrix}, \quad \mathbf{z} = \tau^{-1}(\mathbf{x}, w) = \begin{bmatrix} x_1 \\ x_4 \\ x_3 - x_1 \\ x_5 / (x_4 - w) \\ w \end{bmatrix}, \quad (1.21a)$$

$$\mathbf{u} = \tau_u(\mathbf{z}, \mathbf{w}, \mathbf{v}) = \begin{bmatrix} \dot{w} \\ v_2(z_2 - w) + z_4(z_4 v_1 - \dot{w}) \end{bmatrix}. \quad (1.21b)$$

1.6.2 State transition matrix

For $0 \leq s < t$, the state transition matrix $\Phi(t, s)$ associated with integrator dynamics (1.6) is

$$\Phi(t, s) \equiv \exp(\mathbf{A}(t-s)) \text{blkdiag}(\exp(\mathbf{A}_1(t-s)), \dots, \exp(\mathbf{A}_m(t-s))),$$

with each diagonal block is upper triangular. Specifically, the j th diagonal block of size $r_j \times r_j$ is written element-wise as

$$\exp(\mathbf{A}_j(t-s)) := \begin{cases} \frac{(t-s)^{\ell-k}}{(\ell-k)!} & \text{for } k \leq \ell, \\ 0 & \text{otherwise,} \end{cases} \quad (1.22)$$

where k is the row index, ℓ is the column index, and $k, \ell \in [r_j]$ for each $j \in [m]$. The diagonal entries in (1.22) are unity.

1.6.3 Support function

The support function $h_{\mathcal{X}}(\cdot)$ of a nonempty compact¹ set $\mathcal{X} \subset \mathbb{R}^d$, is given by

$$h_{\mathcal{X}}(\mathbf{y}) := \sup_{\mathbf{x} \in \mathcal{X}} \{\langle \mathbf{y}, \mathbf{x} \rangle \mid \mathbf{y} \in \mathbb{S}^{d-1}\}, \quad (1.23)$$

where $\langle \cdot, \cdot \rangle$ denotes the standard Euclidean inner product, and \mathbb{S}^{d-1} is the unit sphere imbedded in \mathbb{R}^d . Using $\overline{\cdot}$ and conv to respectively denote set closure and convex hull, it is immediate that

$$h_{\mathcal{X}}(\mathbf{y}) = h_{\overline{\text{conv}(\mathcal{X})}}(\mathbf{y}) \quad \forall \mathbf{y} \in \mathbb{S}^{d-1}, \quad (1.24)$$

so the support function of a compact nonconvex set \mathcal{X} is understood as that of the closure of its convex hull. Definition (1.23) being pointwise supremum of linear func-

¹We can extend the definition of support function for any nonempty $\mathcal{X} \subseteq \mathbb{R}^d$ by allowing its range to be $\mathbb{R} \cup \{+\infty\}$. When \mathcal{X} is bounded, then $h_{\mathcal{X}}(\cdot)$ is finite everywhere; otherwise $h_{\mathcal{X}}(\cdot)$ may take value $+\infty$ but remains lower semi-continuous.

tions, it follows that $h_{\mathcal{X}}(\mathbf{y})$ is a convex function in \mathbf{y} .

Geometrically, $h_{\mathcal{X}}(\mathbf{y})$ quantifies the signed distance of the supporting hyperplane of \mathcal{X} with outer normal vector \mathbf{y} , measured from the origin. This distance is negative if and only if the unit vector $\mathbf{y} \in \mathbb{S}^{d-1}$ points into the open halfspace containing the origin.

The support function $h_{\mathcal{X}}(\cdot)$ uniquely determines a compact set \mathcal{X} (up to closure of convexification). This is because the Legendre-Fenchel conjugate of $h_{\mathcal{X}}(\cdot)$ is the indicator function [44, Thm. 13.2]

$$\mathbf{1}_{\mathcal{X}}(\mathbf{x}) := \begin{cases} 0 & \text{if } \mathbf{x} \in \mathcal{X}, \\ +\infty & \text{otherwise.} \end{cases} \quad (1.25)$$

This allows thinking and using $h_{\mathcal{X}}(\cdot)$ as a functional proxy for the set \mathcal{X} .

As a function, the support function has additional structure beyond convexity. A support function is closed² and sublinear [45, p. 123], i.e., a convex and positive homogeneous function of degree 1. So for any compact $\mathcal{X} \subset \mathbb{R}^d$,

$$h_{\mathcal{X}}(a\mathbf{y}) = ah_{\mathcal{X}}(\mathbf{y}), \quad a \in \mathbb{R}_{>0}, \quad \forall \mathbf{y} \in \mathbb{S}^{d-1}.$$

It is well-known [45, Prop. 1.1.3] that a function is sublinear if and only if its epigraph is a nonempty convex cone. Hence the epigraph of any support function is a nonempty closed convex cone.

Conversely, any closed sublinear function $h : \mathbb{S}^{d-1} \mapsto \mathbb{R}$, is the support function of a compact convex set $\mathcal{X}_h := \{\mathbf{x} \in \mathbb{R}^d \mid \langle \mathbf{y}, \mathbf{x} \rangle \leq h(\mathbf{y}) \ \forall \mathbf{y} \in \mathbb{S}^{d-1}\}$; see e.g., [45, Thm. 3.1.1]. Thus, the space of closed sublinear functions has an isomorphism with the space of finite dimensional compact convex sets.

²A function is closed if its epigraph is a closed set.

Support function calculus

In this dissertation we will extensively employed the support function calculus. Support function representation behaves well under common set-valued operations. We next list a collection of known [46, Ch.V], [47, Ch. 1.7] results to highlight that indeed there exist *equivalence* between set-valued operations and functional operations on the corresponding support functions. In the following, all sets $\mathcal{X}, \mathcal{X}_i$ are assumed to be compact convex w.l.o.g.

1. non-membership³ \Leftrightarrow inequality

$$\tilde{\mathbf{x}} \notin \mathcal{X} \Leftrightarrow \exists \mathbf{y} \in \mathbb{S}^{d-1} \text{ such that } \langle \mathbf{y}, \tilde{\mathbf{x}} \rangle > h_{\mathcal{X}}(\mathbf{y}).$$

2. convergence in Hausdorff topology \Leftrightarrow pointwise convergence

$$\{\mathcal{X}_i\} \rightarrow \mathcal{X} \Leftrightarrow h_{\mathcal{X}_i}(\cdot) \rightarrow h_{\mathcal{X}}(\cdot).$$

3. inclusion \Leftrightarrow inequality

$$\mathcal{X}_1 \subseteq \mathcal{X}_2 \Leftrightarrow h_{\mathcal{X}_1}(\mathbf{y}) \leq h_{\mathcal{X}_2}(\mathbf{y}) \quad \forall \mathbf{y} \in \mathbb{S}^{d-1}.$$

4. intersection \Leftrightarrow infimal convolution

$$\mathcal{X} = \bigcap_{i=1}^r \mathcal{X}_i \Leftrightarrow h_{\mathcal{X}}(\mathbf{y}) = \inf_{\mathbf{y}_1 + \dots + \mathbf{y}_r = \mathbf{y}} \{h_{\mathcal{X}_1}(\mathbf{y}_1) + \dots + h_{\mathcal{X}_r}(\mathbf{y}_r)\}.$$

5. affine transformation \Leftrightarrow composition

$$\mathcal{X} = \Gamma \mathcal{X}_0 + \boldsymbol{\gamma}, \Gamma \in \mathbb{R}^{d \times d'}, \boldsymbol{\gamma} \in \mathbb{R}^d \Leftrightarrow h_{\mathcal{X}}(\mathbf{y}) = \langle \mathbf{y}, \boldsymbol{\gamma} \rangle + h_{\mathcal{X}_0}(\Gamma^\top \mathbf{y}) \quad \forall \mathbf{y} \in \mathbb{S}^{d-1}.$$

6. p -sum $\Leftrightarrow p$ -norm

$$\mathcal{X} = \mathcal{X}_1 +_p \dots +_p \mathcal{X}_r \Leftrightarrow h_{\mathcal{X}}(\mathbf{y}) = \left(h_{\mathcal{X}_1}^p(\mathbf{y}) + \dots + h_{\mathcal{X}_r}^p(\mathbf{y}) \right)^{1/p} \quad \forall \mathbf{y} \in \mathbb{S}^{d-1}.$$

6.a. ($p = 1$) Minkowski sum \Leftrightarrow sum

$$\mathcal{X} = \mathcal{X}_1 + \dots + \mathcal{X}_r \Leftrightarrow h_{\mathcal{X}}(\mathbf{y}) = h_{\mathcal{X}_1}(\mathbf{y}) + \dots + h_{\mathcal{X}_r}(\mathbf{y}) \quad \forall \mathbf{y} \in \mathbb{S}^{d-1}.$$

³In fact, stronger equivalence [45, Thm. 2.2.3] holds: we can verify (non-)membership to affine hull, interior, and relative interior of a compact convex set. So the support function “filters” the affine hull, the interior, and the relative interior of the set.

6.b. ($p = \infty$) union \Leftrightarrow pointwise maximum

$$\mathcal{X} = \bigcup_{i=1}^r \mathcal{X}_i \Leftrightarrow h_{\mathcal{X}}(\mathbf{y}) = \max\{h_{\mathcal{X}_1}(\mathbf{y}), \dots, h_{\mathcal{X}_r}(\mathbf{y})\} \quad \forall \mathbf{y} \in \mathbb{S}^{d-1}.$$

Furthermore, many geometric functionals of interest can be expressed in terms of the support function. For instance, the *width* [47, p. 42] of $\mathcal{X} \subset \mathbb{R}^d$ in the direction $\mathbf{y} \in \mathbb{S}^{d-1}$ equals $h_{\mathcal{X}}(\mathbf{y}) + h_{\mathcal{X}}(-\mathbf{y})$, and the *diameter* is the maximal width:

$$\text{diam}(\mathcal{X}) := \max_{\mathbf{y} \in \mathbb{S}^{d-1}} (h_{\mathcal{X}}(\mathbf{y}) + h_{\mathcal{X}}(-\mathbf{y})).$$

The (two-sided) *Hausdorff distance metric* δ_{H} between compact convex $\mathcal{X}_1, \mathcal{X}_2 \subset \mathbb{R}^d$, can be computed using the corresponding support functions as (Chapter 5)

$$\delta_{\text{H}}(\mathcal{X}_1, \mathcal{X}_2) = \sup_{\mathbf{y} \in \mathbb{S}^{d-1}} |h_{\mathcal{X}_1}(\mathbf{y}) - h_{\mathcal{X}_2}(\mathbf{y})|.$$

The following Lemmas will be useful (proofs are given in Appendix A.1 and A.2 respectively).

Lemma 1.1. *Let $\{\mathcal{K}_i\}_{i \in \mathbb{N}}$ be a sequence of compact convex sets in \mathbb{R}^d . Then, $\mathcal{K}_i \rightarrow \mathcal{K} \Leftrightarrow h_{\mathcal{K}_i}(\cdot) \rightarrow h_{\mathcal{K}}(\cdot)$.*

Lemma 1.2. *Let $\{\mathcal{K}_i\}_{i \in \mathbb{N}}$ be a sequence of compact convex sets in \mathbb{R}^d . Let $\text{vol}(\cdot)$ denote the d -dimensional volume. If $\mathcal{K}_i \rightarrow \mathcal{K}$, then $\text{vol}(\mathcal{K}_i) \rightarrow \text{vol}(\mathcal{K})$ as $i \rightarrow \infty$.*

Support functions also appear in set duality as presented next.

1.6.4 Polar dual

The *polar dual* \mathcal{K}° of any non-empty set $\mathcal{K} \subset \mathbb{R}^d$ is given by

$$\mathcal{K}^\circ := \{\mathbf{y} \in \mathbb{R}^d \mid \langle \mathbf{y}, \mathbf{x} \rangle \leq 1 \quad \text{for all } \mathbf{x} \in \mathcal{K}\}. \quad (1.26)$$

From this definition, it is immediate that \mathcal{K}° contains the origin, and is a closed convex set. The bipolar $(\mathcal{K}^\circ)^\circ = \text{closure}(\text{conv}(\mathcal{K} \cup \{\mathbf{0}\}))$. Thus, if \mathcal{K} is compact convex and contains the origin, then we have the involution $(\mathcal{K}^\circ)^\circ = \mathcal{K}$. From (1.23) and (1.26),

notice that \mathcal{K}° is the *unit support function ball*, i.e., $\mathcal{K}^\circ = \{\mathbf{y} \in \mathbb{R}^d \mid h_{\mathcal{K}}(\mathbf{y}) \leq 1\}$. In Ch. 2.1.4, we will mention some properties of the polar dual of the integrator reach set.

1.6.5 Vector measure

Let \mathcal{F} be a σ -field of the subsets of a set. A countably additive mapping $\tilde{\mu} : \mathcal{F} \mapsto \mathbb{R}^d$ is termed a *vector measure*. Here, “countably additive” means that for any sequence $\{\Omega_i\}_{i=1}^\infty$ of disjoint sets in \mathcal{F} such that their union is in \mathcal{F} , we have $\tilde{\mu}(\cup_{i=1}^\infty \Omega_i) = \sum_{i=1}^\infty \tilde{\mu}(\Omega_i) < \infty$. Some of the early investigations of vector measures were due to Liapounoff [48] and Halmos [49]; relatively recent references are [50, 51].

1.6.6 Zonotope

A *zonotope* $\mathcal{Z} \subset \mathbb{R}^d$ is a finite Minkowski sum of closed line segments or intervals $\{I_i\}_{i=1}^n$ where these intervals are imbedded in the ambient Euclidean space \mathbb{R}^d . Explicitly, for some positive integer n , we write

$$\mathcal{Z} := I_1 \dot{+} \dots \dot{+} I_n \left\{ \mathbf{x} \in \mathbb{R}^d \mid \mathbf{x} = \sum_{i=1}^n \mathbf{x}_i, \quad \mathbf{x}_i \in I_i, \quad i \in [n] \right\}.$$

Thus, a zonotope is the range of an *atomic* vector measure. Alternatively, a zonotope can be viewed as the affine image of the unit cube. A compact convex polytope is a zonotope if and only if all its two dimensional faces are centrally symmetric [47, p. 182]. For instance, the cross polytope $\{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_1 \leq 1\}$, is not a zonotope. Standard references on zonotope include [52, 53], [54, Ch. 2.7].

The set of zonotopes is closed under affine image and Minkowski sum, but not under intersection. In the systems-control literature, a significant body of work exists on computationally efficient over-approximation of reach sets via zonotopes [55–57] and its variants such as zonotope bundles [58], constrained zonotopes [59], complex

zonotopes [60], and polynomial zonotopes [61, 62].

1.6.7 Variety and ideal

Let $p_1, \dots, p_n \in \mathbb{R}[x_1, \dots, x_d]$, the vector space of real-valued d -variate polynomials. The (affine) *variety* $V_{\mathbb{R}[x_1, \dots, x_d]}(p_1, \dots, p_n)$ is the set of all solutions of the system

$$p_1(x_1, x_2, \dots, x_d) = \dots = p_n(x_1, x_2, \dots, x_d) = 0.$$

Given $p_1, \dots, p_n \in \mathbb{R}[x_1, \dots, x_d]$, the set

$$I := \left\{ \sum_{i=1}^n \alpha_i p_i \mid \alpha_1, \dots, \alpha_n \in \mathbb{R}[x_1, \dots, x_d] \right\}$$

is called the *ideal* generated by p_1, \dots, p_n . We write this symbolically as $I = \langle\langle p_1, \dots, p_n \rangle\rangle$.

Roughly speaking, $\langle\langle p_1, \dots, p_n \rangle\rangle$ is the set of all polynomial consequences of the given system of n polynomial equations in d indeterminates. We refer the readers to [63, Ch. 1] for detailed exposition of these concepts.

1.6.8 Minkowski difference

Given two compact convex sets $\mathcal{P}, \mathcal{Q} \subset \mathbb{R}^n$, the Minkowski difference

$$\mathcal{P} \dot{-} \mathcal{Q} := \{\mathbf{p} - \mathbf{q} \mid \mathbf{p} \in \mathcal{P}, \mathbf{q} \in \mathcal{Q}\},$$

and is compact convex. Checking intersection between the sets \mathcal{P}, \mathcal{Q} is then equivalent to verifying if the zero vector $\mathbf{0} \in \mathbb{R}^d$ belongs to $\mathcal{P} \dot{-} \mathcal{Q}$. This can in turn be related to conditions on the support function $h_{\mathcal{P} \dot{-} \mathcal{Q}}(\cdot)$. In particular,

$$\mathcal{P} \cap \mathcal{Q} \neq \emptyset \Leftrightarrow \mathbf{0} \in \mathcal{P} \dot{-} \mathcal{Q} \Leftrightarrow \forall \mathbf{y} \in \mathbb{S}^{d-1}, h_{\mathcal{P} \dot{-} \mathcal{Q}}(\mathbf{y}) \geq 0, \quad (1.27a)$$

$$\mathcal{P} \cap \mathcal{Q} = \emptyset \Leftrightarrow \mathbf{0} \notin \mathcal{P} \dot{-} \mathcal{Q} \Leftrightarrow \exists \mathbf{y} \in \mathbb{S}^{d-1} \text{ such that } h_{\mathcal{P} \dot{-} \mathcal{Q}}(\mathbf{y}) < 0. \quad (1.27b)$$

We refer the readers to [64,65] where these ideas were explored for checking set intersections. We will make use of these ideas in Ch. 4.2 to derive and solve an optimization problem for certifying the intersections of integrator reach sets.

1.6.9 Induced norm

For $1 \leq p, q \leq \infty$, matrix $M \in \mathbb{R}^{m \times n}$ viewed as a linear map $M : \ell_p(\mathbb{R}^n) \mapsto \ell_q(\mathbb{R}^m)$, has an associated induced operator norm

$$\|M\|_{p \rightarrow q} := \sup_{\mathbf{x} \neq \mathbf{0}} \frac{\|M\mathbf{x}\|_q}{\|\mathbf{x}\|_p} = \sup_{\|\mathbf{x}\|_p=1} \|M\mathbf{x}\|_q, \quad (1.28)$$

where as usual $\|\mathbf{x}\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$, $\|M\mathbf{x}\|_q := (\sum_{i=1}^m |(M\mathbf{x})_i|^q)^{1/q}$ for p, q finite, $\|\cdot\|_\infty$ is the sup norm, and $(M\mathbf{x})_i$ denotes the i th component of the vector $M\mathbf{x}$. Several special cases of (1.28) are well known: the case $p = q$ is the standard matrix p norm, the case $p = \infty, q = 1$ is the Grothendieck problem [66, 67] that features prominently in combinatorial optimization, and its generalization $p \in (1, \infty), q = 1$ is the ℓ_p Grothendieck problem [68]. In our development, the operator norm $\|M\|_{2 \rightarrow q}$ arises where $1 < q \leq \infty$.

1.7 Integrator Reach Set with Compact Input Set \mathcal{U}

In this section we investigate the integrator *forward reach set* for the general compact input set \mathcal{U} . Let $\mathcal{X}_t(\mathcal{X}_0)$ denote the forward reach set of (1.6) at time $t > 0$, starting from a given compact convex set of initial conditions $\mathcal{X}_0 \subset \mathbb{R}^d$, i.e.,

$$\begin{aligned} \mathcal{X}_t(\mathcal{X}_0) := & \bigcup_{\text{measurable } \mathbf{u} \in \text{closure}(\text{conv}(\mathcal{U}))} \{ \mathbf{x}(t) \in \mathbb{R}^d \mid (1.6) \text{ and } (1.7) \text{ hold,} \\ & \mathbf{x}(t=0) \in \mathcal{X}_0 \text{ compact convex, } \mathcal{U}(s) \subset \mathbb{R}^m \text{ compact for all } s \in [0, t] \}. \end{aligned} \quad (1.29)$$

In words, $\mathcal{X}_t(\mathcal{X}_0)$ is the set of all states that the controlled dynamics (1.6)-(1.7) can reach at time $t > 0$, starting from the set \mathcal{X}_0 at $t = 0$, with measurable control $\mathbf{u}(\cdot) \in \mathcal{U}$ compact. Notice that definition 1.29 inform us that the reach set \mathcal{X}_t resulting from compact $\mathcal{U}(s)$ is the same as that resulting from the closure of the convex hull of $\mathcal{U}(s)$, $0 \leq s \leq t$. Formally, we can write

$$\mathcal{X}_t(\mathcal{X}_0) = \exp(t\mathbf{A})\mathcal{X}_0 + \int_0^t \exp((t-\tau)\mathbf{A})\mathbf{B}\mathcal{U}(s) \, d\tau \quad (1.30)$$

It is straightforward to prove that $\mathcal{X}_t(\mathcal{X}_0, t)$ is a compact convex subset of \mathbb{R}^d for all $t > 0$. Notice however, that the (space-time) *forward reachable tube*

$$\overline{\mathcal{X}}_t(\mathcal{X}_0) := \bigcup_{0 \leq \tau \leq t} \mathcal{X}_\tau(\mathcal{X}_0), \quad (1.31)$$

where the union symbol denotes a disjoint union, need not be convex in $\mathbb{R}^d \times \mathbb{R}_{>0}$. The following Lemmas will be useful for our further developments (proofs in Appendix A.3).

Lemma 1.3. *Let $F(\cdot)$ be a point-to-set function that is non-empty on $[0, t]$. For $0 \leq s \leq t$, denote its support function as $h_{F(s)}(\mathbf{y}) \equiv h(s, \mathbf{y})$ for any $\mathbf{y} \in \mathbb{R}^d$. Then,*

$$h_{\int_0^t F(s) \, ds}(\mathbf{y}) = \int_0^t h(s, \mathbf{y}) \, ds.$$

Remark 1.1. *A special case of the above result for continuous-time LTI systems was derived in [69, Proposition 2]. Compared to the same, both the statement and proof of Lemma 1.3 are general (valid for any point-to-set function).*

Following (1.23) and Lemma 1.3, we deduce Proposition 1.1 stated next (proof in Appendix A.4).

Proposition 1.1. (Support function for compact \mathcal{U}) *For compact convex $\mathcal{X}_0 \subset \mathbb{R}^d$,*

and compact $\mathcal{U} \subset \mathbb{R}^m$, the support function of the reach set (1.30) is

$$h_{\mathcal{X}_t(\mathcal{X}_0)}(\mathbf{y}) = \sup_{\mathbf{x}_0 \in \mathcal{X}_0} \sum_{j=1}^m \langle \mathbf{y}_j, \exp(t\mathbf{A}_j) \mathbf{x}_{j0} \rangle + \int_0^t \sup_{\mathbf{u} \in \text{closure}(\text{conv}(\mathcal{U}))} \sum_{j=1}^m \{ \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle u_j \} ds, \quad (1.32)$$

where $\text{conv}(\cdot)$ denotes the convex hull and

$$\boldsymbol{\xi}_j(s) := \left(s^{r_j-1}/(r_j-1)! \quad s^{r_j-2}/(r_j-2)! \quad \cdots \quad s \quad 1 \right)^\top, \quad (1.33)$$

In the following section, we will present a systematic study on the geometry of integrator reach set.

2 | Integrator Reach Sets with Time Invariant Set-Valued Uncertainties

Integrators with bounded controls are ubiquitous in systems-control. Other than Brunovsky normal forms for the feedback linearizable nonlinear systems which in turn appear frequently in vehicular dynamics, they are also commonly used as benchmark problems to demonstrate the performance of the reach set computation algorithms. Despite their prominence, specific results on the geometry of the integrator reach sets are not available in the systems-control literature. Broadly speaking, the existing results come in two flavors. On one hand, very generic statements are known, e.g., these reach sets are compact convex sets whenever the set of initial conditions is compact convex, and the controls take values from a compact (not necessarily convex) set [3]. On the other hand, several numerical toolboxes [35, 70] are available for tight outer approximation of the reach sets over computationally benign geometric families such as ellipsoids and zonotopes. The lack of concrete geometric results imply the absence of ground truth when comparing the efficacy of different algorithms, and one has to content with graphical or statistical (e.g., Monte Carlo) comparisons. In this chapter we systematically study the geometry of integrator reach sets.

In this section, we study the multi-input integrator reach set, when input set is *time invariant box-valued*, i.e., \mathcal{U} is a hyperrectangle given by

$$\mathcal{U} := [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \cdots \times [\alpha_m, \beta_m] \subset \mathbb{R}^m. \quad (2.1)$$

To assist our memory, we denote this specific integrator reach set via with a box superscript $\mathcal{X}_t^\square(\mathcal{X}_0)$ ¹ and we can rewrite (1.29) as

$$\mathcal{X}_t^\square(\mathcal{X}_0) = \exp(t\mathbf{A})\mathcal{X}_0 \dot{+} \int_0^t \exp(s\mathbf{A})\mathbf{B}\mathcal{U} \, ds. \quad (2.2)$$

¹For the single input ($m = 1$) case, we drop the box superscript.

For $\mathcal{X}_0 \subset \mathbb{R}^d$ compact convex, it is well-known [3, Sec. 2] that $\mathcal{X}_t^\square(\mathcal{X}_0)$ is a compact convex set for all $t > 0$. However, it is not immediate what kind of convex set \mathcal{X}_t^\square is, even for singleton $\mathcal{X}_0 \equiv \{\mathbf{x}_0\}$. To proceed further, we introduce some notations. From (2.1), α_j and β_j are defined as the component-wise minimum and maximum of the input set, respectively, i.e.,

$$\alpha_j := \min_{\mathbf{u} \in \mathcal{U}} u_j, \quad \beta_j := \max_{\mathbf{u} \in \mathcal{U}} u_j, \quad j \in \llbracket m \rrbracket, \quad (2.3)$$

Furthermore, let

$$\mu_j := \frac{\beta_j - \alpha_j}{2}, \quad \nu_j := \frac{\beta_j + \alpha_j}{2}, \quad (2.4)$$

and as in Prop. 1.1, introduce

$$\boldsymbol{\xi}(s) := \begin{pmatrix} \mu_1 \boldsymbol{\xi}_1(s) \\ \vdots \\ \mu_m \boldsymbol{\xi}_m(s) \end{pmatrix}, \quad \boldsymbol{\xi}_j(s) := \begin{pmatrix} s^{r_j-1}/(r_j-1)! \\ s^{r_j-2}/(r_j-2)! \\ \vdots \\ s \\ 1 \end{pmatrix}, \quad (2.5)$$

for $j \in \llbracket m \rrbracket$. Also, let

$$\boldsymbol{\zeta}(t_0, t) = \begin{pmatrix} \mu_1 \boldsymbol{\zeta}_1(t_0, t) \\ \mu_2 \boldsymbol{\zeta}_2(t_0, t) \\ \vdots \\ \mu_m \boldsymbol{\zeta}_m(t_0, t) \end{pmatrix}, \quad \boldsymbol{\zeta}_j(t_0, t) := \int_{t_0}^t \boldsymbol{\xi}_j(t-s) \, ds \in \mathbb{R}^{r_j}, \quad (2.6)$$

for $j \in \llbracket m \rrbracket$. When $t_0 = 0$, we simplify the notations as

$$\boldsymbol{\zeta}(t) := \boldsymbol{\zeta}(0, t), \quad \boldsymbol{\zeta}_j(t) := \boldsymbol{\zeta}_j(0, t) \text{ for all } j \in \llbracket m \rrbracket. \quad (2.7)$$

From (1.7a) we know that the system matrices are block diagonal, so when having the input set as (2.1), each of the m single input integrator dynamics with r_j

dimensional state subvectors for $j \in \llbracket m \rrbracket$, are decoupled from each other. Hence $\mathcal{X}_t^\square(\mathcal{X}_0) \subset \mathbb{R}^d$ is the Cartesian product of these single input integrator reach sets: $\mathcal{X}_{jt}(\mathcal{X}_0) \subset \mathbb{R}^{r_j}$ for $j \in \llbracket m \rrbracket$. However, the ‘‘factor sets’’ in this Cartesian product belong to disjoint mutually orthogonal r_j dimensional subspaces, $j \in \llbracket m \rrbracket$, which allows writing this Cartesian product as a Minkowski sum², i.e.,

$$\mathcal{X}_t^\square = \mathcal{X}_{1t} \times \mathcal{X}_{2t} \times \cdots \times \mathcal{X}_{mt} \equiv \mathcal{X}_{1t} \dot{+} \mathcal{X}_{2t} \dot{+} \cdots \dot{+} \mathcal{X}_{mt} \quad (2.8)$$

Notice that the decoupled dynamics also allows us to write a Minkowski sum decomposition for the set of initial conditions $\mathcal{X}_0 = \mathcal{X}_{10} \dot{+} \cdots \dot{+} \mathcal{X}_{m0}$, and accordingly, the initial condition subvectors $\mathbf{x}_{j0} \in \mathcal{X}_{j0} \subset \mathbb{R}^{r_j}$ for $j \in \llbracket m \rrbracket$. Thus $\mathbf{x}_0 = (\mathbf{x}_{10}, \dots, \mathbf{x}_{m0})^\top$.

Since the support function of the Minkowski sum is equal to the sum of the support functions, we have

$$h_{\mathcal{X}_t^\square(\mathcal{X}_0)}(\mathbf{y}) = \sum_{j=1}^m h_{\mathcal{X}_{jt}(\mathcal{X}_{j0})}(\mathbf{y}_j). \quad (2.9)$$

This leads to the following result (proof in Appendix BB.1) which will come in handy in the ensuing development.

Theorem 2.1. *For compact convex $\mathcal{X}_0 \subset \mathbb{R}^d$, and box-valued input uncertainty set given by (2.1), the support function of the reach set (2.2) is*

$$h_{\mathcal{X}_t^\square(\mathcal{X}_0)}(\mathbf{y}) = \sum_{j=1}^m \left\{ \sup_{\mathbf{x}_{j0} \in \mathcal{X}_{j0}} \langle \mathbf{y}_j, \exp(t\mathbf{A}) \mathbf{x}_{j0} \rangle + \nu_j \langle \mathbf{y}_j, \boldsymbol{\zeta}_j(t) \rangle + \mu_j \int_0^t |\langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle| ds \right\}. \quad (2.10)$$

where $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ are defined in (2.4)-(2.6).

The formula (2.10) upper bounds (1.32) resulting from the same initial condition and arbitrary compact $\mathcal{U} \subset \mathbb{R}^m$ with $\{\alpha_j, \beta_j\}_{j=1}^m$ related to \mathcal{U} via (2.3). Thus, from Property 1.6.3 of support functions, the reach set \mathcal{X}_t^\square with box-valued input uncer-

²In general, the Minkowski sum of a given collection of compact convex sets is not equal to their Cartesian product.

tainty will over-approximate the reach set \mathcal{X}_t associated with arbitrary compact \mathcal{U} , at any given $t > 0$, provided $\{\alpha_j, \beta_j\}_{j=1}^m$ are defined as (2.3).

When \mathcal{U} is compact but not box-valued, then we can quantify the quality of the aforesaid over-approximation in terms of the two-sided Hausdorff distance metric dist between the convex compact sets $\mathcal{X}_t^\square, \mathcal{X}_t \subset \mathbb{R}^d$, expressible [47, Thm. 1.8.11] in terms of their support functions $h_{\mathcal{X}_t^\square}(\cdot), h_{\mathcal{X}_t}(\cdot)$ as

$$\text{dist}(\mathcal{X}_t^\square, \mathcal{X}_t) = \sup_{\|\mathbf{y}\|_2=1} |h_{\mathcal{X}_t^\square}(\mathbf{y}) - h_{\mathcal{X}_t}(\mathbf{y})|. \quad (2.11)$$

Thanks to Property 1.6.3 of support functions, the absolute value in (2.11) can be dispensed since $\mathcal{X}_t \subseteq \mathcal{X}_t^\square$ with set equality if \mathcal{U} is box, in which case $h_{\mathcal{X}_t^\square}(\cdot) = h_{\mathcal{X}_t}(\cdot)$ and $\text{dist} = 0$.

It is known that [71, Prop. 6.1] the set $\mathcal{X}_t(\mathcal{X}_0)$ resulting from a linear time invariant dynamics such as (1.6)-(1.7) remains invariant under the closure of convexification of the input set \mathcal{U} . Therefore, it is possible that $\mathcal{X}_t = \mathcal{X}_t^\square$ and $\text{dist} = 0$ even when the compact set \mathcal{U} is nonconvex. For instance, the reach set $\mathcal{X}_t(\mathcal{X}_0)$ resulting from some compact convex $\mathcal{X}_0 \subset \mathbb{R}^d$ and dynamics (1.6)-(1.7) with the nonconvex input uncertainty set $\{-1, 1\}^m$, is identical to $\mathcal{X}_t^\square(\mathcal{X}_0)$ resulting from the same \mathcal{X}_0 , same dynamics, and the box-valued input uncertainty set (2.1) with $\alpha_j = -1, \beta_j = 1$ for all $j \in \llbracket m \rrbracket$.

Likewise, for the same compact convex $\mathcal{X}_0 \subset \mathbb{R}^d$, the reach set $\mathcal{X}_t(\mathcal{X}_0)$ resulting from (1.6)-(1.7) with the nonconvex input set $\{\mathbf{u} \in \mathbb{R}^m \mid \|\mathbf{u}\|_p \leq 1\}$, $0 < p < 1$, is the same as that resulting from the cross-polytope $\{\mathbf{u} \in \mathbb{R}^m \mid \|\mathbf{u}\|_1 \leq 1\}$. More generally, for $0 < p < \infty$, suppose $\mathcal{X}_t^{\|\cdot\|_p}(\mathcal{X}_0)$ results from the unit p norm ball input uncertainty set $\{\mathbf{u} \in \mathbb{R}^m \mid \|\mathbf{u}\|_p \leq 1\}$. Let $\mathbf{M}^\top(\tau) := \exp(\tau \mathbf{A}) \mathbf{B} = \text{blkdiag}(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_m)$. If $\mathcal{X}_t^\square(\mathcal{X}_0)$ results from the same \mathcal{X}_0 , same dynamics, and input uncertainty set (2.1) with $\alpha_j = -1, \beta_j = 1$ for all $j \in \llbracket m \rrbracket$, then using [72, Thm. 1] and [73] (see Ch. 4.2 and 5.3) (2.11)

simplifies to

$$\text{dist}\left(\mathcal{X}_t^\square, \mathcal{X}_t^{\|\cdot\|^p}\right) = \sup_{\|\mathbf{y}\|_2=1} \int_0^t (\|\mathbf{M}(\tau)\mathbf{y}\|_1 - \|\mathbf{M}(\tau)\mathbf{y}\|_q) d\tau \quad (2.12)$$

where q is the Hölder conjugate of $\max\{1, p\}$, i.e., $\frac{1}{\max\{1, p\}} + \frac{1}{q} = 1$, and $1 < q \leq \infty$. In this case, the positive value (2.12) quantifies the quality of strict over-approximation $\mathcal{X}_t^{\|\cdot\|^p} \subset \mathcal{X}_t^\square$ for $0 < p < \infty$. The objective in (2.12) being positive homogeneous, admits lossless constraint convexification $\|\mathbf{y}\|_2 \leq 1$, and the corresponding maximal value³ for moderate dimensions d , can be found by direct numerical search. In chapter. 5, we will explore in detail the Hausdorff distance between norm balls and their linear maps as well as the Hausdorff distance between the LTI reach sets under different ℓ_p -norm valued input uncertainties.

In the rest of this chapter, we address the unexplored directions: the exact parametric and implicit equations for the boundary of $\mathcal{X}_t^\square(\mathcal{X}_0)$, the scaling laws for the volume and diameter of this sets and the classification of these sets.

2.1 Taxonomy and Boundary

In this Section, we examine the question “what type of compact convex set $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ is” from several points of view. In doing so, we also derive the equations for the boundary $\partial\mathcal{X}_t^\square(\{\mathbf{x}_0\})$. Notice that for non-singleton \mathcal{X}_0 , the taxonomy question is not well-posed since the classification then will depend on \mathcal{X}_0 . Also, setting $\mathcal{X}_0 \equiv \{\mathbf{x}_0\}$ in (2.2), it is apparent that $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ is a translation of the set-valued integral in (2.2). Thus, classifying $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ amounts to classifying the second summand in (2.2).

³As such, (2.12) has a difference of convex objective, and by the Weierstrass extreme value theorem, the maximum is achieved.

2.1.1 $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ is a Zonoid

A *zonoid* is a compact convex set that is defined as the range of an *atom free* vector measure (see Ch. 1.6.5). Affine image of a zonoid is a zonoid. Minkowski sum of zonoids is also a zonoid. We refer the readers to [74–76], [77, Sec. I] for more details on the properties of a zonoid. By slight abuse of nomenclature, in this dissertation we use the term zonoid up to translation, i.e., we refer to the translation of zonoids as zonoids (instead of using another term such as “zonoidal translates”). Let us mention a few examples.

Any compact convex symmetric set in \mathbb{R}^2 is a zonoid. In dimensions three or more, all ℓ_p norm balls for $p \geq 2$ are zonoids.

An alternative way to think about the zonoid is to view it as the limiting set (convergence with respect to the two-sided Hausdorff distance, see e.g., [78, Appendix B]) of the Minkowski sum of line segments, i.e., the limit of a sequence of *zonotopes* [52, 53, 74]. Formally, given a Hausdorff convergent sequence of zonotopes $\{\mathcal{Z}_j\}$, the zonoid \mathcal{Z}_∞ is $\mathcal{Z}_\infty := \lim_{j \rightarrow \infty} \mathcal{Z}_j$, where $\mathcal{Z}_j := \sum_{i=1}^{n(j)} [\mathbf{a}_{ij}, \mathbf{b}_{ij}]$, $\mathbf{a}_{ij}, \mathbf{b}_{ij} \in \mathbb{R}^d$, for some $\mathbf{a}_{ij} \leq \mathbf{b}_{ij}$ (element-wise vector inequality), and a suitable mapping $n : \mathbb{Z}_+ \mapsto \mathbb{Z}_+$. Our analysis will make use of this viewpoint in Ch. 2.2.1. Our main result in this subsection is the following.

Theorem 2.2. *The reach set (1.30) with $\mathcal{X}_0 \equiv \{\mathbf{x}_0\}$ is a zonoid.*

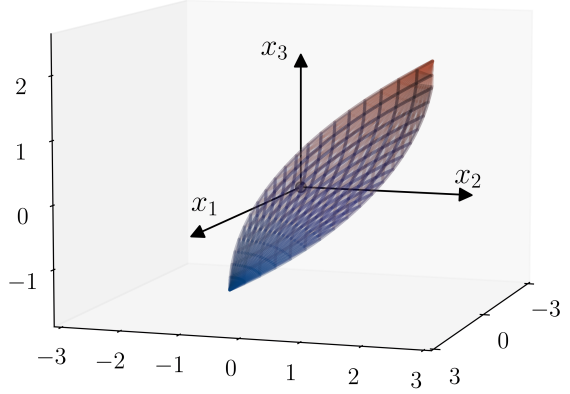


Figure 2.1: The “almond-shaped” integrator reach set $\mathcal{X}_t^\square(\{\mathbf{x}_0\}) \subset \mathbb{R}^3$ with $d = 3$, $m = 1$, $\mathbf{x}_0 = (0.1, 0.2, 0.3)^\top$, $\mathcal{U} \equiv [\alpha, \beta] = [-1, 1]$ at $t = 2.1$. The wireframes correspond to the upper and lower surfaces.

To appreciate Theorem 2.2 via the limiting viewpoint mentioned before, let us write

$$\mathcal{X}_t^\square(\{\mathbf{x}_0\}) = \underbrace{\exp(t\mathbf{A})\mathbf{x}_0 + \sum_{j=1}^m \nu_j \zeta_j(t)}_{\text{first term}} \dot{+} \underbrace{\sum_{j=1}^m \lim_{n \rightarrow \infty} \sum_{i=0}^n \frac{t}{n} \mu_j \boldsymbol{\xi}_j(t_i)}_{\text{second term}} [-1, 1], \quad (2.13)$$

where all summation symbols denote Minkowski sums. The first term in (2.13) denotes a translation. In the second term, the outer summation over index j arises by writing the Cartesian product (2.8) as the Minkowski sum $\mathcal{X}_1 \dot{+} \dots \dot{+} \mathcal{X}_m$. Furthermore, uniformly discretizing $[0, t]$ into n subintervals $[(i-1)t/n, it/n)$, $i = 1, \dots, n$, we write $\int_0^t \exp((t-s)\mathbf{A}_j) \mathbf{b}_j [-\mu_j, \mu_j] ds$ (or its equivalent $\int_0^t \exp(s\mathbf{A}_j) \mathbf{b}_j [-\mu_j, \mu_j] ds$) as the limit of the Minkowski sum over index i . Geometrically, the innermost summands in the second term denote non-uniformly rotated and scaled line intervals in \mathbb{R}^j . In other words, the second term in (2.13) is a Minkowski sum of m sets, each of these sets being the limit of a sequence of sets $\{\mathcal{Z}_n\}$ comprising of zonotopes

$$\mathcal{Z}_n := \sum_{i=0}^n \frac{t}{n} \mu_j \boldsymbol{\xi}_j(t_i) [-1, 1],$$

which are the Minkowski sum of $n+1$ line segments. Since $\lim_{n \rightarrow \infty} \mathcal{Z}_n$ is a zonoid, the second term in (2.13) is a Minkowski sum of m zonoids, and is therefore a zonoid [74, Thm. 1.5]. The entire right hand side of (2.13), then, is translation of a zonoid, and hence a zonoid.

Remark 2.1. *If $\mathcal{X}_0 \subset \mathbb{R}^d$ is not singleton, but instead a zonoid, then $\mathcal{X}_t^\square(\mathcal{X}_0)$ is still a (translated) zonoid. To see this, notice from (2.2) and (2.10) that*

$$\mathcal{X}_t^\square(\mathcal{X}_0) = \exp(t\mathbf{A})\mathcal{X}_0 \dot{+} \mathcal{X}_t^\square(\{\mathbf{0}\}), \quad (2.14)$$

and that $\exp(t\mathbf{A})\mathcal{X}_0$, being linear image of a zonoid, is a zonoid [74, Lemma 1.4]. Thus, (2.14) being Minkowski sum of zonoids, is a zonoid too [74, Thm. 1.5], up to translation.

In the following, we derive formulae for the boundary (Proposition 2.3 and Ch.

2.1.3) and volume (Theorem 2.6) of the integrator reach set with $\mathcal{X}_0 = \{\mathbf{x}_0\}$ (singleton set). From (2.14), it is clear that one cannot expect similar closed form formulae for arbitrary compact (or even arbitrary compact convex) \mathcal{X}_0 . In this sense, our closed form formulae are as general as one might hope for. For a specific non-singleton \mathcal{X}_0 , one can use these formulae to first derive the boundary (resp. volume) of $\mathcal{X}_t^\square(\{\mathbf{0}\})$, and then use (2.14) to get *numerical estimates* for the boundary (resp. volume) of $\mathcal{X}_t^\square(\mathcal{X}_0)$ (cf. Remark 2.2).

Parametric representation of $\partial\mathcal{X}_t^\square(\{\mathbf{x}_0\})$

In theorem 2.3 below, we derive a parametric representation of $\mathbf{x}^{\text{bdy}} \in \partial\mathcal{X}_t^\square(\{\mathbf{x}_0\})$, the boundary of the reach set. Then we use this representation to establish semialgebraicity of $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ in Theorem 2.5 that follows.

Theorem 2.3. *For relative degree vector $\mathbf{r} = (r_1, \dots, r_m)^\top$, and fixed $\mathbf{x}_0 \in \mathbb{R}^d$ comprising of subvectors $\mathbf{x}_{j0} \in \mathbb{R}^{r_j}$ where $j \in \llbracket m \rrbracket$, consider the reach set (2.2) with singleton $\mathcal{X}_0 \equiv \{\mathbf{x}_0\}$ and compact $\mathcal{U} \subset \mathbb{R}^m$. For $j \in \llbracket m \rrbracket$, define μ_1, \dots, μ_m and ν_1, \dots, ν_m as in (2.3)-(2.4). Let the indicator function $\mathbf{1}_{k \leq \ell} := 1$ for $k \leq \ell$, and $:= 0$ otherwise. Then the components of*

$$\mathbf{x}^{\text{bdy}} = \begin{pmatrix} \mathbf{x}_1^{\text{bdy}} \\ \mathbf{x}_2^{\text{bdy}} \\ \vdots \\ \mathbf{x}_m^{\text{bdy}} \end{pmatrix} \in \partial\mathcal{X}_t^\square(\{\mathbf{x}_0\}), \quad \mathbf{x}_j^{\text{bdy}} \in \mathbb{R}^{r_j}, \quad j \in \llbracket m \rrbracket,$$

admit parametric representation in terms of the parameters $\boldsymbol{\sigma}_j = (\sigma_1, \sigma_2, \dots, \sigma_{r_j-1}) \in \mathcal{W}_{jt} \subset \mathbb{R}^{r_j-1}$ where \mathcal{W}_{jt} is the Weyl chamber:

$$\mathcal{W}_{jt} := \{\boldsymbol{\sigma} \in \mathbb{R}^{n-1} \mid 0 \leq \sigma_1 \leq \sigma_2 \leq \dots \leq \sigma_{r_j-1} \leq t\}. \tag{2.15}$$

This parameterization is given by

$$\begin{aligned}
 x_{j,k}^{\text{bdy}}(\boldsymbol{\sigma}) &= \sum_{\ell=1}^{r_j} \mathbf{1}_{k \leq \ell} \frac{t^{\ell-k}}{(\ell-k)!} \mathbf{x}_{j0,\ell} + \frac{\nu_j t^{r_j-k+1}}{(r_j-k+1)!} \\
 &\quad \pm \frac{\mu_j}{(r_j-k+1)!} \left\{ (-1)^{r_j-1} t^{r_j-k+1} + 2 \sum_{q=1}^{r_j-1} (-1)^{q+1} \sigma_q^{r_j-k+1} \right\}, \quad (2.16)
 \end{aligned}$$

where $\mathbf{x}_{j,k}^{\text{bdy}}(\boldsymbol{\sigma}_j)$ denotes the k th component of the j th subvector $\mathbf{x}_j^{\text{bdy}}$ for $k \in \llbracket r_j \rrbracket$ archived for parameter vector $\boldsymbol{\sigma}_j$, and $x_{j0,\ell}$ is the ℓ -th component of the initial state \mathbf{x}_{j0} .

The proof of the above is deferred to Appendix B.3. The following is a consequence of the \pm appearing in (2.16).

Corollary 2.4. *The single input integrator reach set $\mathcal{X}_{jt}(\{\mathbf{x}_0\}) \subset \mathbb{R}^{r_j}$ has two bounding surfaces for each $j \in \llbracket m \rrbracket$. In other words, there exist $p_j^{\text{upper}}, p_j^{\text{lower}} : \mathbb{R}^{r_j} \mapsto \mathbb{R}$ such that*

$$\mathcal{X}_{jt}(\{\mathbf{x}_0\}) = \{\mathbf{x} \in \mathbb{R}^{r_j} \mid p_j^{\text{upper}}(\mathbf{x}) \leq 0, p_j^{\text{lower}}(\mathbf{x}) \leq 0\},$$

with boundary $\partial \mathcal{X}_{jt}(\{\mathbf{x}_0\}) = \{\mathbf{x} \in \mathbb{R}^{r_j} \mid p_j^{\text{upper}}(\mathbf{x}) = 0\} \cup \{\mathbf{x} \in \mathbb{R}^{r_j} \mid p_j^{\text{lower}}(\mathbf{x}) = 0\}$.

During the proof of Theorem 2.5 stated below, it will turn out that in fact $p_j^{\text{upper}}, p_j^{\text{lower}} \in \mathbb{R}[x_1, \dots, x_{r_j}]$ for all $j \in \llbracket m \rrbracket$. In words, $p_j^{\text{upper}}, p_j^{\text{lower}}$ are real algebraic hyper-surfaces for all $j \in \llbracket m \rrbracket$.

Let us exemplify the parameterization (2.16) for the case $\mathbf{r} = (r_1, r_2)^\top = (2, 3)^\top$. In this case,

$$\begin{pmatrix} \mathbf{x}_{1,1}^{\text{bdy}}(\sigma_1) \\ \mathbf{x}_{1,1}^{\text{bdy}}(\sigma_1) \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{10,1} + t\mathbf{x}_{10,2} + \nu_1(t^2/2) \pm \mu_1(\sigma_1^2 - t^2/2) \\ \mathbf{x}_{10,2} + \nu_1 t \pm \mu_1(2\sigma_1 - t) \end{pmatrix}, \quad (2.17)$$

and⁴

$$\begin{pmatrix} \mathbf{x}_{2,1}^{\text{bdy}}(\boldsymbol{\sigma}_2) \\ \mathbf{x}_{2,2}^{\text{bdy}}(\boldsymbol{\sigma}_2) \\ \mathbf{x}_{2,3}^{\text{bdy}}(\boldsymbol{\sigma}_2) \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{20,1} + t\mathbf{x}_{20,2} + (t^2/2)\mathbf{x}_{20,3} \\ +\nu_2(t^3/6) \pm \mu_2(t^3/6 + 2\sigma_1^3/6 - 2\sigma_2^3/6) \\ \mathbf{x}_{20,2} + t\mathbf{x}_{20,3} + \nu_2(t^2/2) \pm \mu_2(t^2/2 \\ + 2\sigma_1^2/2 - 2\sigma_2^2/2) \\ \mathbf{x}_{20,3} + \nu_2 t \pm \mu_2(t + 2\sigma_1 - 2\sigma_2) \end{pmatrix}. \quad (2.18)$$

In (2.17), taking plus (resp. minus) signs in each of component gives the parametric representation of the curve $p_1^{\text{upper}} = 0$ (resp. $p_1^{\text{lower}} = 0$). These curves are as in [78, Fig. 1(a)], and their union defines $\partial\mathcal{X}_{1t}$. We note that the parameterization (2.17) appeared in [79, p. 111].

Likewise, in (2.18), taking plus (resp. minus) signs in each of component gives the parametric representation of the surface $p_2^{\text{upper}}(\mathbf{x}) = 0$ (resp. $p_2^{\text{lower}} = 0$). The resulting set \mathcal{X}_{2t} is the triple integrator reach set, and is shown in Fig. 2.1.

2.1.2 $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ is semialgebraic

A set in \mathbb{R}^d is called *basic semialgebraic* if it can be written as a finite conjunction of polynomial inequalities and equalities, the polynomials being in $\mathbb{R}[x_1, \dots, x_d]$. Finite union of basic semialgebraic sets is called a *semialgebraic set*. A semialgebraic set need not be basic semialgebraic; see e.g., [80, Example 2.2].

Semialgebraic sets are closed under finitely many unions and intersections, complement, topological closure, polynomial mapping including projection [81, 82], and Cartesian product. For details on semialgebraic sets, we refer the readers to [83, Ch. 2]; see [84, Appendix A.4.4] for a short summary.

⁴As a reminder, We use bold-faced small letters for vectors and bold-faced capital letters for matrices.

Now we come to the main result of this subsection.

Theorem 2.5. *The reach set (2.2) with $\mathcal{X}_0 \equiv \{\mathbf{x}_0\}$ is semialgebraic.*

Let us illustrate the bounding curves and surfaces for (2.17) and (2.18) respectively, in the implicit form. Eliminating the parameter σ_1 from (2.17) reveals that $p_1^{\text{upper}}, p_1^{\text{lower}}$ are parabolas. In particular,

$$p_1^{\text{upper}}(\mathbf{x}_{1,1}^{\text{bdy}}, \mathbf{x}_{1,2}^{\text{bdy}}) = \frac{1}{4} \left(\frac{\mathbf{x}_{1,2}^{\text{bdy}} - \mathbf{x}_{10,2} - \nu_1 t}{\mu_1} + t \right)^2 - \frac{\mathbf{x}_{1,1}^{\text{bdy}} - \mathbf{x}_{10,1} - t\mathbf{x}_{10,2} - \nu_1 \frac{t^2}{2}}{\mu_1} - \frac{t^2}{2}, \quad (2.19a)$$

$$p_1^{\text{lower}}(\mathbf{x}_{1,1}^{\text{bdy}}, \mathbf{x}_{1,2}^{\text{bdy}}) = -\frac{1}{4} \left(-\frac{\mathbf{x}_{1,2}^{\text{bdy}} - \mathbf{x}_{10,2} - \nu_1 t}{\mu_1} + t \right)^2 - \frac{\mathbf{x}_{1,1}^{\text{bdy}} - \mathbf{x}_{10,1} - t\mathbf{x}_{10,2} - \nu_1 \frac{t^2}{2}}{\mu_1} + \frac{t^2}{2}. \quad (2.19b)$$

Similarly, eliminating the parameters σ_1, σ_2 from (2.18) reveals that $p_2^{\text{upper}}, p_2^{\text{lower}}$ are quartic polynomials. In particular,

$$\begin{aligned} p_2^{\text{upper}}(\mathbf{x}_{2,1}^{\text{bdy}}, \mathbf{x}_{2,2}^{\text{bdy}}, \mathbf{x}_{2,3}^{\text{bdy}}) = & \\ & \frac{1}{16} \left(\frac{\mathbf{x}_{2,3}^{\text{bdy}} - \mathbf{x}_{20,3} - \nu_2 t}{\mu_2} - t \right)^4 + 3 \left(\frac{\mathbf{x}_{2,2}^{\text{bdy}} - \mathbf{x}_{20,2} - t\mathbf{x}_{20,3} - \nu_2 \frac{t^2}{2}}{\mu_2} - \frac{t^2}{2} \right)^2 \\ & - 6 \left(\frac{\mathbf{x}_{2,1}^{\text{bdy}} - \mathbf{x}_{20,1} - t\mathbf{x}_{20,2} - \frac{t^2}{2}\mathbf{x}_{20,3} - \nu_2 \frac{t^3}{6}}{\mu_2} - \frac{t^3}{6} \right) \times \left(\frac{\mathbf{x}_{2,3}^{\text{bdy}} - \mathbf{x}_{20,3} - \nu_2 t}{\mu_2} - t \right), \quad (2.20) \end{aligned}$$

and the formula for $p_2^{\text{lower}}(\mathbf{x}_{2,1}^{\text{bdy}}, \mathbf{x}_{2,2}^{\text{bdy}}, \mathbf{x}_{2,3}^{\text{bdy}})$ follows mutatis mutandis.

A natural question is whether one can generalize the implicitizations as in (2.19), (2.20) to arbitrary state dimensions. This is what we address next.

2.1.3 Implicitization of $\partial\mathcal{X}_t^\square(\{\mathbf{x}_0\})$

To derive the implicit equations for the bounding algebraic hypersurfaces $p_j^{\text{upper}}, p_j^{\text{lower}} \in \mathbb{R}[x_1, \dots, x_{r_j}]$ for all $j \in \llbracket m \rrbracket$, we need to eliminate the parameters $(\sigma_1, \sigma_2, \dots, \sigma_{r_j-1})$

from (2.16). For this purpose, it is helpful to write (2.16) succinctly as

$$\rho_{j,k}^{\pm} = \sum_{q=1}^{r_j-1} (-1)^{q+1} \sigma_q^{r_j-k+1}, \quad k \in \llbracket r_j \rrbracket, \quad (2.21)$$

where

$$\rho_{j,k}^{\pm} := \frac{(r_j - k + 1)!}{2\mu_j} \left\{ \mathbf{x}_{j,k}^{\text{bdy}} - \sum_{\ell=1}^{r_j} \mathbf{1}_{k \leq \ell} \frac{t^{\ell-k}}{(\ell-k)!} \mathbf{x}_{j,0,\ell} \right\} - \frac{1}{2} \left\{ \pm (-1)^{r_j-1} t^{r_j-k+1} + \frac{V_j}{\mu_j} t^{r_j-k+1} \right\}. \quad (2.22)$$

To simplify the rather unpleasant notation $\rho_{j,k}^{\pm}$, we will only address the $m = 1$ case. In (2.21), this allows us to replace r_j by d , and to drop the subscript j from the ρ 's. This does not invite any loss of generality in terms of implicitization since post derivation, we can be replaced by r_j to recover the respective p_j 's.

With slight abuse of notation, we will also drop the superscript \pm from the ρ 's in (2.21). Recall that the plus (resp. minus) superscript in the ρ 's indicates p_j^{upper} (resp. p_j^{lower}). From (2.22), it is clear that in either case, the $\rho_{j,k}$ is affine in $\mathbf{x}_{j,k}^{\text{bdy}}$, which is the k th coordinate of the boundary point for the j th block. Importantly, for $k \in \llbracket r_j \rrbracket$, the quantity $\rho_{j,k}$ does not depend on any other component of the boundary point than the k th component. Again, the plus-minus superscripts can be added back post implicitization. Thus, the notationally simplified version of (2.21) that suffices for implicitization, is

$$\rho_k = \sum_{q=1}^{d-1} (-1)^{q+1} \sigma_q^{d-k+1}, \quad k = 1, \dots, d, \quad (2.23)$$

which is a system of d homogeneous polynomials in variables $(\sigma_1, \sigma_2, \dots, \sigma_{d-1})$. The objective is to derive the implicitized polynomial $\wp(\rho_1, \rho_2, \dots, \rho_d)$ associated with (2.23).

When $d = 2$, the parameterization (2.23) becomes

$$\rho_1 = \sigma_1^2, \quad \rho_2 = \sigma_1,$$

and we get degree 2 implicitized polynomial

$$\wp(\rho_1, \rho_2) = \rho_2^2 - \rho_1 = 0. \quad (2.24)$$

For $k = 1, 2$, substituting for the ρ_1, ρ_2 in (2.24) from (2.22) with appropriate plus-minus signs recovers (2.19). When $d = 3$, the parameterization (2.23) becomes

$$\rho_1 = \sigma_1^3 - \sigma_2^3, \quad \rho_2 = \sigma_1^2 - \sigma_2^2, \quad \rho_3 = \sigma_1 - \sigma_2,$$

and using elementary algebra, we get degree 4 implicitized polynomial

$$\wp(\rho_1, \rho_2, \rho_3) = \rho_3^4 - 4\rho_3\rho_1 + 3\rho_2^2 = 0. \quad (2.25)$$

As before, for $k = 1, 2, 3$, substituting for the ρ_1, ρ_2, ρ_3 in (2.25) from (2.22) with appropriate plus-minus signs recovers (2.20). However, for $d = 4$ or higher, it is practically impossible to derive the implicitization via brute force algebra.

A principled way to implicitize (2.23) is due to G. Zaimi [85], and starts with defining $\lambda_k := \rho_{d-k+1}$ for $k = 1, \dots, d$. Introduce the sequence $A_k(\sigma_1, \sigma_2, \dots, \sigma_{d-1})$ via the generating function (see e.g., [86, Ch. 1])

$$F(\tau) = \sum_{k \geq 0} A_k \tau^k = \frac{(1 - \sigma_1 \tau)(1 - \sigma_3 \tau) \cdots}{(1 - \sigma_2 \tau)(1 - \sigma_4 \tau) \cdots}. \quad (2.26)$$

Taking the logarithmic derivative of (2.26), and then using the generating functions $(1 - \sigma_q \tau)^{-1} = \sum_{k \geq 0} (\sigma_q \tau)^k$ for all $q = 1, \dots, d - 1$, yields

$$\frac{F'(\tau)}{F(\tau)} = -\sigma_1 \sum_{k \geq 0} (\sigma_1 \tau)^k + \sigma_2 \sum_{k \geq 0} (\sigma_2 \tau)^k - \sigma_3 \sum_{k \geq 0} (\sigma_3 \tau)^k + \cdots. \quad (2.27)$$

Integrating (2.27) with respect to τ , we obtain

$$F(\tau) = \exp\left(-\sum_{k=1}^d \frac{\lambda_k}{k} \tau^k\right). \quad (2.28)$$

Equating (2.26) and (2.28) allows us to compute A_k as a degree k polynomial of the λ 's. On the other hand, since the generating function (2.26) is a rational function

with denominator polynomial of degree $\delta := \lfloor \frac{d-1}{2} \rfloor$, the following Hankel determinant vanishes⁵

$$\det[A_{d-2\delta+i+j}]_{i,j=0}^{\delta} = 0. \quad (2.29)$$

Substituting the A_k 's obtained as degree k polynomials of the λ 's into (2.29) gives an implicit polynomial in indeterminate $(\lambda_1, \dots, \lambda_d)$ of degree $(\delta + 1)(d - \delta)$. Finally, reverting back the λ 's to the ρ 's result in the desired implicit polynomial $\wp(\rho_1, \rho_2, \dots, \rho_d)$, which is also of degree $(\delta + 1)(d - \delta)$.

For instance, when $d = 3$, the relation (2.29) becomes

$$\det \begin{pmatrix} A_1 & A_2 \\ A_2 & A_3 \end{pmatrix} = 0. \quad (2.30)$$

In this case, equating (2.26) and (2.28) gives

$$A_1 = -\lambda_1, \quad A_2 = \frac{1}{2}\lambda_1^2 - \frac{1}{2}\lambda_2, \quad A_3 = -\frac{1}{6}\lambda_1^3 + \frac{1}{2}\lambda_1\lambda_2 - \frac{1}{3}\lambda_3.$$

Substituting these back in (2.30) yields the quartic polynomial $\lambda_1^4 + 3\lambda_2^2 - 4\lambda_3\lambda_1 = 0$, which under the mapping $(\lambda_1, \lambda_2, \lambda_3) \mapsto (\rho_3, \rho_2, \rho_1)$ recovers (2.25), and thus (2.20).

In summary, (2.29) is the desired implicitization of the bounding hypersurfaces of the single input integrator reach set (up to the change of variables). The Cartesian product of these implicit hypersurfaces gives the implicitization in the multi input case.

2.1.4 Polar dual of $\mathcal{X}_t^{\square}(\{\mathbf{x}_0\})$

From convex geometry standpoint, it is natural to ask what kind of characterization is possible for the polar dual (see Ch. 1.6.4) of the integrator reach set \mathcal{X} or \mathcal{X}^{\square} . We know in general that \mathcal{X}° will be compact convex. Depending on the choice of $\mathbf{x}_0, \mathcal{U}$

⁵This result goes back to Kronecker [87]. See also [88, p. 5, Lemma III].

and t , the set $\mathcal{X}(\{\mathbf{x}_0\}, t)$ may not contain the origin, and thus the bipolar

$$(\mathcal{X}_t(\{\mathbf{x}_0\}))^{\circ\circ} = \text{closure}(\text{conv}(\mathcal{X}_t(\{\mathbf{x}_0\}) \cup \{\mathbf{0}\})),$$

that is, we do not have the involution in general.

Furthermore, since $\mathcal{X}_t(\{\mathbf{x}_0\})$ is semialgebraic from Ch. 2.1.2, so must be its polar dual $(\mathcal{X}_t(\{\mathbf{x}_0\}))^\circ$; see e.g., [84, Ch. 5, Sec. 5.2.2].

We also know from Ch. 2.1.1 that $\mathcal{X}(\{\mathbf{x}_0\}, t)$ is a zonoid. However, the polar of a zonoid is not a zonoid in general [89, 90], and we should not expect $(\mathcal{X}(\{\mathbf{x}_0\}, t))^\circ$ to be one. Fig. 2.2 shows $\mathcal{X}(\{\mathbf{x}_0\}, t)$ and $(\mathcal{X}(\{\mathbf{x}_0\}, t))^\circ$ for the double integrator ($d = 2$, $m = 1$).

2.1.5 Summary of taxonomy

So far we explained that the compact convex set $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ is semialgebraic, and a translated zonoid. Two well-known subclasses of convex semialgebraic sets are the *spectrahedra* and the *spectrahedral shadows*. The spectrahedra, a.k.a. *linear matrix inequality (LMI) representable sets* are affine slices of the symmetric positive semidefinite cone. The spectrahedral shadows, a.k.a. *lifted LMI or semidefinite representable sets* are the projections of spectrahedra. The spectrahedral shadows subsume the class of spectrahedra; e.g., the set $\{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^4 + x_2^4 \leq 1\}$ is a spectrahedral shadow but not a spectrahedron. The polar duals of spectrahedra are spectrahedral shadows [84, Ch. 5, Sec. 5.5].

We note that the integrator reach set is not a spectrahedron. To see this, we resort to the contrapositive of [91, Thm. 3.1]. Specifically, the number of intersections made by a generic line passing through an interior point of the d -dimensional integrator reach set with its real algebraic boundary is not equal to the degree of the bounding algebraic hypersurfaces, the latter we know from Ch. 2.1.3 to be $(\lfloor \frac{d-1}{2} \rfloor + 1)(d - \lfloor \frac{d-1}{2} \rfloor)$.

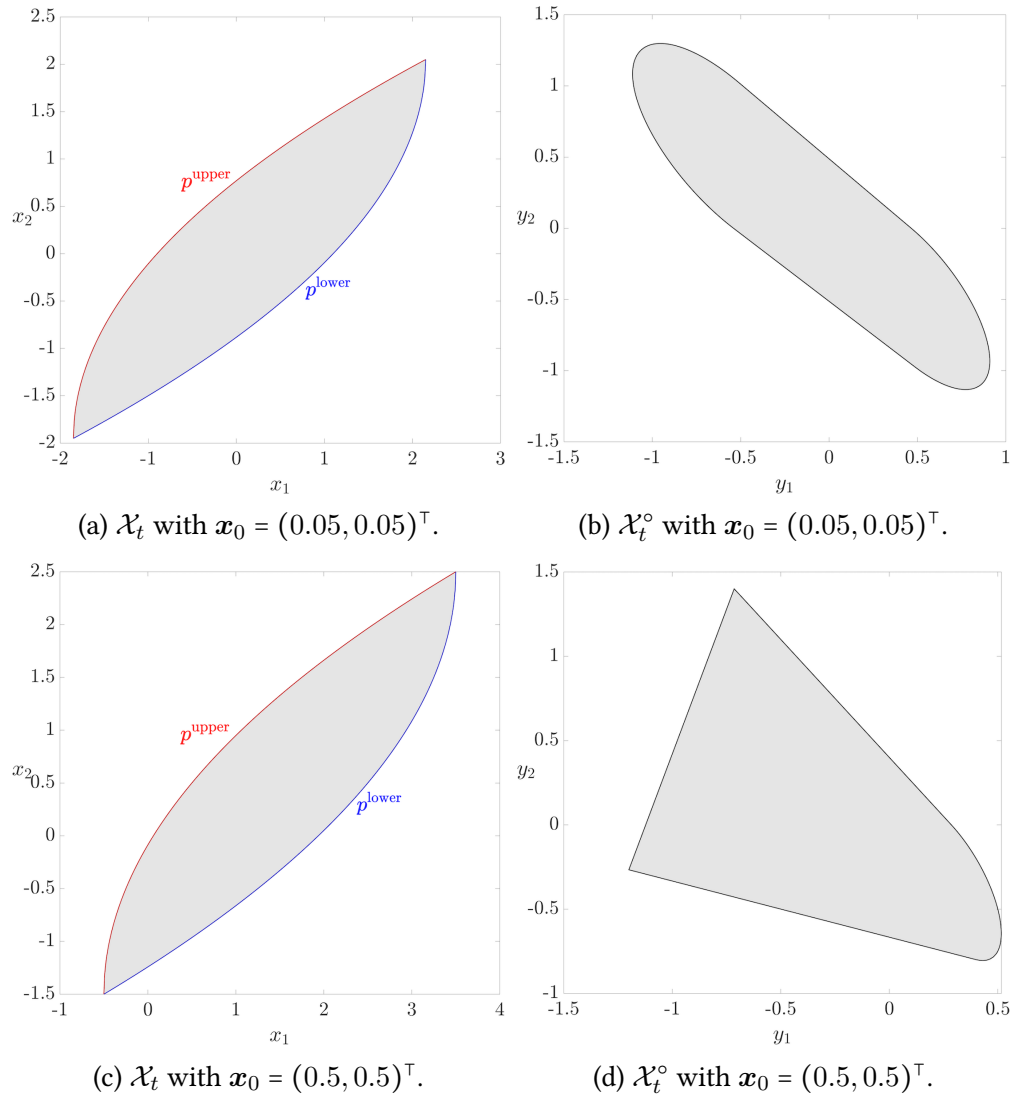
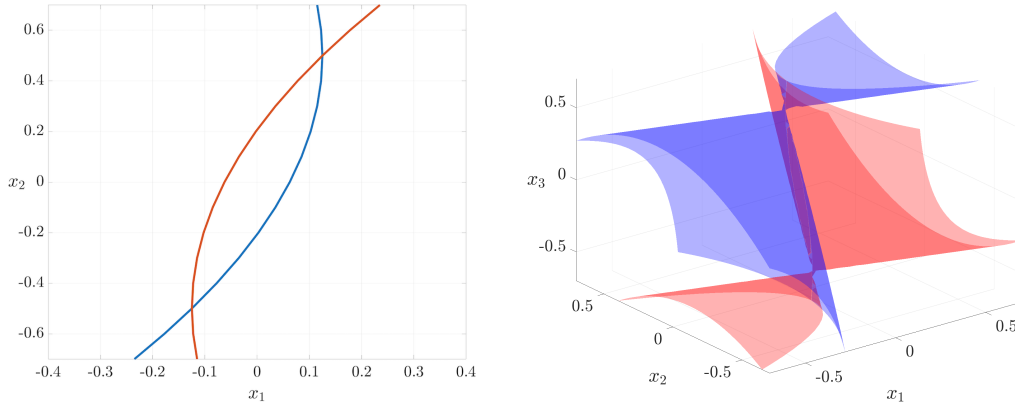


Figure 2.2: The double integrator reach set $\mathcal{X}_t(\{\mathbf{x}_0\})$ and its polar dual $(\mathcal{X}_t(\{\mathbf{x}_0\}))^\circ$ for different \mathbf{x}_0 at $t = 2$, $\mathcal{U} \equiv [\alpha, \beta] = [-1, 1]$. The curves $p^{\text{upper}}, p^{\text{lower}}$ defining the reach set boundary (see Corollary 2.4 and the discussion thereafter) are shown too.



(a) Real algebraic curves $p^{\text{upper}}, p^{\text{lower}}$ for the double integrator. (b) Real algebraic surfaces $p^{\text{upper}}, p^{\text{lower}}$ for the triple integrator.

Figure 2.3: The bounding polynomials for the double and triple integrator reach sets at $t = 0.5$ with $x_0 = \mathbf{0}$ and $\mu = 1$.

In other words, the integrator reach set is not rigidly convex, see [91, Sec. 3.1 and 3.2].

Fig. 2.3 helps visualize this for $m = 1$. From Fig. 2.3a, we observe that a generic line for $d = 2$ has 4 intersections with the bounding real algebraic curves whereas from (2.19), we know that $p^{\text{upper}}, p^{\text{lower}}$ are degree 2 polynomials. Likewise, Fig. 2.3b reveals that a generic line for $d = 3$ has 6 intersections with the bounding real algebraic surfaces whereas from (2.20), we know that the polynomials $p^{\text{upper}}, p^{\text{lower}}$ in this case, are of degree 4.

Could the integrator reach set be spectrahedral shadow? Some calculations show that *sufficient* conditions as in [92] do not seem to hold. However, this remains far from conclusive. We summarize our taxonomy results in Fig. 2.4; the highlighted region shows where the integrator reach set belongs. To answer whether this highlighted region can be further narrowed down, seems significantly more challenging.

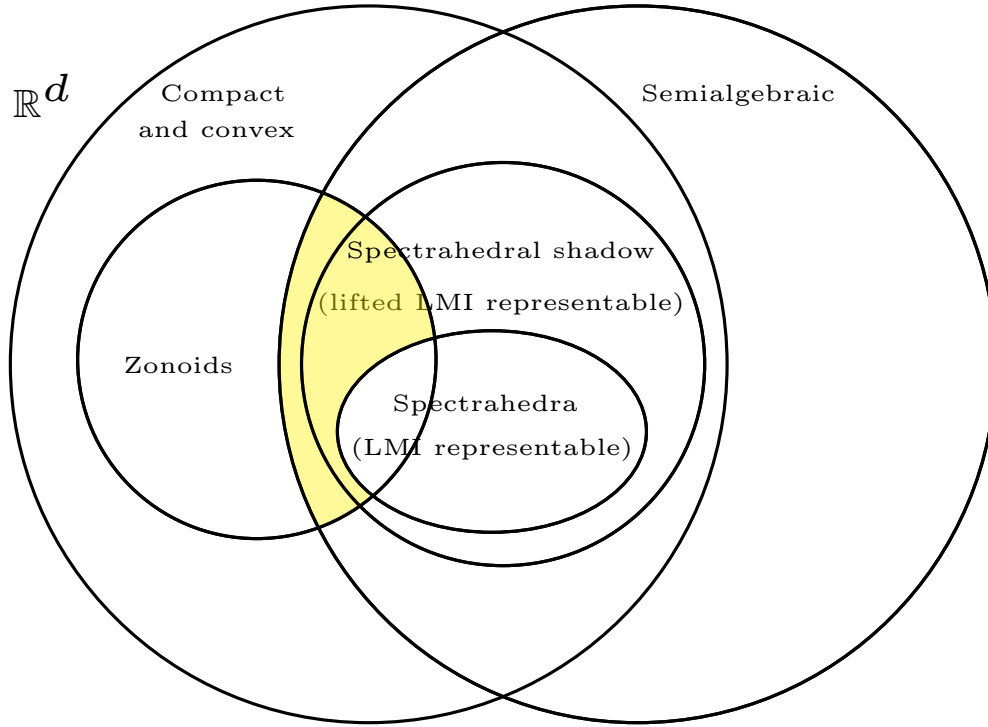


Figure 2.4: The summary of the taxonomy for the integrator reach set.

2.2 Size of $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ with Time Invariant Set-Valued Inputs

Inputs

We next quantify the “size” of the reach set $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ by computing two functionals: its d -dimensional volume (Ch. 2.2.1), and its diameter or maximum width (Ch. 2.2.2). In Sections 2.2.3 and 2.2.4, we discuss how these functionals scale with the state dimension d .

2.2.1 Volume

The following result gives the volume formula for the integrator reach set.

Theorem 2.6. Fix $\mathbf{x}_0 \in \mathbb{R}^d$, let $\mathcal{X}_0 \equiv \{\mathbf{x}_0\}$ and compact $\mathcal{U} \subset \mathbb{R}^m$. Consider the integrator dynamics (1.6)-(1.7) with d states, m inputs, and relative degree vector $\mathbf{r} = (r_1, r_2, \dots, r_m)^\top$. Define μ_1, \dots, μ_m as in (2.3)-(2.4). Then the d -dimensional volume of

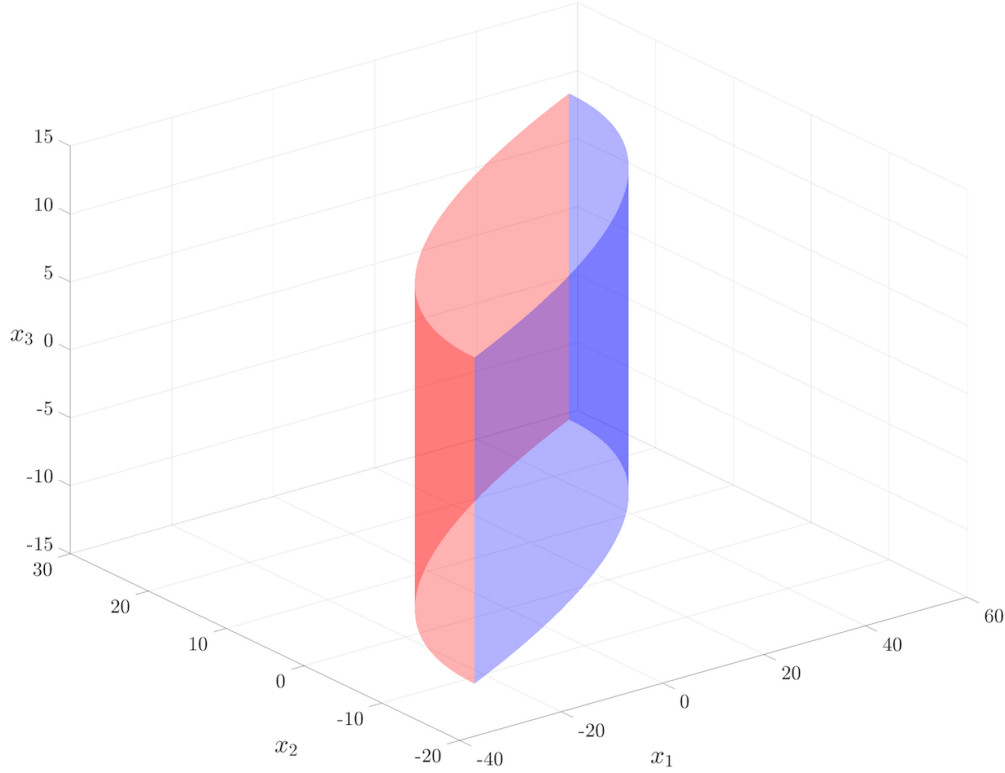


Figure 2.5: The integrator reach set $\mathcal{X}_{t=4}^\square(\{\mathbf{x}_0\})$ with $m = 2$, $\mathbf{r} = (2, 1)^\top$, $\mathbf{x}_0 = (1, 1, 0)^\top$, $[\alpha_1, \beta_1] = [-5, 5]$, $[\alpha_2, \beta_2] = [-3, 3]$.

the integrator reach set (2.2) at time $t > 0$ is

$$\text{vol}(\mathcal{X}_t^\square(\{\mathbf{x}_0\})) = 2^d \prod_{j=1}^m \left\{ \mu_j^{r_j} t^{r_j(r_j+1)/2} \prod_{k=1}^{r_j-1} \frac{k!}{(2k+1)!} \right\}. \quad (2.31)$$

For a simple illustration of Theorem 2.6, consider $d = 3$, $m = 2$ with $\mathbf{r} = (2, 1)^\top$. The corresponding reach set $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ at $t = 4$ is shown in Fig. 2.5 for $\mathbf{x}_0 = (1, 1, 0)^\top$, $\mathcal{U} = [-5 \times 5] \times [-3, 3]$. Here $\mu_1 = 5$ and $\mu_2 = 3$.

This reach set, being a direct product of the double integrator reach set \mathcal{X}_{1t} (cf. Fig. 2.2) and the single integrator reach set $\mathcal{X}_{2t} = \{\mathbf{x}_{0,3}\} \dot{+} [-\mu_2 t, \mu_2 t]$, is a cylinder⁶. In [78], we explicitly derived that $\text{vol}(\mathcal{X}_{1t}) = \frac{2}{3} \mu_1^2 t^3$, and therefore, the volume of this

⁶Here, the notation $\mathbf{x}_{0,3}$ stands for the third component of vector \mathbf{x}_0 .

cylindrical set must be equal to “height of the cylinder \times cross sectional area”, i.e.,

$$2\mu_2 t \times \frac{2}{3}\mu_1^2 t^3 = \frac{4}{3}\mu_1^2 \mu_2 t^4.$$

Indeed, a direct application of the formula (2.31) recovers the above expression.

Remark 2.2. *If the initial set \mathcal{X}_0 is not singleton, then computing the volume of the reach set (2.2) requires us to compute the volume of a Minkowski sum. Notice that*

$$\begin{aligned} \text{vol}(\exp(t\mathbf{A})\mathcal{X}_0) &= |\det(\exp(t\mathbf{A}))| \text{vol}(\mathcal{X}_0) = \exp(\text{trace}(t\mathbf{A})) \text{vol}(\mathcal{X}_0) \\ &= \exp\left(\sum_{j=1}^m \text{trace}(t\mathbf{A}_j)\right) \text{vol}(\mathcal{X}_0) = \text{vol}(\mathcal{X}_0), \end{aligned}$$

since from (1.7b), $\text{trace}(\mathbf{A}_j) = 0$ for all $j \in \llbracket m \rrbracket$. Therefore, combining (2.2), (2.31) with the classical Brunn-Minkowski inequality, we have a bound

$$(\text{vol}(\mathcal{X}(\mathcal{X}_0, t)))^{1/d} \geq (\text{vol}(\mathcal{X}_0))^{1/d} + 2 \left(\prod_{j=1}^m \left\{ \mu_j^{r_j} t^{r_j(r_j+1)/2} \prod_{k=1}^{r_j-1} \frac{k!}{(2k+1)!} \right\} \right)^{1/d}.$$

The above bound holds for any compact $\mathcal{X}_0 \subset \mathbb{R}^d$, not necessarily convex.

2.2.2 Diameter

We now focus on another measure of the “size” of the integrator reach set, namely its diameter, or maximal width.

By definition, the *width* [47, p. 42] of the reach set $\mathcal{X}_t^\square(\mathcal{X}_0)$, is

$$w_{\mathcal{X}_t^\square(\{\mathbf{x}_0\})}(\boldsymbol{\eta}) := h_{\mathcal{X}_t^\square(\mathcal{X}_0)}(\boldsymbol{\eta}) + h_{\mathcal{X}_t^\square(\mathcal{X}_0)}(-\boldsymbol{\eta}), \quad (2.32)$$

where $\boldsymbol{\eta} \in \mathbb{S}^{d-1}$ (the unit sphere embedded in \mathbb{R}^d), and the support function $h_{\mathcal{X}_t^\square(\mathcal{X}_0)}(\cdot)$ is given by (2.10). In other words, (2.32) gives the width of the reach set in the direction $\boldsymbol{\eta}$.

For singleton $\mathcal{X}_0 \equiv \{\mathbf{x}_0\}$, combining (2.10) and (2.32), we have

$$w_{\mathcal{X}_t^\square(\{\mathbf{x}_0\})}(\boldsymbol{\eta}) = \int_0^t \left\{ |\langle \boldsymbol{\eta}, \boldsymbol{\xi}(t-s) \rangle| + | \langle -\boldsymbol{\eta}, \boldsymbol{\xi}(t-s) \rangle | \right\} ds = 2 \int_0^t |\langle \boldsymbol{\eta}, \boldsymbol{\xi}(t-s) \rangle| ds, \quad (2.33)$$

where the last equality follows from the fact that $\boldsymbol{\xi}(t-s)$ in (2.5) is component-wise nonnegative for all $0 \leq s \leq t$.

The *diameter* of the reach set is its maximal width:

$$\text{diam}(\mathcal{X}_t(\mathcal{X}_0)) := \max_{\boldsymbol{\eta} \in \mathbb{S}^{d-1}} w_{\mathcal{X}_t(\mathcal{X}_0)}(\boldsymbol{\eta}). \quad (2.34)$$

Notice that (2.33) is a convex function of $\boldsymbol{\eta}$; see e.g., [93, p. 79]. Thus, computing (2.34) amounts to maximizing a convex function over the unit sphere. We next derive a closed form expression for (2.34).

Theorem 2.7. Fix $\mathbf{x}_0 \in \mathbb{R}^d$, let $\mathcal{X}_0 \equiv \{\mathbf{x}_0\}$ and compact $\mathcal{U} \subset \mathbb{R}^m$. Consider the integrator dynamics (1.6)-(1.7) with d states, m inputs, and relative degree vector $\mathbf{r} = (r_1, r_2, \dots, r_m)^\top$. Define μ_1, \dots, μ_m as in (2.3)-(2.4). The diameter of the integrator reach set (2.2) at time $t > 0$ is

$$\text{diam}(\mathcal{X}_t^\square(\{\mathbf{x}_0\})) = 2 \|\boldsymbol{\zeta}(t)\|_2 = 2 \left(\sum_{j=1}^m \mu_j^2 \|\boldsymbol{\zeta}_j\|^2 \right)^{1/2}, \quad (2.35)$$

wherein $\boldsymbol{\zeta}(t)$ is defined as in Ch. 1.6.3, and the i th component of the subvector $\boldsymbol{\zeta}_j(t) \in \mathbb{R}^{r_j}$ is

$$\int_0^t \frac{\sigma^{(r_j-i)}}{(r_j-i)!} ds = \frac{t^{r_j-i+1}}{(r_j-i+1)!}, \quad i = 1, 2, \dots, r_j. \quad (2.36)$$

To illustrate Theorem 2.7, consider the triple integrator with $d = 3$ and $m = 1$. In this case, $\mathcal{U} = [\alpha, \beta]$, $\mu := (\beta - \alpha)/2$, and we can parameterize the unit vector $\boldsymbol{\eta} \in \mathbb{S}^2$

as

$$\boldsymbol{\eta} \equiv \begin{pmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{pmatrix}, \quad \theta \in [0, \pi], \quad \phi \in [0, 2\pi).$$

Thus (2.34) reduces to

$$2\mu \max_{\substack{\theta \in [0, \pi] \\ \phi \in [0, 2\pi)}} \int_0^t |s^2 (\sin \theta \cos \phi) / 2 + s \sin \theta \sin \phi + \cos \theta| ds.$$

Furthermore, $\boldsymbol{\zeta}(t) = (t^3/6, t^2/2, t)^\top$, and we obtain

$$\boldsymbol{\eta}^{\max} = \begin{pmatrix} \sin \theta^{\max} \sin \phi^{\max} \\ \sin \theta^{\max} \cos \phi^{\max} \\ \cos \theta^{\max} \end{pmatrix} = \frac{\pm 1}{\sqrt{t^4 + 9t^2 + 36}} \begin{pmatrix} t^2 \\ 3t \\ 6 \end{pmatrix},$$

where \pm means that either all components are plus or all minus. Thus, the maximizing tuples $(\phi^{\max}, \theta^{\max}) \in [0, \pi] \times [0, 2\pi)$ are given by

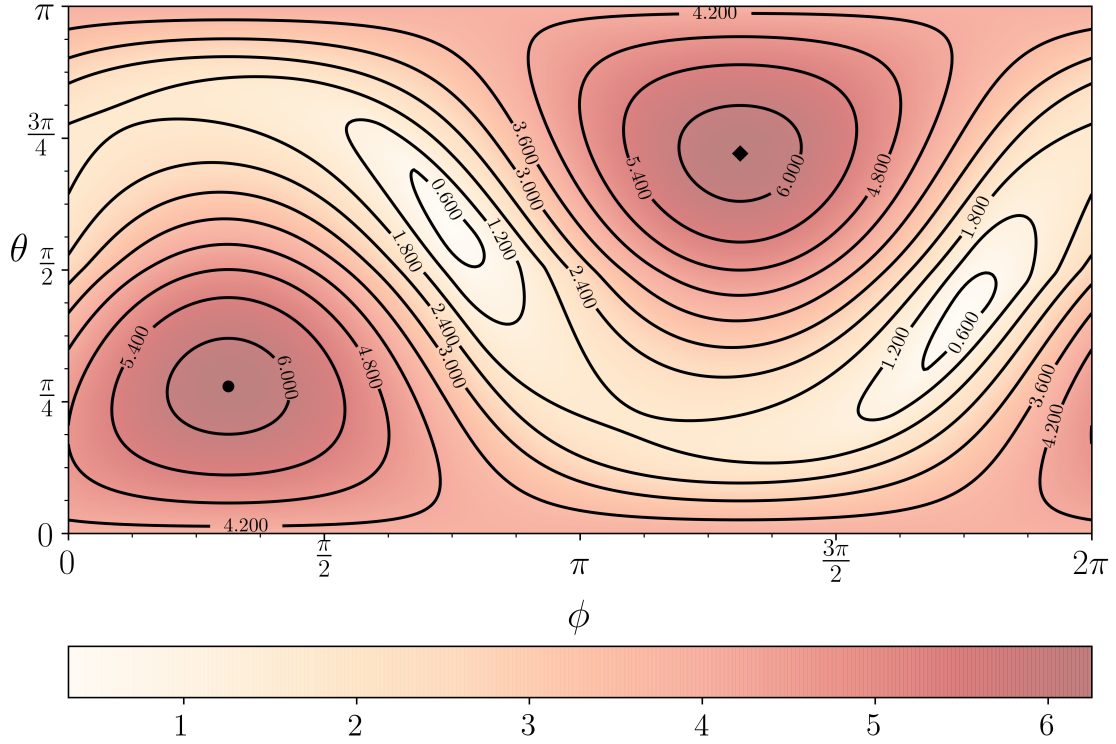
$$(\phi^{\max}, \theta^{\max}) \begin{cases} (\arctan(3/t), \arccos(6/\sqrt{t^4 + 9t^2 + 36})), \\ (\pi + \arctan(3/t), \arccos(-6/\sqrt{t^4 + 9t^2 + 36})). \end{cases} \quad (2.37)$$

Hence, the diameter of the triple integrator reach set at time t is equal to $(\mu t/3) \times \sqrt{t^4 + 9t^2 + 36}$.

Fig. 2.6 shows how the width of the integrator reach set for $d = 3$, $m = 1$ varies over $(\phi, \theta) \in [0, \pi] \times [0, 2\pi)$, which parameterize the unit sphere \mathbb{S}^2 . The location of the maximizers are given by (2.37), and are depicted in Fig. 2.6 via filled black circle and filled black square.

For a visualization of the width and diameter for the double integrator, see [78, Fig. 2].

We now turn to investigate how the volume and the diameter of the integrator reach set (2.2) scale with time and the state dimension. For clarity, we focus on the



- $(\arctan(3/t), \arccos(6/\sqrt{t^4 + 9t^2 + 36}))$
- ◆ $(\pi + \arctan(3/t), \arccos(-6/\sqrt{t^4 + 9t^2 + 36}))$

Figure 2.6: The width (2.33) for the single input triple integrator reach set $\mathcal{X}_{t=2}^\square(\{\mathbf{x}_0\})$ is shown as a function of $(\phi, \theta) \in [0, \pi] \times [0, 2\pi)$, which parameterize the unit sphere \mathbb{S}^2 . Here $\mathcal{U} = [-1, 1]$ and hence $\mu = 1$. The darker (resp. lighter) hues correspond to the higher (resp. lower) widths. The filled black circle and the filled black square correspond to the maximizers $(\phi^{\max}, \theta^{\max})$ given by (2.37).

single input case.

2.2.3 Scaling of the volume

Fig. 2.7 plots the volume (2.31) for the single input ($m = 1$) case against time t for varying state space dimension d . In this case, $\mathcal{U} = [\alpha, \beta]$, and therefore $\mu := (\beta - \alpha)/2$. As expected, the volume of the reach set increases with time for any fixed d .

Let us now focus on the scaling of the volume with respect to the state dimension d . For $m = 1$, using the known asymptotic [94] for $\prod_{k=1}^{d-1} (2k+1)!/k!$, we find the $d \rightarrow \infty$

asymptotic for the volume:

$$\text{vol}(\mathcal{X}_t^\square(\{\mathbf{x}_0\})) \sim (2\mu)^{d_t d(d+1)/2} \frac{\exp\left(\frac{3}{2}d^2 + \frac{1}{12}\right)}{c \times 2^{(2d^2 - \frac{1}{12})} d^{(d^2 + \frac{1}{12})}},$$

where $c \approx 1.2824\dots$ is the Glaisher-Kinkelin constant [95, Sec. 2.15].

Fig. 2.7 shows that when t is small, the volume of the larger dimensional reach set stays lower than its smaller dimensional counterpart. In particular, given two state space dimensions d, δ with $d > \delta$, and all other parameters kept fixed, there exists a critical time t_{cr} when the volume of the d dimensional reach set overtakes that of the δ dimensional reach set.

For any $d > \delta$, the critical time t_{cr} satisfies

$$\underbrace{\text{vol}(\mathcal{X}_{t_{\text{cr}}}^\square(\{\mathbf{x}_0\}) \subset \mathbb{R}^d)}_{d \text{ dimensional volume}} = \underbrace{\text{vol}(\mathcal{X}_{t_{\text{cr}}}^\square(\{\mathbf{x}_0\}) \subset \mathbb{R}^\delta)}_{\delta \text{ dimensional volume}},$$

which together with (2.31) yields

$$t_{\text{cr}} = (2\mu)^{-\frac{2}{d+\delta+1}} \left(\prod_{k=\delta}^{d-1} \frac{(2k+1)!}{k!} \right)^{\frac{2}{(d-\delta)(d+\delta+1)}}. \quad (2.38)$$

In particular, for $\delta = d - 1$, we get

$$t_{\text{cr}} = \left(\frac{1}{2\mu} \frac{(2d-1)!}{(d-1)!} \right)^{1/d}, \quad d = 2, 3, \dots \quad (2.39)$$

For instance, when $\mu = 1$, $d = 3$, $\delta = 2$, we have $t_{\text{cr}} = (30)^{1/3} \approx 3.1072$. When $\mu = 1$, $d = 4$, $\delta = 3$, we have $t_{\text{cr}} = (420)^{1/4} \approx 4.5270$. The dashed vertical lines in Fig. 2.7 show the critical times given by (2.39).

Applying Stirling's approximation $n! \sim \sqrt{2\pi n}(n/e)^n$, we obtain the $d \rightarrow \infty$ asymptotic for (2.39): $t_{\text{cr}} \sim \frac{4}{e} d \mu^{-\frac{1}{d}} 2^{-\frac{3}{2d}}$, where \sim denotes asymptotic equivalence [96, Ch. 1.4], and e is the Euler number.

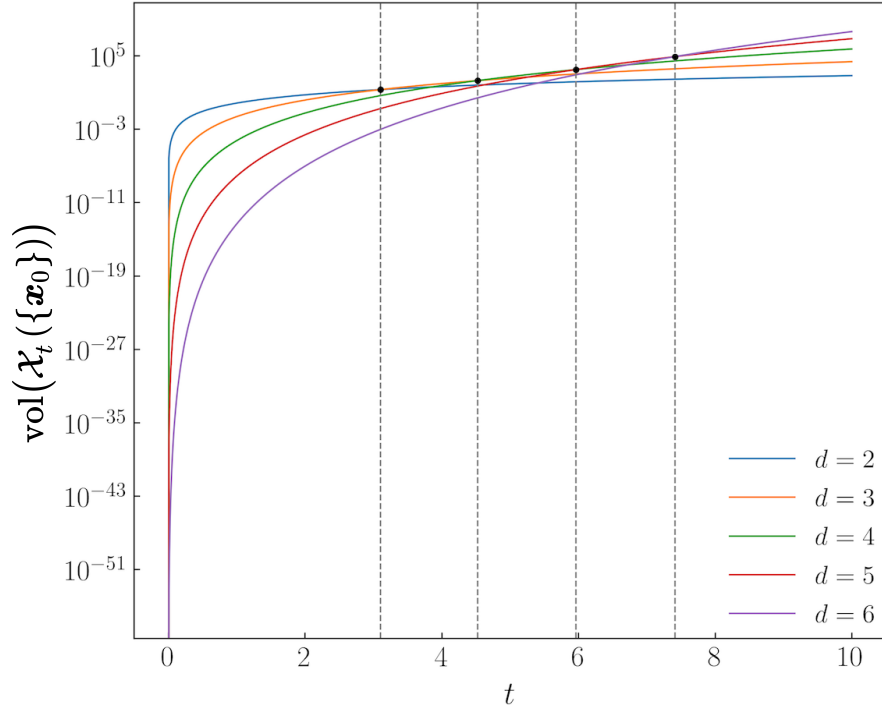


Figure 2.7: For single input ($m = 1$), the volume of the integrator reach set $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ computed from (2.31) is plotted against time t for state dimensions $d = 2, 3, \dots, 6$ with $\mathcal{U} = [\alpha, \beta] = [-1, 1]$, $\mu := (\beta - \alpha)/2 = 1$. The dashed vertical lines show the critical times given by (2.39).

2.2.4 Scaling of the diameter

Fig. 2.8 plots the diameter of (2.35) for the single input ($m = 1$) case against time t for varying state space dimension d . As earlier, $\mathcal{U} = [\alpha, \beta]$, $\mu := (\beta - \alpha)/2$. As expected, the diameter of the reach set increases with time for any fixed d .

As $d \rightarrow \infty$, the diameter approaches a limiting curve shown by the dotted line in Fig. 2.8. To derive this limiting curve, notice that for $m = 1$, the formula (2.35) gives

$$\lim_{d \rightarrow \infty} \text{diam}(\mathcal{X}_t^\square(\{\mathbf{x}_0\})) = \lim_{d \rightarrow \infty} 2\mu \sqrt{\sum_{j=1}^d \left(\frac{t^j}{j!}\right)^2}. \quad (2.40)$$

We write the partial sum

$$\sum_{j=1}^d \left(\frac{t^j}{j!}\right)^2 = \underbrace{\sum_{j=1}^{\infty} \left(\frac{t^j}{j!}\right)^2}_{=: S_1} - \underbrace{\sum_{j=d+1}^{\infty} \left(\frac{t^j}{j!}\right)^2}_{=: S_2}, \quad (2.41)$$

and by ratio test, note that both the sums S_1, S_2 converge. In particular, S_1 converges to $I_0(2t) - 1$, where $I_0(\cdot)$ is the zeroth order modified Bessel function of the first kind. This follows from the very definition of the ν th order modified Bessel function of the first kind, given by

$$I_\nu(z) := (z/2)^\nu \sum_{j=0}^{\infty} \frac{(z^2/4)^j}{j! \Gamma(\nu + j + 1)}, \quad \nu \in \mathbb{R},$$

where $\Gamma(\cdot)$ denotes the Gamma function.

On the other hand, using the definition of the generalized hypergeometric function⁷

$${}_1F_2(a_1; b_1, b_2; z) := \sum_{n=0}^{\infty} \frac{(a_1)_n}{(b_1)_n (b_2)_n} \frac{z^n}{n!},$$

we find that

$$S_2 = \frac{t^{2(d+1)} {}_1F_2(1; d+2, d+2; t^2)}{((d+1)!)^2}.$$

Therefore, (2.41) evaluates to

$$S_1 - S_2 = I_0(2t) - 1 - \frac{t^{2(d+1)} {}_1F_2(1; d+2, d+2; t^2)}{((d+1)!)^2}. \quad (2.42)$$

Combining (2.40), (2.41), (2.42), and using the continuity of the square root function on $[0, \infty)$, we deduce that

$$\lim_{d \rightarrow \infty} \text{diam}(\mathcal{X}_t^\square(\{\mathbf{x}_0\})) = 2\mu \sqrt{\lim_{d \rightarrow \infty} (S_1 - S_2)} = 2\mu \sqrt{I_0(2t) - 1}. \quad (2.43)$$

That $\lim_{d \rightarrow \infty} S_2$ exists and equals to zero, follows from (2.41) and the continuity of the square:

$$\lim_{d \rightarrow \infty} S_2 = \lim_{j \rightarrow \infty} \left(\frac{t^j}{j!} \right)^2 = \left(\lim_{j \rightarrow \infty} \frac{t^j}{j!} \right)^2 = 0.$$

To see the last equality, let $a_j := t^j/j!$. By the ratio test, $\limsup_{j \rightarrow \infty} |a_{j+1}/a_j| = \lim_{j \rightarrow \infty} t/j = 0 < 1$, hence $\{a_j\}$ is a Cauchy sequence and $\lim_{j \rightarrow \infty} a_j = 0$.

⁷Here, $(\cdot)_n$ denotes the Pochhammer symbol [97, p. 256] or rising factorial.

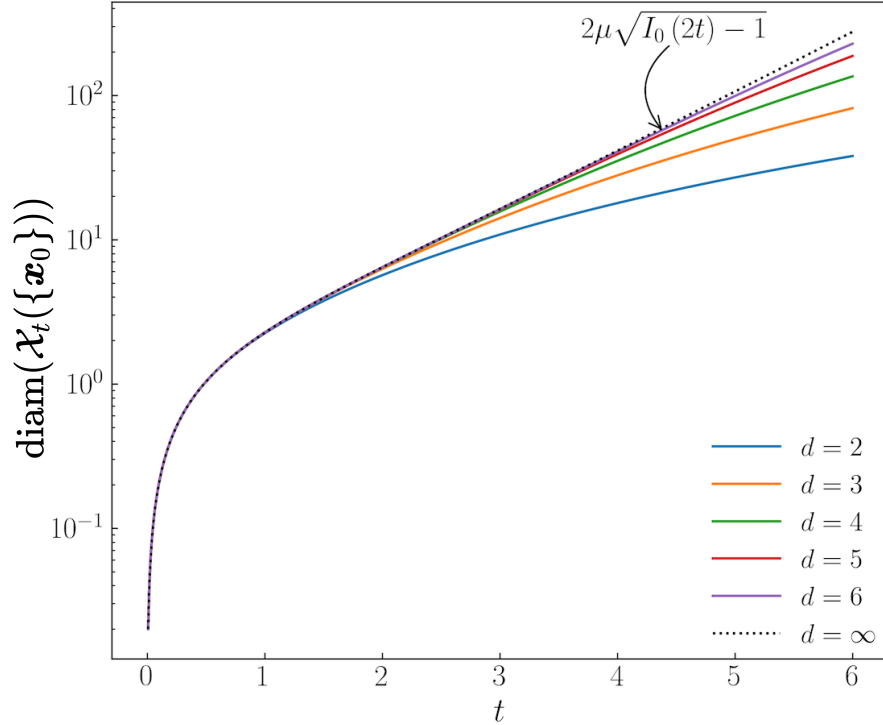


Figure 2.8: For single input ($m = 1$), the diameter of the integrator reach set $\mathcal{X}_t(\{x_0\})$ computed from (2.35) is plotted against time t for state dimensions $d = 2, 3, \dots, 6$ with $\mathcal{U} = [\alpha, \beta] = [-1, 1]$, $\mu := (\beta - \alpha)/2 = 1$. As $d \rightarrow \infty$, the diameter converges to $2\mu\sqrt{I_0(2t) - 1}$, shown by the dotted line.

The dotted line in Fig. 2.8 is the curve (2.43).

2.3 Benchmarking Over-approximations of Integrator Reach Sets

In practice, a standard approach for safety and performance verification is to compute “tight” over-approximation of the reach sets of the underlying controlled dynamical system. Several numerical toolboxes such as [35, 70] are available which over-approximate the reach sets using simple geometric shapes such as zonotopes and ellipsoids. Depending on the interpretation of the qualifier “tight”, different optimization problems ensue, e.g., minimum volume outer-approximation [98–105].

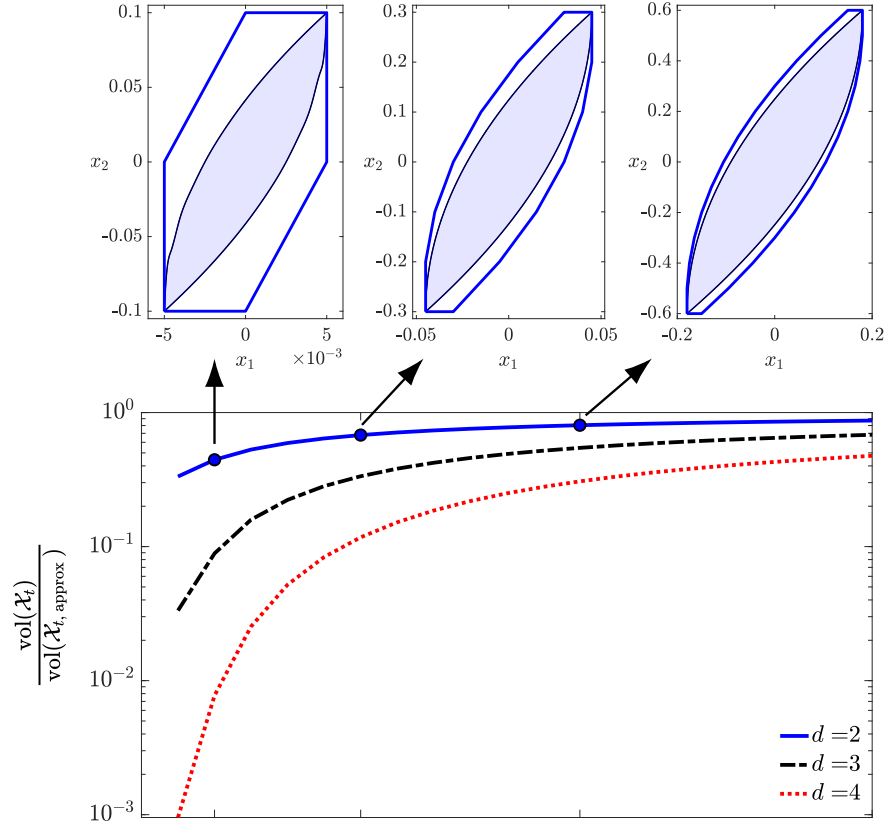


Figure 2.9: (Top) Zonotopic over-approximations of the double integrator reach sets; (bottom) the ratio of the volume of the single input integrator reach set \mathcal{X}_t and that of its zonotopic over-approximation $\mathcal{X}_{t,\text{approx}}$ for $d = 2, 3, 4$, plotted against time $t \in [0, 1]$. The results are computed using the CORA toolbox with $\mu = 1$, $\mathcal{X}_0 = \{\mathbf{0}\}$.

One potential application of our results in Ch. 2.2 is to help quantify the conservatism in different over-approximation algorithms by taking the integrator reach set as a benchmark case. For instance, Fig. 2.9 shows the conservatism in zonotopic over-approximations $\mathcal{X}_{t,\text{approx}}$ of the single input integrator reach sets $\mathcal{X}_t(\{\mathbf{x}_0\}) \subseteq \mathcal{X}_{t,\text{approx}}(\{\mathbf{0}\})$ for $d = 2, 3, 4$ with $0 \leq t \leq 1$ and $\mu = 1$, computed using the CORA toolbox [36, 70]. To quantify the conservatism, we used the volume formula (2.31) for computing the ratio of the volumes $\text{vol}(\mathcal{X}_t)/\text{vol}(\mathcal{X}_{t,\text{approx}}) \in [0, 1]$. The results shown in Fig. 2.9 were obtained by setting the zonotope order 50 in the CORA toolbox, which means that the number of zonotopic segments used by CORA for over-approximation

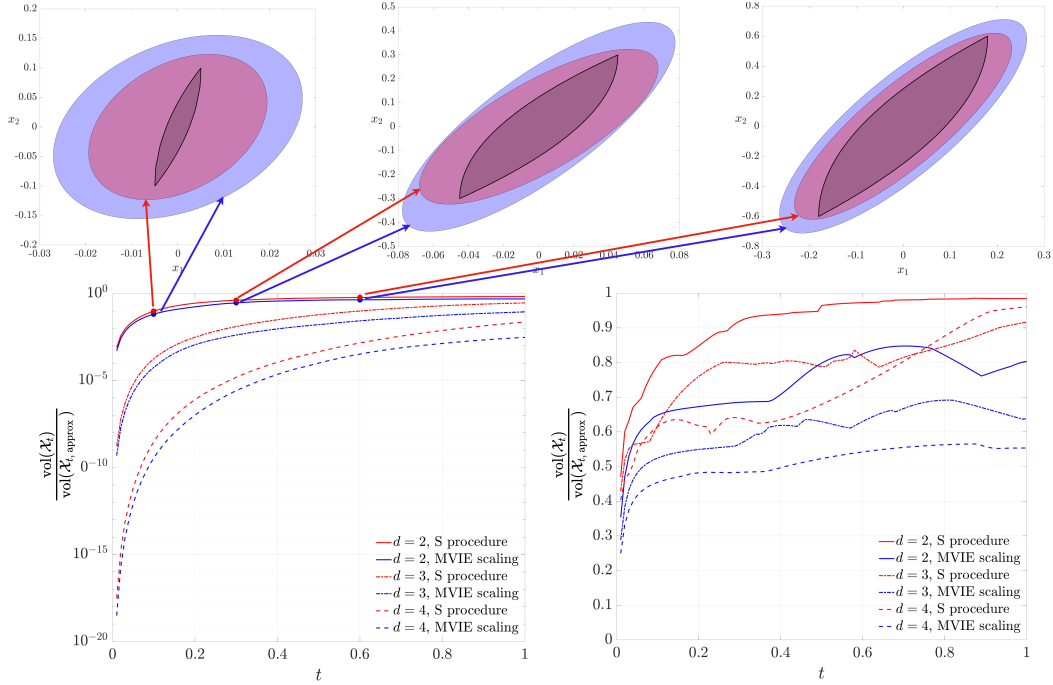


Figure 2.10: (Top) Ellipsoidal over-approximations of the double integrator reach sets; (bottom) the ratio of the volume (left) and diameter (right) of the single input integrator reach set \mathcal{X}_t and that of its ellipsoidal over-approximation $\mathcal{X}_{t,\text{approx}}$ for $d = 2, 3, 4$, plotted against time $t \in [0, 1]$. Two different ellipsoidal over-approximations are shown: one (in red) based on the S procedure, and the other (in blue) obtained by scaling the maximum volume inner ellipsoid (MVIE) of the intersection of a parameterized family of ellipsoids. The results are computed for $\mu = 1$, $\mathcal{X}_0 = \{\mathbf{0}\}$.

was $\leq 50d$. As expected, increasing the zonotope order improves the accuracy at the expense of computational speed, but among the different dimensional volume ratio curves, trends similar to Fig. 2.9 remain. It is possible [77, Thm. 1.1, 1.2] to compute the optimal zonotope order as function of the desired approximation accuracy (i.e., desired Hausdorff distance from the zonoid).

For the numerical results shown in Fig. 2.9, we found the diameters of the over-approximating zonotopes for $d = 2, 3, 4$, to be the same as that of the true diameters given by (2.35) for all times.

Fig. 2.10 depicts the conservatism in ellipsoidal over-approximations $\mathcal{X}_{t,\text{approx}}$ of the single input integrator reach sets $\mathcal{X}_t(\{\mathbf{0}\}) \subseteq \mathcal{X}_{t,\text{approx}}(\{\mathbf{0}\})$ for $d = 2, 3, 4$ with $0 \leq$

$t \leq 1$ and $\mu = 1$, following the algorithms in ellipsoidal toolbox [35]. Specifically, the reach set at time t , is over-approximated by the *intersection* of a carefully constructed parameterized family of ellipsoids $\mathcal{E}(\mathbf{q}(t), \mathbf{Q}_{\ell_i(t)}(t))$ defined as

$$\{\mathbf{x} \in \mathbb{R}^d \mid (\mathbf{x} - \mathbf{q}(t)) (\mathbf{Q}_{\ell_i(t)}(t))^{-1} (\mathbf{x} - \mathbf{q}(t))^\top \leq 1\},$$

for unit vectors $\ell_i(0) \in \mathbb{R}^d$, $i = 1, \dots, N$. The choice of $\ell_i(0)$ determines $\ell_i(t) := \exp(-\mathbf{A}^\top t) \ell_i(0)$, which in turn parameterizes the $d \times d$ symmetric positive definite shape matrix $\mathbf{Q}_{\ell_i(t)}(t)$; we refer the readers to [106, Ch. 3.2], [79, Ch. 3] where the corresponding evolution equations were derived using optimal control. The center vectors $\mathbf{q}(t) \in \mathbb{R}^d$, and the shape matrices $\mathbf{Q}_{\ell_i(t)}(t)$ for this parameterized family of ellipsoids are constructed such that $\cap_{i=1}^N \mathcal{E}(\mathbf{q}(t), \mathbf{Q}_{\ell_i(t)}(t))$ is guaranteed to be a superset of the reach set at time t for any finite N , and for $N \rightarrow \infty$, recovers the reach set at that time.

For the results shown in Fig. 2.10, we used $N = 20$ randomly chosen unit vectors $\ell_i(0) \in \mathbb{R}^d$. Ideally, one would like to compute the (unique) minimum volume outer ellipsoid (MVOE), a.k.a. the Löwner-John ellipsoid [107, 108] of the convex set $\cap_{i=1}^{20} \mathcal{E}(\mathbf{q}(t), \mathbf{Q}_{\ell_i(t)}(t))$, which is a semi-infinite programming problem [93, Ch. 8.4.1], and has no known *exact* semidefinite programming (SDP) reformulation. We computed two different relaxations of this problem: one based on the S procedure [109, Ch. 3.7.2], and the other by homothetic scaling of the maximum volume inner ellipsoid (MVIE) [107, Thm. III] of the set $\cap_{i=1}^{20} \mathcal{E}(\mathbf{q}(t), \mathbf{Q}_{\ell_i(t)}(t))$. Both of these lead to solving SDP problems, and both are guaranteed to contain the Löwner-John ellipsoid of the intersection of the parameterized family of ellipsoids. These suboptimal (w.r.t. the MVOE criterion) solutions, computed using `cvx` [110], are shown in Fig. 2.10.

Fig. 2.10 shows that the S procedure entail less conservatism compared to the MVIE scaling, in terms of volume. While the volume ratio trends in Fig. 2.10 are

similar to that observed in Fig. 2.9, the approximation quality are lower. In light of the results in Ch. 2.1.1, this is not surprising: the integrator reach sets being zonoids (i.e., Hausdorff limit of zonotopes), the zonotopic outer-approximations are expected to perform better than other over-approximating shape primitives.

The main point here is that our results in Ch. 2.2 provide the ground truth for the size of the integrator reach set, thereby help benchmarking the performance of reach set approximation algorithms.

In the following section we discuss the geometry of integrator reachset when it is subjected to time-varying compact set-valued uncertainties in input.

3 | Integrator Reach Sets with Time Varying Set-Valued Uncertainties

In this chapter, we generalize our results from Chap. 2 on the exact geometry of the integrator reach sets for the case that the input $\mathbf{u}(t) \in \mathcal{U}(t)$, where $\mathcal{U}(t)$ is a compact set-valued trajectory. In contrast, the results in Ch. 2 assumed *time-invariant* compact set-valued uncertainties, i.e., $\mathbf{u}(t) \in \mathcal{U}$ where the compact set \mathcal{U} is constant w.r.t. time t .

Beyond intrinsic interests, these generalizations are motivated by applications involving differentially flat nonlinear systems, as we elaborate in the ensuing Sections 4.2 and 4.3. In those Sections, we will highlight how computing the reach sets of differentially flat systems and checking their intersections reduce to analogous questions in the Brunovsky normal coordinates with *time-varying* set-valued uncertainties even when the original inputs have *time-invariant* set-valued uncertainties. Resolving these questions, are made possible by our basic geometric results presented next.

3.1 Auxiliary Definitions

As in Ch. 2, we consider d dimensional integrator dynamics with m inputs and a relative degree vector $\mathbf{r} = (r_1, \dots, r_m)^\top \in \mathbb{N}^m$ (i.e., an $m \times 1$ vector of natural numbers) where $r_1 + \dots + r_m = d$. We let $\llbracket m \rrbracket := \{1, 2, \dots, m\}$.

The integrator reach set $\mathcal{X}_t \subset \mathbb{R}^d$ at time t starting from an initial set $\mathcal{X}_0 \subset \mathbb{R}^d$, subject to the integrator dynamics (1.6), is

$$\begin{aligned} \mathcal{X}_t^\square(\{\mathcal{X}_0\}) := \{ \mathbf{x} \in \mathbb{R}^d \mid \dot{\mathbf{x}} = \text{blkdiag}(\mathbf{A}_1, \dots, \mathbf{A}_m) \mathbf{x} + \\ \text{blkdiag}(\mathbf{b}_1, \dots, \mathbf{b}_m) \mathbf{u}, \mathbf{x}(0) \in \mathcal{X}_0, \mathbf{u}(s) \in \mathcal{U}(s) \subset \mathbb{R}^m, \text{ for all } s \in [0, t] \}, \end{aligned} \tag{3.1}$$

Where the input set

$$\mathcal{U}(s) := [\alpha_1(s), \beta_1(s)] \times [\alpha_2(s), \beta_2(s)] \times \cdots \times [\alpha_m(s), \beta_m(s)] \subset \mathbb{R}^m. \quad (3.2)$$

As is well known, the reach set (3.1) is compact convex provided \mathcal{X}_0 is compact convex (irrespective of whether the input sets $\mathcal{U}(\cdot)$ are convex or not). In this dissertation, we will need to discuss reach sets in different coordinates, and we will sometimes contract the integrator reach set and the original reach set notations in (3.1) and (1.2) as \mathcal{X}_t and \mathcal{Z}_t , respectively to help avoid notational clutter. Following a similar logic as in 2.8, we have $\mathcal{X}_t = \mathcal{X}_{1t} \times \mathcal{X}_{2t} \times \cdots \times \mathcal{X}_{mt} \equiv \mathcal{X}_{1t} \dot{+} \mathcal{X}_{2t} \dot{+} \cdots \dot{+} \mathcal{X}_{mt}$.

For any $s \in [0, t]$, and $j \in \llbracket m \rrbracket$, define the vector-valued extremum trajectories

$$\alpha_j(s) := \min_{\mathbf{u}(s) \in \mathcal{U}(s)} u_j(s), \quad \beta_j(s) := \max_{\mathbf{u}(s) \in \mathcal{U}(s)} u_j(s), \quad (3.3)$$

which are well-defined since $\mathcal{U}(\cdot)$ is compact at all times. Also, for $j \in \llbracket m \rrbracket$, let

$$\mu_j(s) := (\beta_j(s) - \alpha_j(s))/2, \quad \nu_j(s) := (\beta_j(s) + \alpha_j(s))/2. \quad (3.4)$$

Now we have

$$\boldsymbol{\xi}(s) := \begin{pmatrix} \mu_1(s) \boldsymbol{\xi}_1(s) \\ \vdots \\ \mu_m(s) \boldsymbol{\xi}_m(s) \end{pmatrix}, \quad \boldsymbol{\xi}_j(s) := \begin{pmatrix} s^{r_j-1}/(r_j-1)! \\ s^{r_j-2}/(r_j-2)! \\ \vdots \\ s \\ 1 \end{pmatrix}. \quad (3.5)$$

Notice that the above definitions resemble the ones defined in (2.3)-(2.5), except that the variables $\boldsymbol{\mu}(\cdot)$ and $\boldsymbol{\nu}(\cdot)$ in (3.4) are now trajectories instead of having constant values over time.

Now we can rewrite (3.1) as

$$\mathcal{X}_t^\square(\{\mathbf{x}_0\}) = \prod_{j=1}^m \left\{ \exp(t\mathbf{A}_j)\mathcal{X}_0 + \int_0^t \boldsymbol{\xi}_j(t-s) [\alpha_j(s), \beta_j(s)] ds \right\}. \quad (3.6)$$

3.1.1 Support function

In the following theorem, we present the support function of the integrator reach set under time-varying boxed valued input set (3.2)

Proposition 3.1. *For compact convex $\mathcal{X}_0 \subset \mathbb{R}^d$, and compact boxed-valued $\mathcal{U}(t) \subset \mathbb{R}^m$ as in (3.2), the support function of the reach set (3.1) is*

$$h_{\mathcal{X}_t^\square(\{\mathbf{x}_0\})} = \sum_{j=1}^m \left\{ h_{\mathcal{X}_{j0}}(\exp(t\mathbf{A}_j^\top) \mathbf{y}) + \int_0^t [\nu_j(s) \langle \mathbf{y}_j, \boldsymbol{\xi}(t-s) \rangle + \mu_j(s) |\langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle|] ds \right\}. \quad (3.7)$$

The proof of Proposition 3.1 follows a similar logic as the proof of Theorem 2.1 given in Appendix B.1.

3.2 Size of $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ with Time Varying Set-Valued Inputs

Next we discuss how to quantify the size of the reach set (3.1). Specifically, we show how the volume and the diameter of $\mathcal{X}(\mathbf{x}_0, t)$ can be computed numerically using the generalized parametric boundary formula (3.9). We will also provide an example to demonstrate our numerical computation.

3.2.1 Boundary of \mathcal{X}_t^\square with time varying set-valued inputs

The parametric equations describing the bounding surfaces of the integrator reachset with boxed-valued uncertainty as in (3.2), $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$, is presented in Proposition 3.2.

Proposition 3.2. *Consider the integrator reach set (3.6) with time-varying range input set (3.2) with relative degree vector $\mathbf{r} = (r_1, \dots, r_m)^\top$, and fixed $\mathbf{x}_0 \in \mathbb{R}^d$ comprising of subvectors $\mathbf{x}_{j0} \in \mathbb{R}^{r_j}$ where $j \in \llbracket m \rrbracket$.*

For $k, \ell \in \mathbb{N}$, let the indicator function $\mathbf{1}_{k \leq \ell} := 1$ for $k \leq \ell$, and $:= 0$ otherwise. Define a function $\chi_j(t, \mathbf{x}_0) : \mathbb{R}_{>0} \times \mathbb{R}^{r_j} \mapsto \mathbb{R}^{r_j}$, component-wise given by

$$\chi_{j,\ell} := \sum_{k=1}^{r_j} \mathbf{1}_{k \leq \ell} \frac{t^{\ell-k}}{(\ell-k)!} x_{j0,\ell}, \quad \text{for all } \ell \in \llbracket r_j \rrbracket, \quad (3.8)$$

where $x_{j0,\ell}$ denotes the ℓ -th component of the initial state \mathbf{x}_{j0} . Then, $\mathbf{x}_j^{\text{bdy}} \in \partial \mathcal{X}_{jt}(\{\mathbf{x}_{j0}\})$ admits $\boldsymbol{\sigma}$ -parameterization:

$$\begin{aligned} \mathbf{x}_j^{\text{bdy}}(\boldsymbol{\sigma}_j) &= \chi(t, \mathbf{x}_{j0}) + \int_0^t \nu_j(s) \boldsymbol{\xi}_j(t-s) ds \pm \int_0^{\sigma_1} \mu_j(s) \boldsymbol{\xi}_j(t-s) ds \\ &\mp \int_{\sigma_1}^{\sigma_2} \mu_j(s) \boldsymbol{\xi}_j(t-s) ds \pm \dots \pm (-1)^n \int_{\sigma_{n-1}}^t \mu_j(s) \boldsymbol{\xi}(t-s) ds. \end{aligned} \quad (3.9)$$

The parameter vector $\boldsymbol{\sigma}_j = (\sigma_1, \sigma_2, \dots, \sigma_{r_j-1}) \in \mathcal{W}_{jt} \subset \mathbb{R}^{r_j-1}$ where \mathcal{W}_{jt} is the Weyl chamber defined in (2.15), and the time varying trajectories $\boldsymbol{\mu}_j$, $\boldsymbol{\nu}_j$ and $\boldsymbol{\xi}_j$ are given in (3.4) and (3.5) respectively.

The proof of proposition 3.2 can be easily deduced from the proof of Theorem 2.3 given in Appendix B.3.

Without loss of generality, for better readability, in the rest of this section we focus on a single input integrator reach set and simplify the notations by dropping the subscript "j". The following interesting remarks are the direct consequence of Proposition 3.2.

Remark 3.1. *Each single input ($m = 1$) integrator reach set $\mathcal{X}_t(\mathbf{x}_0) \subset \mathbb{R}^d$ has two*

bounding surfaces $\partial\mathcal{X}^{\text{Upper}}$ and $\partial\mathcal{X}^{\text{lower}}$, with parametric representation

$$\partial\mathcal{X}(\boldsymbol{\sigma}) := \partial\mathcal{X}^{\text{upper}}(\boldsymbol{\sigma}) \cup \partial\mathcal{X}^{\text{lower}}(\boldsymbol{\sigma}),$$

where $\boldsymbol{\sigma} \in \mathcal{W}_t \subset \mathbb{R}^{d-1}$. The bounding hyper-surfaces $\partial\mathcal{X}^{\text{upper}}(\boldsymbol{\sigma})$ and $\partial\mathcal{X}^{\text{lower}}(\boldsymbol{\sigma})$ can be obtained by taking the alternating sum in (3.9). For instance, if the third term of the sum in RHS of (3.9) has plus rather than minus and alternating signs thereafter, we obtain the $\partial\mathcal{X}^{\text{upper}}(\boldsymbol{\sigma})$, and vice-versa.

Remark 3.2. From (3.9), we notice that the equations of the boundary still only depend on the extremum curves $\alpha(s)$ and $\beta(s)$ for $0 \leq s \leq t$. However, unlike the case of time independent input set \mathcal{U} , the integrator reach set \mathcal{X}_t^\square is no longer in general semialgebraic unless these extremum curves are polynomial at all times. Notice that the reach set (3.6) is still a zonoid.

Remark 3.3. For each single input ($m = 1$) integrator reach set $\mathcal{X}_t \subset \mathbb{R}^d$, we say that a subset of $\mathcal{W}_t \subset \mathbb{R}^{d-1}$ is “degenerate” if its $d - 1$ dimensional measure equals to zero. Then, for $d > 2$, any parameter vector from either of the degenerate sets

$$\mathcal{W}_t^0 := \{\boldsymbol{\sigma} \in \mathcal{W}_t \mid \sigma_1 = 0\}, \quad (3.10a)$$

$$\mathcal{W}_t^t := \{\boldsymbol{\sigma} \in \mathcal{W}_t \mid \sigma_{d-1} = t\}, \quad (3.10b)$$

transforms (3.9) in dimension d into (3.9) of dimension $d - 1$. Another degeneracy occurs for parameter sets of the form

$$\begin{aligned} \mathcal{W}_t^k := \{\boldsymbol{\sigma} \in \mathcal{W}_t \mid \sigma_{i_0} = \sigma_{i_1} = \dots = \sigma_{i_k}, \quad k \in \llbracket d - 2 \rrbracket \\ i_0 < i_1 < \dots < i_k, \quad \text{and } i_0, i_1, \dots, i_k \in \llbracket d - 1 \rrbracket\}, \end{aligned} \quad (3.11)$$

for which (3.9) in dimension d will resemble (3.9) in dimension $d - 2k + 2$. The image of this degenerate sets has zero measure and does not contribute to the area of $\partial\mathcal{X}_t$. Let us call the union of degenerate sets as $\mathcal{W}_t^{\text{dgen}} = \mathcal{W}_t^0 \cup \mathcal{W}_t^1 \cup \mathcal{W}_t^2 \cup \dots \cup \mathcal{W}_t^{d-2} \cup \mathcal{W}_t^t$.

Remark 3.4. *The bounding hypersurfaces of a single input integrator reach set ($m=1$) $\partial\mathcal{X}^{\text{upper}}$ and $\partial\mathcal{X}^{\text{lower}}$ touch in the regions of parameter space created by (3.10). Specifically, $\partial\mathcal{X}^{\text{upper}}$ with $\sigma_1 = 0$ sweeps a hypersurface identical to $\partial\mathcal{X}^{\text{lower}}$ with $\sigma_{n-1} = t$, and vice versa. For instance, in 3 dimensional case with zero initial conditions, when $\sigma_1 = 0$ in the upper surface and $\sigma_2 = t$ in the lower surface, we get*

$$\begin{aligned}\mathbf{x}^{\text{upper}}(\boldsymbol{\sigma}) &= \int_0^t \nu(\tau)\xi(t-\tau)d\tau - \int_0^{\sigma_2} \mu(\tau)\xi(t-\tau)d\tau + \int_{\sigma_2}^T \mu(\tau)\xi(t-\tau)d\sigma, \\ \mathbf{x}^{\text{lower}}(\boldsymbol{\sigma}_j) &= \int_0^t \nu(\tau)\xi(t-\tau)d\tau - \int_0^{\sigma_1} \mu(\tau)\xi(t-\tau)d\tau + \int_{\sigma_1}^T \mu(\tau)\xi(t-\tau)d\tau,\end{aligned}$$

both of which represent the same curve for $0 \leq \sigma_1 \leq \sigma_2 \leq t$.

3.2.2 Volume of $\mathcal{X}_t^\square(\{x_0\})$ with time varying set-valued inputs

As discussed in earlier sections, having a computational handle on volume is helpful in providing ground truth to quantify the conservatism of numerical algorithms which over-approximate the reach set via simpler geometric shapes such as variants of ellipsoids [102, 104, 105, 111, 112] or variants of zonotopes [55, 58, 59, 62, 70, 113, 114].

Again, first consider the single input case ($m=1$). We start by noticing that each $\boldsymbol{\sigma} \in \mathcal{W}_t$ assigns a pair of points $(\mathbf{x}^{\text{upper}}, \mathbf{x}^{\text{lower}})$ on $\partial\mathcal{X}_t$, one on each bounding hypersurfaces mentioned in Remark 3.1, i.e., $\mathbf{x}^{\text{upper}}(\boldsymbol{\sigma}) \in \partial\mathcal{X}^{\text{upper}}(\boldsymbol{\sigma})$ and $\mathbf{x}^{\text{lower}}(\boldsymbol{\sigma}) \in \partial\mathcal{X}^{\text{lower}}(\boldsymbol{\sigma})$.

Since \mathcal{X}_t is convex, each $\mathbf{x} \in \mathcal{X}_t$ can be written as a convex combination of two points in \mathcal{X}_t . In the following, we show a stronger result: any point in \mathcal{X}_t can be written as a convex combination of a pair $(\mathbf{x}^{\text{upper}}, \mathbf{x}^{\text{lower}})$ evaluated at the same parameter $\boldsymbol{\sigma} \in \mathcal{W}_t$. This result will find use in volume computation (Thm. 3.2).

Theorem 3.1. *For any $\mathbf{x} \in \mathcal{X}_t$, there exists $(\boldsymbol{\sigma}, \lambda) \in \mathcal{W}_t \times [0, 1]$ such that*

$$\mathbf{x} = \boldsymbol{\pi}(\boldsymbol{\sigma}, \lambda) := \lambda\mathbf{x}^{\text{upper}}(\boldsymbol{\sigma}) + (1 - \lambda)\mathbf{x}^{\text{lower}}(\boldsymbol{\sigma}), \quad (3.12)$$

i.e., the parametric map $\boldsymbol{\pi} : \mathcal{W}_t \times [0, 1] \rightarrow \mathcal{X}_t$ is surjective.

The proof of theorem 3.1 is given in Appendix C.1.

Remark 3.5. We clarify here that (3.9) gives a parameterization of $\partial\mathcal{X}_t$, while (3.12) gives a parameterization of \mathcal{X}_t .

Theorem 3.2. The n dimensional Lebesgue volume of the Single input integrator reach set ($m = 1$) (3.6) at time t , is

$$\text{vol}_n(\mathcal{X}_t(\{\mathbf{x}_0\})) = \int_0^1 \int_{\mathcal{W}_t} |\det(\mathbf{D}\boldsymbol{\pi})| d\boldsymbol{\sigma} d\lambda \quad (3.13)$$

where $\mathbf{D}\boldsymbol{\pi}$ denotes the Jacobian of $\boldsymbol{\pi}$ in (3.12), and $\boldsymbol{\sigma} \in \mathcal{W}_t$ as in (2.15). In particular,

$$\left| \det(\mathbf{D}\boldsymbol{\pi}) \right| = \mu(\sigma_1)\mu(\sigma_2)\cdots\mu(\sigma_{d-1})|(4\lambda - 2)^{d-1}| \times \left| \det \begin{pmatrix} \frac{(t - \sigma_1)^{d-1}}{(d-1)!} & \cdots & \frac{(t - \sigma_{d-1})^{d-1}}{(d-1)!} & \zeta_1(\boldsymbol{\sigma}) \\ \vdots & \vdots & \vdots & \vdots \\ (t - \sigma_1) & \cdots & (t - \sigma_{d-1}) & \zeta_{d-1}(\boldsymbol{\sigma}) \\ 1 & \cdots & 1 & \zeta_d(\boldsymbol{\sigma}) \end{pmatrix} \right|, \quad (3.14)$$

where $\zeta(\boldsymbol{\sigma}) := \mathbf{x}^{\text{upper}}(\boldsymbol{\sigma}) - \mathbf{x}^{\text{lower}}(\boldsymbol{\sigma})$.

The volume of the multi-input integrator reach set, denoted as \mathcal{X}_t^\square , is determined by the product of the volumes of its individual constituent single-input integrator reach sets, i.e., $\text{vol}(\mathcal{X}_t^\square) = \prod_{j=1}^m \text{vol}(\mathcal{X}_{jt})$.

The proof of theorem 3.2 is given in Appendix C.2. In the upcoming chapter, we will explore and showcase practical applications stemming from our results.

3.3 Recovering $\partial\mathcal{X}_t^\square$ for Time Invariant \mathcal{U}

In this section, we recover the parametric boundary of the integrator reach set \mathcal{X}_t^\square for the time invariant input set given in (2.16), by specializing the parametric boundary formula for the general time varying boxed valued input uncertainty derived in (3.9).

Consider the support function expression in (3.7). For the time invariant input set (2.1), we can rewrite the integral in the RHS of the support function expression (3.7) as

$$\int_0^t \left[\nu_j(s) \langle \mathbf{y}_j, \boldsymbol{\xi}(t-s) \rangle + \mu_j(s) |\langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle| \right] ds \quad (3.15)$$

$$= \nu_j \int_0^t \langle \mathbf{y}_j, \boldsymbol{\xi}_j(s) \rangle ds + \mu_j \int_0^t |\langle \mathbf{y}_j, \boldsymbol{\xi}_j(s) \rangle| ds. \quad (3.16)$$

correspondingly, we can rewrite the parametric boundary (3.9) as

$$\begin{aligned} \mathbf{x}_j^{\text{bdy}}(\boldsymbol{\sigma}_j) &= \boldsymbol{\chi}(t, \mathbf{x}_{j0}) + \nu_j \int_0^t \boldsymbol{\xi}_j(s) ds \pm \mu_j \int_0^{\sigma_1} \boldsymbol{\xi}_j(s) ds \\ &\mp \mu_j \int_{\sigma_1}^{\sigma_2} \boldsymbol{\xi}_j(s) ds \pm \dots \pm (-1)^n \mu_j \int_{\sigma_{n-1}}^t \boldsymbol{\xi}(s) ds. \end{aligned} \quad (3.17)$$

Notice that for the time invariant input set, the extremum trajectories α_j, β_j , are constant at all times for $j \in \llbracket m \rrbracket$. Therefore, we can factor ν_j and μ_j out of the integral in the RHS of (3.17). Form (1.33) we have

$$\xi_{j,k}(s) = \frac{s^{r_j-k}}{(r_j-k)!}, \quad \text{for } k \in \llbracket r_j \rrbracket, \text{ and } j \in \llbracket m \rrbracket.$$

Let $x_{j,k}^{\text{bdy}}$ denote the k component of $\mathbf{x}_j^{\text{bdy}}$. We simplify each term in the write hand side of $x_{j,k}^{\text{bdy}}$ given in (3.17) as follows.

$$+ \nu_j \int_0^t \xi_{j,k}(s) ds = + \frac{\nu_j t^{r_j-k+1}}{(r_j-k+1)!}, \quad (3.18a)$$

$$\pm \mu_j \int_0^{\sigma_1} \xi_{j,k}(s) ds = \pm \frac{\mu_j \sigma_1^{r_j-k+1}}{(r_j-k+1)!}, \quad (3.18b)$$

$$\pm \mu_j \int_{\sigma_{i_1}}^{\sigma_{i_2}} \xi_j(s) ds = \pm \frac{\mu_j}{(r_j-k+1)!} \left\{ \sigma_{i_2}^{r_j-k+1} - \sigma_{i_1}^{r_j-k+1} \right\}, \quad \text{for } 1 < i_1 < i_2 < r_j - 1, \quad (3.18c)$$

$$\pm \mu_j \int_{\sigma_{r_j-1}}^t \xi_j(s) ds = \pm \frac{\mu_j}{(r_j-k+1)!} \left\{ t^{r_j-k+1} - \sigma_{r_j-1}^{r_j-k+1} \right\}. \quad (3.18d)$$

By substituting equations (3.18) into (3.17), and also using (3.8), we recover the para-

metric boundary formula for the time invariant input set given in (2.16)

$$\begin{aligned}
 x_{j,k}^{\text{bdy}}(\boldsymbol{\sigma}) &= \sum_{\ell=1}^{r_j} \mathbf{1}_{k \leq \ell} \frac{t^{\ell-k}}{(\ell-k)!} \mathbf{x}_{j0,\ell} + \frac{\nu_j t^{r_j-k+1}}{(r_j-k+1)!} \\
 &\quad \pm \frac{\mu_j}{(r_j-k+1)!} \left\{ (-1)^{r_j-1} t^{r_j-k+1} + 2 \sum_{q=1}^{r_j-1} (-1)^{q+1} \sigma_q^{r_j-k+1} \right\}.
 \end{aligned}$$

4 | Applications

In this chapter, we leverage our results to precisely determine the reach set and the volume of any controllable single-input LTI system. Subsequently, we demonstrate the application of our results in detecting intersections among the reach sets of feedback linearizable systems, and introduce a systematic procedure for learning the reach sets of the same.

4.1 Exact Computation of LTI Reach Set

Motivated by safety and performance verification of controlled dynamical systems, a vast body of works in systems-control literature have studied the problem of computing or approximating the reach sets of linear control systems [1, 3, 55, 115–118] in particular. See [6] for a recent survey on this broad topic.

In this section, we propose a new semi-analytical method to compute the reach set of a controllable single input LTI system with known bounds on its input range.

We show (Ch. 4.1.1) how certain generalizations of our results from Chapters 3 and 2 (see also [78, 119, 120]) on the exact geometry (e.g., boundary, volume) of integrator reach sets can enable computing the controllable LTI reach sets with bounded input. Our results reveal that the controllable canonical form serves as a bridge to transfer such geometric results from the integrator to the original state coordinates.

4.1.1 Brunovsky normal form and LTI reach set

Consider a *controllable* single input LTI system

$$\dot{z} = \tilde{A}z + \tilde{b}v, \quad \tilde{A} \in \mathbb{R}^{d \times d}, \quad \tilde{b} \in \mathbb{R}^d, \quad v_{\min} \leq v(\cdot) \leq v_{\max} \quad (4.1)$$

with state $\mathbf{z} \in \mathbb{R}^d$ and input $v \in \mathbb{R}$. For continuous bounded v , i.e., $v(\cdot) \in C([0, t])$ and $v_{\min} \leq v(\cdot) \leq v_{\max}$, define the *LTI reach set*

$$\mathcal{Z}_t(\{\mathbf{z}_0\}) := \{\mathbf{z}(t) \in \mathbb{R}^d \mid (4.1), \quad \mathbf{v}(s) \in [v_{\min}, v_{\max}] \forall s \in [0, t]\}. \quad (4.2)$$

Thanks to the Lyapunov convexity theorem [48, 49, 121, 122], the reach set (4.2) is guaranteed to be compact, convex [71, Prop. 6.1], [3], and in particular, *zonoids* (Chapter 2).

In this work, we consider the case that v_{\min}, v_{\max} are constants, i.e., the LTI input in (4.1) has time-invariant bounds.

Let \mathbf{q}^\top denote the last row of the inverse of its controllability matrix, and define the nonsingular matrix

$$\mathbf{M} := \left(\mathbf{q} \quad \tilde{\mathbf{A}}^\top \mathbf{q} \quad \dots \quad (\tilde{\mathbf{A}}^\top)^{d-1} \mathbf{q} \right)^\top. \quad (4.3)$$

As is well-known [123, Sec. 6], the invertible linear map $\mathbf{z} \mapsto \mathbf{x} := \mathbf{M}\mathbf{z}$ transforms (4.1) into a *controllable canonical form*

$$\dot{\mathbf{x}} = \mathbf{A}_{\text{con}} \mathbf{x} + \mathbf{b}v, \quad (4.4)$$

where

$$\mathbf{A}_{\text{con}} := \mathbf{M} \tilde{\mathbf{A}} \mathbf{M}^{-1} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -c_0 & -c_1 & -c_2 & \dots & -c_{n-1} \end{pmatrix}, \quad \mathbf{b} := \mathbf{M} \tilde{\mathbf{b}} = (0 \ 0 \ \dots \ 0 \ 1)^\top. \quad (4.5)$$

In particular, \mathbf{A}_{con} is a *companion matrix* (see e.g., [124, p. 195]) whose last row is in terms of the (real) coefficients of the monic characteristic polynomial of $\tilde{\mathbf{A}}$, given by

$$p(\lambda) := \lambda^d + c_{d-1} \lambda^{d-1} + \dots + c_1 \lambda + c_0. \quad (4.6)$$

Letting $\mathbf{c} := (c_0, c_1, \dots, c_{d-1})^\top$, we can succinctly write \mathbf{A}_{con} in (4.5) as

$$\mathbf{A}_{\text{con}} = \begin{bmatrix} \mathbf{0}_{(d-1) \times 1} & \mathbf{I}_{d-1} \\ & -\mathbf{c}^\top \end{bmatrix}. \quad (4.7)$$

Defining a new input

$$u := -\langle \mathbf{c}, \mathbf{x} \rangle + v \quad (4.8)$$

further transforms (4.4) into the Brunovsky normal a.k.a. the n th order integrator form with state \mathbf{x} and input u , given by (1.6).

4.1.2 Semi-analytical algorithm

We propose to use the input correspondence (4.8) for explicitly computing (4.2) in two steps:

step 1: explicitly compute the (boundary of the compact) integrator reach set \mathcal{X}_t for to-be-determined input range $[\alpha(s), \beta(s)] \forall s \in [0, t]$.

step 2: compute $\mathcal{Z}_t(\{z_0\}) = \mathbf{M}^{-1} \mathcal{X}_t(\{\mathbf{M}z_0\})$.

For **step 1**, notice from (4.8) that even when the original control input range is time invariant, i.e., v_{\min}, v_{\max} are constants, still the transformed input u has a time-varying range $[\alpha(s), \beta(s)] \forall s \in [0, t]$. This is because the input correspondence (4.8) occurs via time-varying \mathbf{x} . Notice that $\forall s \in [0, t]$, the *linear* state dependence in (4.8) allows us to write

$$\alpha(s) = -\langle \mathbf{c}, e^{s\mathbf{A}_{\text{con}}} \mathbf{M}z_0 \rangle + I_{\min}(s), \quad (4.9a)$$

$$\beta(s) = -\langle \mathbf{c}, e^{s\mathbf{A}_{\text{con}}} \mathbf{M}z_0 \rangle + I_{\max}(s), \quad (4.9b)$$

where

$$I_{\min}(s) := \inf_{v(\cdot) \in C([0,s])} I(v), \quad \text{subject to } v_{\min} \leq v(\cdot) \leq v_{\max}, \quad (4.10a)$$

$$I_{\max}(s) := \sup_{v(\cdot) \in C([0,s])} I(v), \quad \text{subject to } v_{\min} \leq v(\cdot) \leq v_{\max}, \quad (4.10b)$$

and

$$I(v) := v(s) - \int_0^s f(\tau)v(\tau)d\tau, \quad f(\tau) := \langle \mathbf{c}, e^{(s-\tau)\mathbf{A}_{\text{con}}} \mathbf{b} \rangle. \quad (4.11)$$

In Ch. 4.1.3, we semi-analytically determine $I_{\min}(\cdot)$, $I_{\max}(\cdot)$ from (4.10), and thus $\alpha(\cdot)$, $\beta(\cdot)$ from (4.9). Then, from (3.9) we derive $\partial\mathcal{X}_t$ of the integrator reach set (3.6) with time-varying input range. They together complete **step 1**.

Step 2 amounts to a simple computation: transform \mathcal{X}_t via a known linear map to obtain the desired \mathcal{Z}_t . This is justified because the map $\mathbf{x} \mapsto \mathbf{z} = \mathbf{M}^{-1}\mathbf{x}$ is a homeomorphism, so the image of the boundary $\partial\mathcal{X}_t$ is the boundary of the image \mathcal{Z}_t .

Remark 4.1. *We note that the time-varying input range $\alpha(\cdot)$, $\beta(\cdot)$ in (4.9) depends on both $\tilde{\mathbf{A}}$, $\tilde{\mathbf{b}}$. In particular, their dependence on $\tilde{\mathbf{b}}$ occurs through the matrix \mathbf{M} in (4.3), which itself depends on $\tilde{\mathbf{b}}$ via \mathbf{q} .*

4.1.3 Input set in Brunovsky coordinates

To determine the time-varying input range $[\alpha(\cdot), \beta(\cdot)]$, we first express the f in (4.11) as a *finite* sum involving the eigenvalues of the original state matrix $\tilde{\mathbf{A}}$.

Lemma 4.1. *Suppose that the original state matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{d \times d}$ has d distinct eigenvalues $\lambda_1, \dots, \lambda_d \in \mathbb{C}$. Then the function f in (4.11) can be expressed as*

$$f(\tau) = - \sum_{i=1}^d \frac{\lambda_i^d}{\prod_{j \neq i} (\lambda_i - \lambda_j)} e^{\lambda_i(s-\tau)}, \quad 0 \leq \tau \leq s. \quad (4.12)$$

The proof is provided in Appendix D.1.

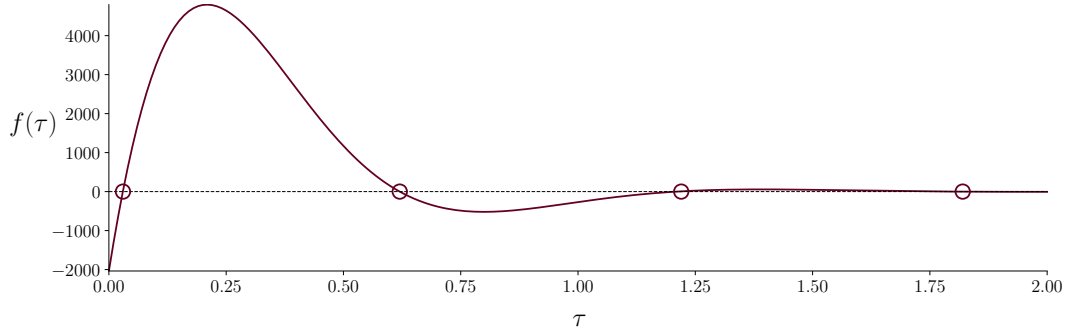


Figure 4.1: The $f(\tau)$ (solid line) in (4.12) for $\tilde{\mathbf{A}} = \begin{bmatrix} 6 & 7 & 2 \\ -4 & -2 & 1 \\ -5 & 3 & 2 \end{bmatrix}$, and its four zeros (circular markers) for $\tau \in [0, 2]$. Here $\tilde{\mathbf{A}}$ has one real and two complex conjugate eigenvalues.

Example 4.1. Let $d = 2$ and $\lambda_{1,2} = \rho e^{\pm i\phi}$, $i := \sqrt{-1}$, with $\rho, \phi \neq 0$. Then (4.12) gives

$$f(\tau) = -\frac{\rho}{\sin \phi} \sin(2\phi + (s - \tau)\rho \sin(2\phi)), \quad 0 \leq \tau \leq s.$$

Example 4.2. Let $n = 3$ and $\lambda_1 = 1$, $\lambda_{2,3} = \pm i$, $i := \sqrt{-1}$. Then (4.12) gives

$$f(\tau) = -\frac{1}{2} (e^{s-\tau} + \cos(s - \tau) - \sin(s - \tau)), \quad 0 \leq \tau \leq s.$$

Theorem 4.1 presented next, shows that the zeros of the continuous function f become relevant for our purpose. For a fixed $s \in (0, t]$, the f in (4.12) may have multiple (Fig. 4.1), single or no¹ zeros in its domain $[0, s]$. At our level of generality, it is not possible to bound the number of such zeros in $[0, s]$, and we will numerically find the same.

Using Lemma 4.1, the following Theorem 4.1 quantifies how $\alpha(s), \beta(s)$, and consequently the set $\mathcal{Z}_t(\{z_0\})$, are determined by the integral of f over its zero sub-level and super-level sets.

Theorem 4.1. Suppose $(\tilde{\mathbf{A}}, \tilde{\mathbf{b}})$ is a controllable pair and matrix $\tilde{\mathbf{A}}$ has d distinct eigenvalues. For any $0 \leq s \leq t$, let \mathcal{L}_f^- (resp. \mathcal{L}_f^{++}) denote the zero sublevel (resp. strict super-

¹as in Example 4.2, since $e^\theta + \cos \theta - \sin \theta$ has no positive roots.

level) set of f given by (4.12) over $[0, s]$, i.e.,

$$\mathcal{L}_f^- := \{\tau \in [0, s] \mid f(\tau) \leq 0\}, \quad (4.13a)$$

$$\mathcal{L}_f^{++} := \{\tau \in [0, s] \mid f(\tau) > 0\}. \quad (4.13b)$$

Then the $I_{\min}(\cdot), I_{\max}(\cdot)$ in (4.9)-(4.10) can be computed as

$$I_{\min}(s) = v_{\min} - v_{\min} \int_{\tau \in \mathcal{L}_f^-} f(\tau) d\tau - v_{\max} \int_{\tau \in \mathcal{L}_f^{++}} f(\tau) d\tau, \quad (4.14a)$$

$$I_{\max}(s) = v_{\max} - v_{\max} \int_{\tau \in \mathcal{L}_f^-} f(\tau) d\tau - v_{\min} \int_{\tau \in \mathcal{L}_f^{++}} f(\tau) d\tau. \quad (4.14b)$$

Furthermore, the LTI reach set (4.2) with time-invariant input range $[v_{\min}, v_{\max}]$ can be recovered from the integrator reach set (3.6) with time-varying input range $[\alpha(s), \beta(s)]$ given by (4.9), as

$$\mathcal{Z}_t(\{\mathbf{z}_0\}) = \mathbf{M}^{-1} \mathcal{X}_t(\{\mathbf{M} \mathbf{z}_0\}).$$

The proof of Theorem 4.1 is provided in Appendix D.2.

Remark 4.2. The d dimensional Lebesgue volume of the LTI reach set (4.2) at time t , is

$$\text{vol}(\mathcal{Z}_t(\{\mathbf{z}_0\})) = \frac{1}{|\det(\mathbf{M})|} \text{vol}(\mathcal{X}_t(\{\mathbf{M} \mathbf{z}_0\})), \quad (4.15)$$

where $\text{vol}(\mathcal{X}_t(\{\mathbf{M} \mathbf{z}_0\}))$ is determined via Theorem 3.2.

Having determined the $I_{\min}(\cdot), I_{\max}(\cdot)$ from (4.14), we now provide an example to demonstrate our approach presented in Ch. 4.1.2 to determine \mathcal{Z}_t .

4.1.4 Example for single input LTI reach set

Consider the LTI reach set \mathcal{Z}_t in (4.2) at $t = 3$ with $d = 2$, $z_0 = \mathbf{0}_{2 \times 1}$, for a system of the form (4.1):

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 0.1 & 0.2 \\ -0.3 & 0.1 \end{pmatrix}}_{\tilde{\mathbf{A}}} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \underbrace{\begin{pmatrix} 1 \\ 2 \end{pmatrix}}_{\tilde{\mathbf{b}}} v, \quad (4.16)$$

where the input ($v(\cdot)$) trajectories are in $\{v(\cdot) \in C([0, t]) \mid v(s) \in [-0.2, 0.2] \forall s \in [0, t]\}$. So $v_{\min} = -0.2, v_{\max} = 0.2$.

To compute \mathcal{Z}_t , we follow **step 1** and **step 2** from Ch. 4.1.2. Specifically, we have

$$\mathbf{M} = \begin{pmatrix} 20/11 & -10/11 \\ 5/11 & 3/11 \end{pmatrix}, \quad \mathbf{M}^{-1} = \begin{pmatrix} 3/10 & 1 \\ -1/2 & 2 \end{pmatrix}, \quad (4.17)$$

and (4.16) transforms to the controllable canonical form

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -0.07 & 0.20 \end{pmatrix}}_{\mathbf{A}_{\text{con}}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\mathbf{b}} v, \quad (4.18)$$

i.e., $\mathbf{c} := (0.07, -0.20)^\top$. We find that \mathbf{A} has eigenvalues $\lambda_{1,2} = \rho e^{\pm \iota \phi}$, $\iota := \sqrt{-1}$, with $\rho = 0.2646$, $\phi = 1.1832$, and the f in (4.12) is as in **Example 4.1**.

With this f , we numerically compute $I_{\min}(s), I_{\max}(s)$ from (4.14), and thereby the extremal trajectories $u_{\min}(s), u_{\max}(s)$ from (4.9) $\forall s \in [0, t = 3]$. Using these $u_{\min}(s), u_{\max}(s)$ and the initial condition $\mathbf{x}_0 = \mathbf{M}z_0 = \mathbf{0}_{2 \times 1}$, we use (2.16) to explicitly compute $\partial \mathcal{X}_t(\{\mathbf{0}_{2 \times 1}\})$. We used trapezoidal approximations with step-size $\Delta\tau = 0.01$ for evaluating the integrals in (4.14) and (2.16). This completes the **step 1**.

In **step 2**, we map $\partial \mathcal{X}_t(\{\mathbf{0}_{2 \times 1}\})$ back to $\partial \mathcal{Z}_t(\{\mathbf{0}_{2 \times 1}\})$ via the known linear map $\mathbf{x}^{\text{bdy}} \mapsto \mathbf{M}^{-1} \mathbf{x}^{\text{bdy}} \in \partial \mathcal{Z}_t$.

Fig. 4.2 plots the snapshots of $\mathcal{Z}_t(\{\mathbf{0}_{2 \times 1}\})$ computed as above, at $t = 1, 1.5, 2, 2.5$

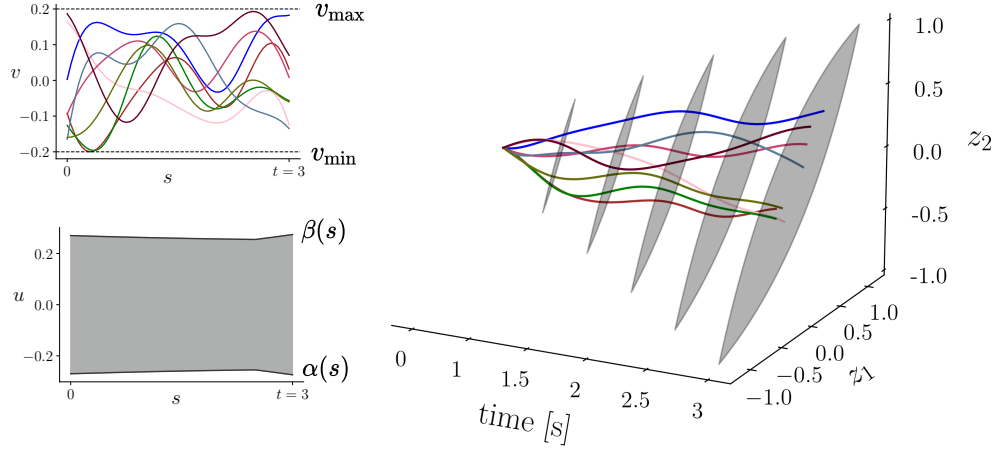


Figure 4.2: The reach sets $\mathcal{Z}_t(\{\mathbf{0}_{2 \times 1}\})$ at $t = 1, 1.5, 2, 2.5, 3$ (grey filled) for **Example 4.1.4**. These sets were computed via the proposed two step method in Ch. 4.1.2. The 8 sample state trajectories shown here correspond to 8 randomly sampled truncated Gaussian process input paths in $\{v(\cdot) \in C([0, t]) \mid v(s) \in [-0.2, 0.2] \forall s \in [0, t]\}$ shown in the left inset plot. The right inset plot shows the time-varying range $[\alpha(s), \beta(s)]$.

and 3, together with 8 sample state trajectories of (4.16) with the same zero initial state. These sample state trajectories correspond to 8 randomly sampled truncated Gaussian process input paths in $\{v(\cdot) \in C([0, t]) \mid v(s) \in [-0.2, 0.2] \forall s \in [0, t]\}$.

Now we like to determine the the volume of \mathcal{Z}_2 . From (3.14) we have

$$\begin{aligned}\zeta_1 &= 2 \left(\int_0^{\sigma_1} (t - \tau) d\tau - \int_{\sigma_1}^t (t - \tau) d\tau \right) = t^2 - 2(t - \sigma_1)^2, \\ \zeta_2 &= 2 \left(\int_0^{\sigma_1} d\tau - \int_{\sigma_1}^t d\tau \right) = 4\sigma_1 - 2t.\end{aligned}$$

Thus (3.14) becomes

$$\begin{aligned}\left| \det(D\pi) \right| &= \mu(\sigma_1) |4\lambda - 2| \left| \det \begin{pmatrix} t - \sigma_1 & t^2 - 2(t - \sigma_1)^2 \\ 1 & 4\sigma_1 - 2t \end{pmatrix} \right| \\ &= \mu(\sigma_1) |4\lambda - 2| |-\sigma_1^2 - (t - \sigma_1)^2|.\end{aligned}\tag{4.19}$$

Using (4.17) and (4.19), formula (4.15) then yields

$$\text{vol}_2(\mathcal{Z}_3) = \frac{11}{10} \int_0^1 \int_{\sigma_1}^2 \mu(\sigma_1) |4\lambda - 2| (\sigma_1^2 + (3 - \sigma_1)^2) d\sigma_1 d\lambda \approx 0.3043,$$

using trapezoidal method with step-size $\Delta\tau = 0.01$ to estimate the integral.

4.2 Intersection Detection

Consider two integrator agents labeled as A and B, each with d states, m inputs and relative degree $\mathbf{r} = (r_1, \dots, r_m)^\top$. Suppose that we are interested in checking if agents A and B, starting from respective initial conditions $\mathbf{x}_0^A, \mathbf{x}_0^B \in \mathbb{R}^d$, and respective compact input sets $\mathcal{U}^A(s), \mathcal{U}^B(s) \subset \mathbb{R}^m$, $0 \leq s \leq t$, may result in *intersecting* reach sets $\mathcal{X}_t^A, \mathcal{X}_t^B \subset \mathbb{R}^d$ at a given time t . In other words, we would like to certify (or falsify) if $\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=)\emptyset$ at a given time t .

Specifically, suppose two static feedback linearizable systems labeled as A and B, each with d states, m inputs and relative degree $\mathbf{r} = (r_1, \dots, r_m)^\top$, have identical dynamics

$$\dot{\mathbf{z}}^A = \mathbf{f}(\mathbf{z}^A, \mathbf{v}^A), \quad \dot{\mathbf{z}}^B = \mathbf{f}(\mathbf{z}^B, \mathbf{v}^B). \quad (4.20)$$

The problem of interest is to check whether their reach sets $\mathcal{Z}_t^A, \mathcal{Z}_t^B \subset \mathbb{R}^d$ at a given time t , resulting from respective initial conditions $\mathbf{z}_0^A, \mathbf{z}_0^B \in \mathbb{R}^d$, and respective compact input sets $\mathcal{V}^A(s), \mathcal{V}^B(s) \subset \mathbb{R}^m$, $0 \leq s \leq t$, intersect or not. Determining so is cumbersome because $\mathcal{Z}_t^A, \mathcal{Z}_t^B$ are compact but nonconvex in general. However, checking if $\mathcal{Z}_t^A \cap \mathcal{Z}_t^B \neq (=)\emptyset$, is *equivalent* to checking if the corresponding zonoids $\mathcal{X}_t^A, \mathcal{X}_t^B \subset \mathbb{R}^d$ in the Brunovsky normal coordinates, intersect or not. To see why, note that $\mathcal{X}_t^A, \mathcal{X}_t^B$ are diffeomorphic transforms (associated with the dynamics in (4.20)) of the sets $\mathcal{Z}_t^A, \mathcal{Z}_t^B$, i.e., there exists a diffeomorphism τ such that $\mathcal{X}_t^A = \tau(\mathcal{Z}_t^A)$, $\mathcal{X}_t^B = \tau(\mathcal{Z}_t^B)$, where $\mathcal{X}_t^A, \mathcal{X}_t^B$ are integrator reach sets (i.e., zonoids) with geometry as in Ch. 3.1.1, associated with initial conditions $\mathbf{x}_0^A := \tau(\mathbf{z}_0^A)$, $\mathbf{x}_0^B := \tau(\mathbf{z}_0^B)$, and some suitably transformed compact input sets $\mathcal{U}^A(s), \mathcal{U}^B(s) \subset \mathbb{R}^m$, $0 \leq s \leq t$. In Ch. 4.3, we will

detail the computation of such set transforms. For now, it suffices to observe that $Z_t^A \cap Z_t^B \neq (=)\emptyset \Leftrightarrow \tau(Z_t^A \cap Z_t^B) \neq (=)\emptyset$. Being a diffeomorphism, τ is injective, and we have (see e.g., [125, p. 388]):

$$\tau(Z_t^A \cap Z_t^B) = \tau(Z_t^A) \cap \tau(Z_t^B) = \mathcal{X}_t^A \cap \mathcal{X}_t^B.$$

Therefore,

$$\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=)\emptyset \Leftrightarrow Z_t^A \cap Z_t^B \neq (=)\emptyset. \quad (4.21)$$

Remark 4.3. *If the dynamics in (4.20) are dynamic feedback linearizable, but not static feedback linearizable, statement (4.21) does not hold anymore, instead we have,*

$$\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq \emptyset \implies Z_t^A \cap Z_t^B \neq \emptyset, \quad (4.22a)$$

$$\mathcal{X}_t^A \cap \mathcal{X}_t^B = \emptyset \iff Z_t^A \cap Z_t^B = \emptyset. \quad (4.22b)$$

since in this case, the diffeomorphism mapping, τ is a function of augmented states ρ , therefore we have $z = \Pi_z(\rho)$, while in the case of static feedback, diffeomorphism, τ is only a function of original states z (please refer to Ch. 1.6.1 for more details).

In the following, we show that our geometric results from Ch. 3 enables exact verification of $\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=)\emptyset$.

4.2.1 Certifying $\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=)\emptyset$

One way to check if $\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=)\emptyset$ is to compute

$$\text{dist}(A, B) := \min_{\mathbf{x}^A \in \mathcal{X}_t^A, \mathbf{x}^B \in \mathcal{X}_t^B} \|\mathbf{x}^A - \mathbf{x}^B\|_2^2. \quad (4.23)$$

Clearly, $\text{dist}(A, B) = (>)0 \Leftrightarrow \mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=)\emptyset$. However, the parametric formula in Ch. 3.2.1 suggest that a direct computation of (4.23) is unwieldy in our case. To

circumvent this difficulty, we take an alternative approach based on the difference set of \mathcal{X}_t^A and \mathcal{X}_t^B , given by the compact convex set

$$\mathcal{X}_t^A \dot{-} \mathcal{X}_t^B := \{\mathbf{x}^A - \mathbf{x}^B \mid \mathbf{x}^A \in \mathcal{X}_t^A, \mathbf{x}^B \in \mathcal{X}_t^B\}. \quad (4.24)$$

Notice that $\mathcal{X}_t^A \dot{-} \mathcal{X}_t^B = \mathcal{X}_t^A \dot{+} (-\mathcal{X}_t^B)$ where $\dot{+}$ denotes the Minkowski sum, and $\mathcal{X}_t^A \dot{-} \mathcal{X}_t^B$ is not the same as the Minkowski a.k.a. Pontryagin difference [47, p. 139], [126].

Checking the intersection between \mathcal{X}_t^A and \mathcal{X}_t^B , is then equivalent to verifying if the zero vector $\mathbf{0} \in \mathbb{R}^n$ belongs to the set (4.24). This can in turn be related [64, 65] to conditions on the support function $h_{\mathcal{X}_t^A \dot{-} \mathcal{X}_t^B}(\cdot)$, because

$$\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq \emptyset \Leftrightarrow \mathbf{0} \in \mathcal{X}_t^A \dot{-} \mathcal{X}_t^B \Leftrightarrow \forall \mathbf{y} \in \mathbb{S}^{n-1}, h_{\mathcal{X}_t^A \dot{-} \mathcal{X}_t^B}(\mathbf{y}) \geq 0, \quad (4.25a)$$

$$\mathcal{X}_t^A \cap \mathcal{X}_t^B = \emptyset \Leftrightarrow \mathbf{0} \notin \mathcal{X}_t^A \dot{-} \mathcal{X}_t^B \Leftrightarrow \exists \mathbf{y} \in \mathbb{S}^{n-1} \text{ such that } h_{\mathcal{X}_t^A \dot{-} \mathcal{X}_t^B}(\mathbf{y}) < 0. \quad (4.25b)$$

Thus motivated, we propose certifying or falsifying the reach set intersection by computing

$$\min_{\mathbf{y} \in \mathbb{S}^{d-1}} h_{\mathcal{X}_t^A \dot{-} \mathcal{X}_t^B}(\mathbf{y}). \quad (4.26)$$

Specifically, $(4.26) \geq (<)0 \Leftrightarrow \mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=)\emptyset$. In other words, (4.26) serves as a variational oracle to certify or falsify the reach set intersection.

Remark 4.4. *In general, which of the two problems (4.23) and (4.26) is computationally more tractable, depends on the sets $\mathcal{X}_t^A, \mathcal{X}_t^B$. For instance, when $\mathcal{X}_t^A, \mathcal{X}_t^B$ are ellipsoids, simple algorithms are known (see e.g., [127]) for solving (4.23), but solving (4.26) requires heavier computation [128]. In our context, $\mathcal{X}_t^A, \mathcal{X}_t^B$ are zonoids, and detecting their intersection using (4.26) turns out to be computationally more benign than (4.23).*

Since support function is distributive over the Minkowski sum, we have

$$h_{\mathcal{X}_t^A \dot{-} \mathcal{X}_t^B}(\mathbf{y}) = h_{\mathcal{X}_t^A}(\mathbf{y}) + h_{-\mathcal{X}_t^B}(\mathbf{y}). \quad (4.27)$$

From the definition of support function, we also have

$$h_{-\mathcal{X}_t^B}(\mathbf{y}) = \sup_{\mathbf{x} \in -\mathcal{X}_t^B} \langle \mathbf{y}, \mathbf{x} \rangle = \sup_{\mathbf{x} \in \mathcal{X}_t^B} \langle \mathbf{y}, -\mathbf{x} \rangle = \sup_{\mathbf{x} \in \mathcal{X}_t^B} \langle -\mathbf{y}, \mathbf{x} \rangle = h_{\mathcal{X}_t^B}(-\mathbf{y}). \quad (4.28)$$

Using (4.27) and (4.28), we rewrite (4.26) as

$$\min_{\|\mathbf{y}\|_2=1} h_{\mathcal{X}_t^A}(\mathbf{y}) + h_{\mathcal{X}_t^B}(-\mathbf{y}). \quad (4.29)$$

Recall that a support function is convex in its argument, and convex composed with affine remains convex. Thus, the objective (4.29) is a sum of convex functions, and hence convex. Because \mathbb{S}^{d-1} is compact, by Weirstrass extreme value theorem, (4.29) admits global minimum. Checking $\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=)\emptyset$ reduces to checking the sign of the minimum in (4.29). We next develop these ideas for two specific choices of time-varying set-valued input uncertainties: norm bounded (Ch. 4.2.2) and hyperrectangular uncertainty sets (Ch. 4.2.3).

4.2.2 The case when $\mathcal{U}^A, \mathcal{U}^B$ are norm balls

In this section, we consider the time-varying norm-bounded input uncertainty sets as in [38, Sec. 4]

$$\mathcal{U}(s) := \left\{ \mathbf{u}(s) \in \mathbb{R}^m \mid \|\mathbf{M}(s)\mathbf{u}(s) + \mathbf{p}(s)\|_p \leq 1 \right\} \quad (4.30)$$

where $\|\cdot\|_p$ denotes the p -norm for $1 \leq p \leq \infty$, and $\mathbf{M}(s) \times \mathbf{p}(s) : [0, t]^2 \mapsto \mathbb{S}_{++}^m \times \mathbb{R}^m$ are given as smooth functions, where \mathbb{S}_{++}^m represent the space of positive definite matrices. By specializing (1.32) to this specific setting, we get the following result (proof in Appendix D.3).

Theorem 4.2. *Consider the integrator dynamics with n states, and m inputs. For compact input uncertainty sets $\mathcal{U}(s) \subset \mathbb{R}^m$ given by (4.30) for all $0 \leq s \leq t$, and initial*

condition $\mathbf{x}_0 \in \mathbb{R}^d$, the support function of the integrator reach set \mathcal{X}_t at time t is

$$h_{\mathcal{X}_t}(\mathbf{y}) = \sum_{j=1}^m \langle \mathbf{y}_j, \exp(t\mathbf{A}_j) \mathbf{x}_{j0} \rangle + \quad (4.31)$$

$$\int_0^t \left[\langle \mathbf{y}, -\exp((t-s)\mathbf{A}) \mathbf{M}^{-1}(s) \mathbf{p}(s) \rangle + \|(\exp((t-s)\mathbf{A}) \mathbf{B} \mathbf{M}^{-1})^\top \mathbf{y}\|_q \right] ds \quad (4.32)$$

where q is the Hölder conjugate of p , i.e., $\frac{1}{p} + \frac{1}{q} = 1$.

Remark 4.5. From definition (1.29) we know that the reach set \mathcal{X}_t resulting from compact $\mathcal{U}(\cdot)$ is the same as that resulting from the closure of the convex hull of $\mathcal{U}(\cdot)$. Consequently, if the p in (4.30) satisfies $0 < p < 1$, thus making the input norm balls nonconvex, then the corresponding reach sets will coincide with that resulting from the $p = 1$ norm ball input uncertainty sets. This allows the effective domain of p in (4.33) to be $(0, \infty]$.

without loss of generality and to aid the readability, we consider the norm ball to be symmetric about the origin, and given by

$$\mathcal{U}(s) := \left\{ \mathbf{u}(s) \in \mathbb{R}^m \mid \|\mathbf{u}(s)\|_p \leq \ell(s) \right\} \text{ for all } s \in [0, t], \quad (4.33)$$

where $\|\cdot\|_p$ denotes the p -norm for $1 \leq p \leq \infty$, and $\ell : [0, t] \mapsto \mathbb{R}_{>0}$ is a given smooth function. For this setting, the support function is simplified to

$$h_{\mathcal{X}_t}(\mathbf{y}) = \sum_{j=1}^m \langle \mathbf{y}_j, \exp(t\mathbf{A}) \mathbf{x}_{j0} \rangle + \int_0^t \ell(s) \|(\exp((t-s)\mathbf{A}) \mathbf{B})^\top \mathbf{y}\|_q ds, \quad (4.34)$$

A translation of the norm ball does not change the arguments. We next detail how having analytic handle on the support function of the reach set as in (4.34) can help detect reach set intersection among integrator agents using (4.29).

Lossless convexification

We suppose that the integrator agents A and B have input uncertainty sets $\mathcal{U}^A(s)$, $\mathcal{U}^B(s)$ as in (4.33) with same p , respective bounds $\ell^A(s)$, $\ell^B(s)$, and respective initial

conditions $\mathbf{x}_0^A, \mathbf{x}_0^B \in \mathbb{R}^d$.

The associated problem (4.29) is nonconvex due to the unit sphere constraint $\|\mathbf{y}\|_2=1$. We convexify the same by relaxing it to the unit ball constraint $\|\mathbf{y}\|_2 \leq 1$. Since $\ell^A(s), \ell^B(s)$ are positive for all $0 \leq s \leq t$, the convexified version of (4.29) becomes

$$\min_{\mathbf{y} \in \mathbb{R}^n, \|\mathbf{y}\|_2 \leq 1} \langle \mathbf{c}(t), \mathbf{y} \rangle + \int_0^t \|(\mathbf{G}(s))^\top \mathbf{y}\|_q \, ds, \quad (4.35)$$

where

$$\mathbf{c}(t) := \exp(t\mathbf{A})(\mathbf{x}_0^A - \mathbf{x}_0^B), \quad (4.36a)$$

$$\mathbf{G}(s) := (\ell^A(s) + \ell^B(s)) \exp((t-s)\mathbf{A})\mathbf{B}. \quad (4.36b)$$

We approximate the integral in (4.35) w.r.t. s via trapezoidal approximation² with uniform step-size $\Delta s > 0$. In particular, uniformly discretizing $[0, t]$ into $K \in \mathbb{N}$ intervals with breakpoints $s_k = k\Delta s$ for $k = 0, 1, \dots, K$, where $\Delta s := t/K$, results in the trapezoidal approximation

$$\int_0^t \|(\mathbf{G}(s))^\top \mathbf{y}\|_q \, ds \approx \frac{\Delta s}{2} \sum_{k=1}^K (\|(\mathbf{G}(s_{k-1}))^\top \mathbf{y}\|_q + \|(\mathbf{G}(s_k))^\top \mathbf{y}\|_q). \quad (4.37)$$

Letting $\|(\mathbf{G}(s_k))^\top \mathbf{y}\|_q \leq \theta_k$, and $\boldsymbol{\theta} := (\theta_0, \theta_1, \dots, \theta_K) \in \mathbb{R}_{\geq 0}^{K+1}$, define

$$\boldsymbol{\eta} := \begin{pmatrix} \mathbf{y} \\ \boldsymbol{\theta} \end{pmatrix} \in \mathbb{R}^{d+K+1}, \quad (4.38a) \quad \mathbf{M}_k := \begin{pmatrix} \mathbf{G}(s_k) & \mathbf{0}_{(1+K) \times m} \end{pmatrix}^\top \in \mathbb{R}^{m \times (d+K+1)} \quad (4.38d)$$

$$\boldsymbol{\omega} := \Delta s \begin{pmatrix} 1/2 \\ \mathbf{1}_{K-1} \\ 1/2 \end{pmatrix} \in \mathbb{R}_{>0}^{K+1}, \quad (4.38b) \quad \mathbf{N} := \left(\mathbf{I}_d \mid \mathbf{0}_{d \times (K+1)} \right) \in \mathbb{R}^{d \times (d+K+1)}. \quad (4.38e)$$

$$\boldsymbol{\kappa}(t) := \begin{pmatrix} \mathbf{c}(t) \\ \boldsymbol{\omega} \end{pmatrix} \in \mathbb{R}^{d+K+1} \quad (4.38c) \quad \tilde{\mathbf{N}} := \left(\mathbf{0}_{(K+1) \times d} \mid \mathbf{I}_{K+1} \right) \in \mathbb{R}^{(K+1) \times (d+K+1)}. \quad (4.38f)$$

²The uniform trapezoidal approximation with local truncation error $\mathcal{O}((\Delta s)^3)$ may be replaced by other approximations such as the three point Simpson's rule with local truncation error $\mathcal{O}((\Delta s)^4)$. While the accuracy of different numerical approximations for the integral may vary depending on the choice of approximation but the nature of the resulting optimization problems will remain the same.

In (4.38), the symbols $\mathbf{1}$, $\mathbf{0}$, \mathbf{I} respectively denote the array of ones, zeros and identity matrix of appropriate sizes.

With the above variable definitions, we transcribe (4.35) into the epigraph form

$$\min_{\boldsymbol{\eta} \in \mathbb{R}^{d+K+1}} \langle \boldsymbol{\kappa}(t), \boldsymbol{\eta} \rangle \quad (4.39a)$$

$$\text{subject to, } \quad \|\mathbf{M}_k \boldsymbol{\eta}\|_q - \langle \mathbf{e}_{d+k}^{d+K+1}, \boldsymbol{\eta} \rangle \leq 0, \text{ for all } k = 0, \dots, K, \quad (4.39b)$$

$$-\tilde{\mathbf{N}} \boldsymbol{\eta} \leq \mathbf{0}, \quad (4.39c)$$

$$\|\mathbf{N} \boldsymbol{\eta}\|_2 \leq 1, \quad (4.39d)$$

where the vector inequality in (4.39c) is elementwise.

Notice that (4.39d) results from the convexification of the nonconvex constraint $\|\mathbf{y}\|_2 = 1$ in (4.29). Modulo the numerical approximation of the integral, the problem (4.39) is a convex relaxation of the problem (4.29). As we show next, this relaxation is exact (i.e., lossless convexification) in the sense that solving (4.39) allows us to certify $\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=) \emptyset$.

Theorem 4.3. *Let p^* be the optimal value of (4.39) with the \leq in (4.39d) replaced by equality. In other words, p^* is the optimal value obtained by solving (4.29) with the numerical approximations for the integrals stated in (4.37). Let \tilde{p}^* be the optimal value of (4.39). Then, the following holds:*

- (i) $\tilde{p}^* \leq 0$.
- (ii) $\tilde{p}^* = 0 \Rightarrow 0 \leq p^* \Leftrightarrow \mathcal{X}_t^A \text{ and } \mathcal{X}_t^B \text{ intersect.}$
- (iii) $\tilde{p}^* < 0 \Rightarrow \tilde{p}^* = p^* < 0 \Leftrightarrow \mathcal{X}_t^A \text{ and } \mathcal{X}_t^B \text{ are disjoint.}$

The proof for Theorem 4.3 is provided in Appendix D.4. The convex problem (4.39) can be solved numerically using standard interior point algorithms.

4.2.3 The Case when $\mathcal{U}^A, \mathcal{U}^B$ are hyperrectangles

We next consider a generalized version of the ∞ -norm bounded input uncertainties in the sense we allow hyperrectangular or box-valued input uncertainty sets of the form (3.2). In this setting has been investigated in 3. Define $\alpha^i(s), \beta^i(s), \mu^i(s), \nu^i(s)$ as in (3.2), associated with respective input sets $\mathcal{U}^i(s)$ for the agents $i \in \{A, B\}$, for all $0 \leq s \leq t$. Following Proposition 3.1, we then obtain $h_{\mathcal{X}_t^A}(\mathbf{y})$ and $h_{\mathcal{X}_t^B}(\mathbf{y})$. It then remains to solve (4.29).

Distributed computation

Instead of directly substituting $h_{\mathcal{X}_t^A}(\mathbf{y})$ and $h_{\mathcal{X}_t^B}(\mathbf{y})$ in (4.29), we make the observation that

$$\mathcal{X}_t^A \cap \mathcal{X}_t^B = \left(\prod_{j=1}^m \mathcal{X}_{jt}^A \right) \cap \left(\prod_{j=1}^m \mathcal{X}_{jt}^B \right) = \prod_{j=1}^m (\mathcal{X}_{jt}^A \cap \mathcal{X}_{jt}^B), \quad (4.40)$$

where \prod denotes the Cartesian product, and $\mathcal{X}_{jt}^A, \mathcal{X}_{jt}^B \subset \mathbb{R}^{r_j}$ are the respective j th single input integrator reach sets resulting from their input sets $[\alpha_j^A(s), \beta_j^A(s)]$ and $[\alpha_j^B(s), \beta_j^B(s)]$, $0 \leq s \leq t$. From (4.40), it follows that $\mathcal{X}_t^A \cap \mathcal{X}_t^B = \emptyset$ iff there exists $j \in \llbracket m \rrbracket$ such that $\mathcal{X}_{jt}^A \cap \mathcal{X}_{jt}^B = \emptyset$.

Therefore, it suffices to check whether these *single input* integrator reach sets $\mathcal{X}_{jt}^A, \mathcal{X}_{jt}^B$ intersect or not. Consequently, problem (4.29) can be solved in a *distributed manner*, i.e., by separately solving

$$\min_{\mathbf{y}_j \in \mathbb{R}^{r_j}, \|\mathbf{y}_j\|_2=1} h_{\mathcal{X}_{jt}^A}(\mathbf{y}_j) + h_{\mathcal{X}_{jt}^B}(-\mathbf{y}_j), \quad (4.41)$$

for all $j \in \llbracket m \rrbracket$, and then checking the signs of these m minimum values. In summary, \mathcal{X}_t^A and \mathcal{X}_t^B intersect iff (4.41) yields ≥ 0 for all $j \in \llbracket m \rrbracket$. We relax the unit sphere constraint $\|\mathbf{y}_j\|_2 = 1$ to $\|\mathbf{y}_j\|_2 \leq 1$ in subproblems (4.41) with support functions $h_{\mathcal{X}_{jt}}$ given by (3.7). Following the same steps as Ch. 4.2.2, we can rewrite subproblems

(4.41) for each $j \in \llbracket m \rrbracket$, as SOCP:

$$\min_{\boldsymbol{\eta}_j^{\text{box}} \in \mathbb{R}^{r_j+K+1}} \langle \boldsymbol{\kappa}_j^{\text{box}}(t), \boldsymbol{\eta}_j^{\text{box}} \rangle \quad (4.42a)$$

$$\text{subject to } \mathbf{M}_j^{\text{box}} \boldsymbol{\eta}_j^{\text{box}} \leq \mathbf{0}, \quad (4.42b)$$

$$\|\mathbf{N}_j^{\text{box}} \boldsymbol{\eta}_j^{\text{box}}\|_2 \leq 1, \quad (4.42c)$$

where

$$\mathbf{c}_j^{\text{box}}(t) := \exp(t\mathbf{A}_j) (\mathbf{x}_{j0}^{\text{A}} - \mathbf{x}_{j0}^{\text{B}}) + \int_0^t (\nu_j^{\text{A}}(s) - \nu_j^{\text{B}}(s)) \boldsymbol{\xi}_j(t-s) ds, \quad (4.43a)$$

$$\boldsymbol{\gamma}_j^{\text{box}}(s) := (\mu_j^{\text{A}}(s) + \mu_j^{\text{B}}(s)) \boldsymbol{\xi}_j(t-s). \quad (4.43b)$$

$$\boldsymbol{\eta}_j^{\text{box}} := \begin{pmatrix} \mathbf{y}_j \\ \boldsymbol{\theta}_j \end{pmatrix} \in \mathbb{R}^{r_j+K+1}, \quad (4.43c) \quad \mathbf{\Gamma}_j^{\text{box}} := \begin{pmatrix} \boldsymbol{\gamma}_j^{\text{box}\top}(s_0) \\ -\boldsymbol{\gamma}_j^{\text{box}\top}(s_0) \\ \vdots \\ \boldsymbol{\gamma}_j^{\text{box}\top}(s_K) \\ -\boldsymbol{\gamma}_j^{\text{box}\top}(s_K) \end{pmatrix} \in \mathbb{R}^{2(K+1) \times r_j}, \quad (4.43d)$$

$$\boldsymbol{\omega}_j^{\text{box}} := \Delta s \begin{pmatrix} 1/2 \\ \mathbf{1}_{K-1} \\ 1/2 \end{pmatrix} \in \mathbb{R}_{>0}^{K+1}, \quad \mathbf{M}_j^{\text{box}} := \left(\begin{array}{c|c} \mathbf{\Gamma}_j^{\text{box}} & -\mathbf{I}_{K+1} \otimes \mathbf{1}_2 \\ \hline \mathbf{0}_{(K+1) \times r_j} & -\mathbf{I}_{K+1} \end{array} \right) \in \mathbb{R}_{(r_j+K+1)}^{3(K+1) \times}, \quad (4.43e) \quad (4.43f)$$

$$\boldsymbol{\kappa}_j^{\text{box}}(t) := \begin{pmatrix} \mathbf{c}_j^{\text{box}}(t) \\ \boldsymbol{\omega}_j^{\text{box}} \end{pmatrix} \in \mathbb{R}^{r_j+K+1}, \quad \mathbf{N}_j^{\text{box}} := \left(\begin{array}{c|c} \mathbf{I}_{r_j} & \mathbf{0}_{r_j \times (K+1)} \end{array} \right) \in \mathbb{R}^{r_j \times (r_j+K+1)}, \quad (4.43g) \quad (4.43h)$$

and the symbol \otimes denotes the Kronecker product. We observe that (4.42c) results from the convexification of the nonconvex constraint $\|\mathbf{y}_j\|_2 = 1$ in (4.41) for each $j \in \llbracket m \rrbracket$. As in Ch. 4.2.2, this turns out to be a lossless convexification in the sense that solving (4.42) allows us to certify $\mathcal{X}_{jt}^{\text{A}} \cap \mathcal{X}_{jt}^{\text{B}} \neq (=) \emptyset$. We summarize this in the following

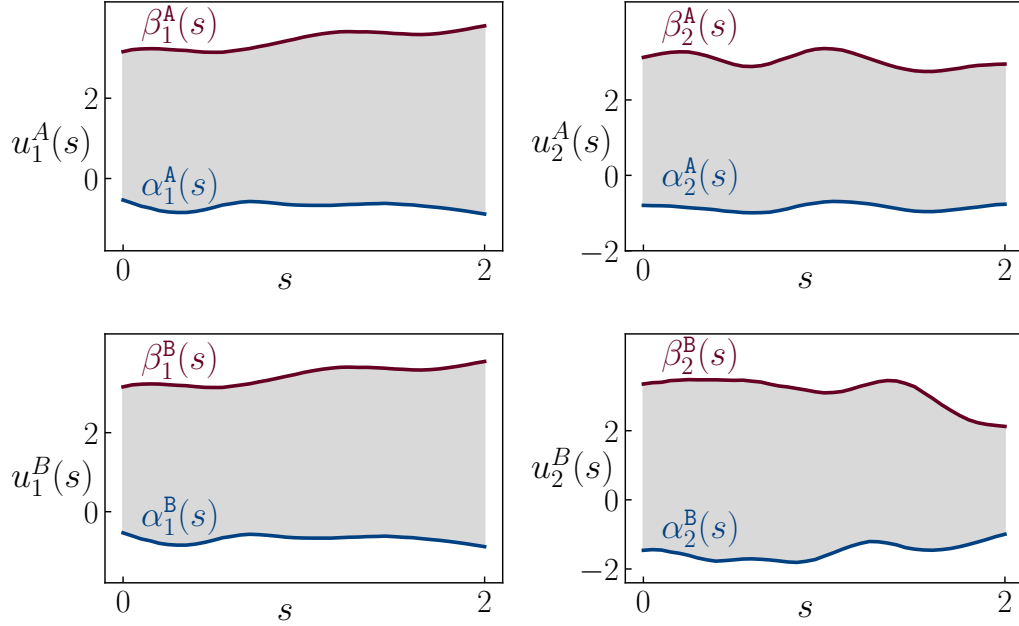


Figure 4.3: The input uncertainties for agent A with $u_j^A(s) \in [\alpha_j^A(s), \beta_j^A(s)]$, and agent B with $u_j^B(s) \in [\alpha_j^B(s), \beta_j^B(s)]$, where $j \in \{1, 2\}$ for the example in Ch. 4.2.4. The controlled dynamics of the agents are given by (1.13). At each $s \in [0, 2]$, $\alpha_j^i(s)$ and $\beta_j^i(s)$ respectively represent the minimum and maximum of the j th coordinate of the input set $\mathcal{U}^i(t)$ for $i \in \{A, B\}$.

statement whose proof follows the same steps as in Appendix D.4, and is omitted.

Theorem 4.4. For $j \in \llbracket m \rrbracket$, let p_j^* be the optimal value of (4.42) with the \leq in (4.42c) replaced by equality. Let \tilde{p}_j^* be the optimal value of (4.42). Then, the following holds:

- (i) $\tilde{p}_j^* \leq 0$.
- (ii) $\tilde{p}_j^* = 0 \Rightarrow 0 \leq p_j^* \Leftrightarrow \mathcal{X}_{jt}^A$ and \mathcal{X}_{jt}^B intersect.
- (iii) $\tilde{p}_j^* < 0 \Rightarrow \tilde{p}_j^* = p_j^* < 0 \Leftrightarrow \mathcal{X}_{jt}^A$ and \mathcal{X}_{jt}^B are disjoint.

Since (4.40) tells us $\mathcal{X}_t^A \cap \mathcal{X}_t^B = \emptyset$ iff there exists $j \in \llbracket m \rrbracket$ such that $\mathcal{X}_{jt}^A \cap \mathcal{X}_{jt}^B = \emptyset$, therefore the distributed computation of (4.42) allows certification or falsification of integrator reach sets subject to box-valued input uncertainties.

In the following, we provide a numerical example to illustrate our results.

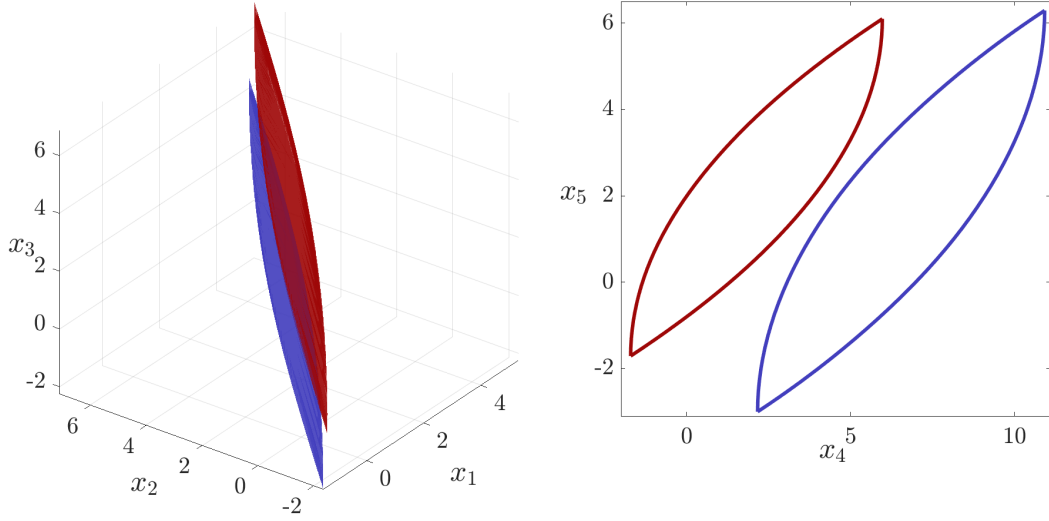


Figure 4.4: Intersection detection between two integrator reach sets \mathcal{X}_t^A (red) and \mathcal{X}_t^B (blue) described in Ch. 4.2.4 with relative degree vectors $\mathbf{r}^A = \mathbf{r}^B = (3, 2)^\top$. The agents start from initial conditions $\mathbf{x}_0^A = (0.5, \mathbf{0}_{1 \times 4})^\top$ and $\mathbf{x}_0^B = (\mathbf{0}_{1 \times 3}, 5, 0)^\top$. In this case, $\mathcal{X}_t^A = \mathcal{X}_{1t}^A \times \mathcal{X}_{2t}^A$, $\mathcal{X}_t^B = \mathcal{X}_{1t}^B \times \mathcal{X}_{2t}^B \subset \mathbb{R}^5$. *Left*: Intersection between the sets \mathcal{X}_{1t}^A (red) and \mathcal{X}_{1t}^B (blue). *Right*: no intersection between the sets \mathcal{X}_{2t}^A (red) and \mathcal{X}_{2t}^B (blue). These plots are made by generalizing the parametric boundary formula of the integrator reach sets in [129, proposition 1] to account for time-varying set-valued uncertainties.

4.2.4 Numerical example for intersection detection

Let us consider two integrator agents A and B with relative degrees $\mathbf{r}^A = \mathbf{r}^B = (3, 2)^\top$ as in (1.13), starting from respective initial conditions $\mathbf{x}_0^A = (0.5, \mathbf{0}_{1 \times 4})^\top$, $\mathbf{x}_0^B = (\mathbf{0}_{1 \times 3}, 5, 0)^\top$. Given two box-valued input sets $\mathcal{U}^A(s) = [\alpha_1^A(s), \beta_1^A(s)] \times [\alpha_2^A(s), \beta_2^A(s)]$ and $\mathcal{U}^B(s) = [\alpha_1^B(s), \beta_1^B(s)] \times [\alpha_2^B(s), \beta_2^B(s)]$ shown in Fig. 4.4, we want to check if there is an intersection at time $t = 2$ between the reach sets of agent A, denoted as \mathcal{X}_t^A , and that of agent B, denoted as \mathcal{X}_t^B .

As explained in Ch. 4.2.3, it suffices to check if there will be an intersection between each corresponding j th single input reach sets \mathcal{X}_{jt}^A and \mathcal{X}_{jt}^B , for $j \in \{1, 2\}$.

Using (4.43), we construct the matrices $\mathbf{M}_j^{i, \text{box}}$, $\mathbf{N}_j^{i, \text{box}}$ and the vector $\boldsymbol{\kappa}_j^{i, \text{box}}$ for $i \in \{A, B\}$. Then, we solve the optimization problems (4.42) for each $j \in \{1, 2\}$ via MATLAB CVX toolbox [110, 130] with $\Delta s = 0.05$. The runtimes are 0.38 s and 0.37

s for $j = 1$ and $j = 2$, respectively. The corresponding optimal values are $(\tilde{p}_1^*, \tilde{p}_2^*) = (0, -0.54)$. These optimal values imply $\mathcal{X}_{1t}^A \cap \mathcal{X}_{1t}^B \neq \emptyset$, and $\mathcal{X}_{2t}^A \cap \mathcal{X}_{2t}^B = \emptyset$. Therefore, we conclude: $\mathcal{X}_t^A \cap \mathcal{X}_t^B = \emptyset$.

In agreement with Theorem 4.4, for $\tilde{p}_1^* = 0$, CVX returns $\|\mathbf{N}_1 \boldsymbol{\eta}_1\|_2 = 0$, while for $\tilde{p}_2^* < 0$, it returns $\|\mathbf{N}_2 \boldsymbol{\eta}_2\|_2 = 1$ demonstrating that the convexification (4.42c) is lossless. This scenario is illustrated in Fig. 4.4.

In this section, we presented a variational formulation for certifying or falsifying intersection of the reach sets of integrator agents subject to set-valued input uncertainties. The proposed nonconvex formulation is shown to enjoy lossless convexification for time-varying norm bounded as well as generic hyperrectangular input uncertainties, thus being amenable to convex programming for tractable computation.

4.3 Learning Reach Set for Full State Feedback Linearizable Systems

Consider the a full-state differentially flat system in (1.5) as described in Ch. 1.6.1. We suppose that \mathcal{V} is a known compact convex set, modeling physical actuation bounds. Thanks to homeomorphism $\tau_u : \mathcal{V} \rightarrow \mathcal{U}$, the set $\mathcal{U}(t)$ is guaranteed to be compact but not necessarily convex. Furthermore, even if the set \mathcal{V} is time-invariant, the set $\mathcal{U}(t)$ will be time-varying due to the state-dependence in (1.8a) and (1.10).

Instead of directly approximating \mathcal{Z}_t via Monte Carlo, our idea is to make use of the exact support function, the parametric formula and the knowledge of the diffeomorphism τ . Thus, at a conceptual level, we'd like to follow the computational sequence:

$$\mathcal{V} \longrightarrow \mathcal{U}(t) \longrightarrow \partial \mathcal{X}_t^\square \longrightarrow \partial \mathcal{R}_t \longrightarrow \partial \mathcal{Z}_t.$$

In words, our strategy involves the following three steps.

- i. (Compute the extremals of the control mapping)** Given compact convex \mathcal{V} , the tensor and vector field pair $(C(x), d(x)) \in \mathbb{R}^{m \times m} \times \mathbb{R}^m$, estimate $\mathcal{U}(s)$ from $\mathcal{V} \xrightarrow{\tau_u} \mathcal{U}(s)$ for all $0 \leq s \leq t$ within the family of p -norm balls.
- ii. (Compute the reach set in normal coordinates)** Using $\mathcal{U}(s)$ for all $0 \leq s \leq t$ from Step 1, compute \mathcal{X}_t^\square using our support function or the parametric formula in Theorems 2.1 and 2.3. Return \mathcal{X}_t as the Cartesian product.
- iii. (Transform the reach set back in original coordinates)** Numerically map the boundary of \mathcal{X}_t^\square back to the original state space using the known diffeomorphism $z = \tau^{-1}(x)$, to over-approximate the reach set \mathcal{Z}_t in the original state space.

The main challenge in our algorithm appears in step 1, estimating $\mathcal{U}(\cdot)$ from $\mathcal{V}(\cdot)$ via the homeomorphism τ_u . Since, estimation of $\mathcal{U}(s)$ amounts to solving a fixed point problem which is not contractive. For instance for the boxed valued case, evaluation of the extremals of the set-valued trajectory $(\mathcal{U}(s))_{s=0}^t$ i.e., $((\alpha(s))_{s=0}^t, (\beta(s))_{s=0}^t)$, leads to the following fixed point problem

$$\begin{aligned} \alpha(t) &= \mathbf{T}_{\min}(\alpha(t), \beta(t)) := \min_{(z, w)^\top \in \tau^{-1}(\mathcal{X}_t^\square)} [C(z, w)v + d(z, w)], \\ \beta(t) &= \mathbf{T}_{\max}(\alpha(t), \beta(t)) := \max_{(z, w)^\top \in \tau^{-1}(\mathcal{X}_t^\square)} [C(z, w)v + d(z, w)], \end{aligned} \quad (4.44)$$

The minimization and maximization in (4.44) are component-wise. Since these minimization and maximization problems involve continuous objectives with compact constraint sets, hence by Weirstrass extreme value theorem, the maps \mathbf{T}_{\min} , \mathbf{T}_{\max} are unique. Furthermore, since $\mathcal{V} \times \mathcal{X}_t^\square$ is compact, and the compact reach set \mathcal{X}_t^\square is

continuous in $(\alpha(t), \beta(t))$, hence by Berge's maximum theorem [131, p. 115], the maps T_{\min}, T_{\max} are continuous in $(\alpha(t), \beta(t))$.

4.3.1 Approximating feedback linearizable reach set via scenario optimization

Here we adapt an approach based on scenario optimization (see [132–134]) to estimate the input set \mathcal{U}_s via sample random trajectories $\{u^{(i)}(t)\}_{i=1}^N$ with probability approximately correct (PAC) guarantee.

Consider a random variable $u(s)$, drawn from set \mathcal{U} via probability density function (pdf) denoted as \mathbb{P}_u , where $0 \leq s \leq t$. In an ideal setting, the support of this random variable is the input set \mathcal{U} , i.e., $\mathbb{P}_u(\mathbf{u}(s) \in \mathcal{U}_s) > 0$ for any $u(s)$ inside \mathcal{U}_s . Any approximated set that partially encompasses the reachable set will possess a probability less than \mathbb{P}_u for any random samples.

Our goal is to derive an approximation of the input set denoted by $\mathcal{U}_{s,\theta}$ parameterized by $\theta \in \Theta \in \mathbb{R}^{n_\theta}$, such that given $\varepsilon \in (0, 1)$, it ensures

$$\mathbb{P}_u(\mathbf{u}(s) \in \mathcal{U}_{s,\theta}) \geq 1 - \varepsilon_u, \quad \text{for all } 0 \leq s \leq t.$$

This is termed as an ε -accurate approximation w.r.t to the distribution \mathbb{P}_u . The set Θ is compact, convex and represents the domain of the optimization variables.

The probabilistic design problem to determine $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ solves

$$\begin{aligned} & \underset{\theta}{\operatorname{argmin}} \quad \operatorname{Vol}(\mathcal{U}_{s,\theta}) \\ & \text{subject to} \quad \mathbb{P}_u(\mathbf{u}(s) \in \mathcal{U}_{s,\theta}) \geq 1 - \varepsilon_u, \end{aligned} \quad (4.45)$$

so the process of approximating the input set is simplified to identifying the design parameter θ , for all $0 \leq s \leq t$. Here, we are interested to estimate the integrator input

set within the family p-norm balls. Reflecting this chance constraints to (4.45) gives

$$\begin{aligned} & \underset{\mathbf{M}, \mathbf{p}}{\operatorname{argmin}} && -\log \det(\mathbf{M}) \\ & \text{subject to} && \mathbb{P}_u (\|\mathbf{M}\mathbf{u} - \mathbf{p}\|_p - 1 \leq 0) \leq 1 - \varepsilon_u. \end{aligned} \quad (4.46)$$

Problem (4.46) although being convex, still entails an infinite number of constraints and is categorized as NP-hard. Consider an instance of random i.i.d extraction $\Delta = \{\mathbf{u}_i\}_{i=0}^N \sim \mathbb{P}_u^N$, such that $\mathbb{P}_u^N = \prod^N \mathbb{P}_u$. Instead of dealing with the intractable probabilistic constraint, we consider a randomized optimization problem using independent random samples known as scenario optimization program

$$\begin{aligned} & \underset{[\mathbf{M}, \mathbf{p}] \in \mathcal{S}}{\operatorname{argmin}} && -\log \det(\mathbf{M}) \\ & \text{subject to} && \mathbb{P}_u (\|\mathbf{M}\mathbf{u}_i - \mathbf{p}\|_p - 1 \leq 0) \leq 1 - \varepsilon_u, \end{aligned} \quad (4.47)$$

where, each i th constraint is enforced by the realization of the random variable \mathbf{u}_i known as a scenario. Despite the original design problem (4.46), the scenario optimization has finite number of constraints and is computationally tractable and attains a unique solution (under the convexity assumption).

Since the value of $\mathbb{P}_u (\|\mathbf{M}\mathbf{u}_i - \mathbf{p}\|_p - 1 \leq 0)$ depends on the set of random extraction of $\{\mathbf{u}_i\}_{i=0}^N \sim \mathbb{P}_u^N$, we qualify these extractions using the confidence parameter $\delta_u \in (0, 1)$, in the sense that

$$\mathbb{P}_u (\|\mathbf{M}\mathbf{u}_i - \mathbf{p}\|_p - 1 \leq 0) \geq 1 - \varepsilon_u, \quad \mathbf{w}. \mathbb{P}_u^N \geq 1 - \delta_u \quad (4.48)$$

Intuitively, condition (4.48) entails that with confidence $(1 - \delta)$, the optimal solution satisfy the constraints with accuracy $1 - \varepsilon_u$. Denote the number of decision variables $n_\theta = m(m + 1)/2 + m$ (\mathbf{M} is symmetric) for $1 \leq p \leq \infty$, then conditions (4.48) entails

that \mathbb{P}_u^N is the binomial distribution given by

$$\mathbb{P}_u^N = \sum_{i=1}^{n_\theta} \binom{N}{i} \varepsilon_u^i (1 - \varepsilon_u)^{N-i}, \quad (4.49)$$

where N is the number of required constraints (scenarios) to satisfy the PAC condition (4.48), also known as the *sample complexity*. It is noteworthy that the term on the right-hand side of Equation (4.49) converges to zero rapidly as the number of scenarios increases. This observation implies that, for sufficiently large values of N , the likelihood of obtaining a solution characterized as "undesirable" in terms of violation probability (i.e., one with a violation exceeding the specified threshold) becomes exceedingly low [135]. Equation (4.49) can also be inverted to explicitly express the relationship between N and the desired probability levels, as follows.

$$1 - \delta_u \leq \sum_{i=1}^{n_\theta} \binom{N}{i} \varepsilon_u^i (1 - \varepsilon_u)^{N-i}. \quad (4.50)$$

Campi [132] has shown that the above upper bound holds for all convex chance-constrained optimization problems. The *support constraint* of problem (4.47) is a constraint whose removal changes the solution of (4.47). This problem is *fully supported*, since the number of support constraints is equal to the number of design variables n_{theta} . Therefore we can replace " \leq " with " $=$ ". and further simplify (4.50)

$$N = \left\lceil \frac{e}{\varepsilon_u (e - 1)} \left(\log \frac{1}{\delta_u} + n_\theta \right) \right\rceil. \quad (4.51)$$

When approximating $\mathcal{U}_{s,\theta}$ within a hyper-rectangular as (2.1), $n_\theta = 2m$, and the scenario optimization in (4.47) reduces to finding the extremal curves for $j = 1 \in \llbracket m \rrbracket$

$$\hat{\alpha}_j(s) := \min_{i=1, \dots, N} u_j^{(i)}(s) + e_j^-(s), \quad (4.52)$$

$$\hat{\beta}_j(s) := \max_{i=1, \dots, N} u_j^{(i)}(s) + e_j^+(s), \quad (4.53)$$

where $e_j^-(s) \leq 0 \leq e_j^+(s)$ are arbitrary safety margins in estimating $\mathcal{U}(s)$. For the rest of this paper, we consider $e_j^-(s) = e_j^+(s) = 0$ for all for $0 \leq s \leq t$, unless otherwise is stated. In this case, $\hat{\alpha}_j, \hat{\beta}_j$ are respectively the *sample minimum and maximum* along the j th control direction in the normal coordinates, at any given time. The inclusion

$$[\hat{\alpha}_j(s), \hat{\beta}_j(s)] \subseteq [\alpha_j(s), \beta_j(s)], \quad (4.54)$$

holds since the smallest *axis-aligned* hyperrectangle enclosing all the samples can be no bigger than the same enclosing all possible realizations of the vector $\mathbf{u}(s)$ for $0 \leq s \leq t$.

Using the sample estimates $[\hat{\alpha}_j(t), \hat{\beta}_j(t)]$, for $j = 1, \dots, m$, we compute the estimated reach set boundary $\partial \hat{\mathcal{X}}_t^\square$ in the normal coordinates using our analytical formula for the integrator reach sets. We then map that boundary via the known diffeomorphism τ^{-1} to compute $\partial \hat{\mathcal{Z}}_t$, an estimate for the boundary of the true compact reach set \mathcal{Z}_t . The question now becomes: what kind of guarantees can be given for such estimates?

In the Theorem 4.5 stated next, we show that (4.54), and the computational sequence mentioned above, allows us to construct the boundary $\partial \hat{\mathcal{Z}}_t$ of the compact set $\hat{\mathcal{Z}}_t$ such that we have the *deterministic under-approximation guarantee*: $\hat{\mathcal{Z}}_t \subseteq \mathcal{Z}_t$ at any time t .

Theorem 4.5. *Let $\mathcal{Z}_t(z_0, \mathcal{V})$ be the reach set of a feedback linearizable dynamic at time t , with state vector $z \in \mathbb{R}^d$ and compact input set $\mathcal{V} \subset \mathbb{R}^m$. There exist a diffeomorphism mapping of states, $\tau : \mathcal{R}_t \rightarrow \mathcal{X}_t^\square$ and the homeomorphism mapping of inputs, $\tau_u : \mathcal{V} \rightarrow \mathcal{U}(t)$, where \mathcal{X}_t^\square and \mathcal{R}_t are the reach sets in normal coordinates $\mathbf{x} \in \mathbb{R}^d$, and augmented states $\boldsymbol{\rho} \in \mathbb{R}^{n_w+d}$, respectively.*

At any time t , we can then guarantee

$$\hat{Z}_t(\mathbf{z}_0, \mathcal{V}) \subseteq Z_t(\mathbf{z}_0, \mathcal{V}),$$

where $\hat{Z}_t = \Pi_z(\hat{R}_t)$ and $\hat{R}_t = \tau^{-1}(\hat{\mathcal{X}}_t^\square)$. Here, Π_z is the projection operator into the z coordinates and $\hat{\mathcal{X}}_t^\square$ is the estimated integrator reach set using sample pairs $(\hat{\alpha}_j(t), \hat{\beta}_j(t))$ given in (4.53).

The proof of Theorem 4.5 is provided in Appendix D.5.

4.3.2 Probabilistic inclusion

As detailed in Ch. 4.3.1, we start with N sample trajectories bounded in the original input set \mathcal{V} and approximate $\mathcal{U}(t)$ within a norm ball, which we use to calculate the integrator reach set $\hat{\mathcal{X}}_t$. Employing the inverse diffeomorphism τ^{-1} , we then estimate the original reach set at time t , $\hat{Z}_t = \Pi_z(\tau^{-1}(\hat{\mathcal{X}}_t^\square))$. Assuming condition (4.48) holds, then we want to know ε_z, δ_z such that

$$\mathbb{P}(\mathbf{z}(t) \in \hat{Z}_t) > 1 - \varepsilon_z, \quad \text{with probability } 1 - \delta_z.$$

In other words, we are interested to know how the probabilistic inclusion changes during the following nonlinear transformations:

$$\underbrace{\hat{\mathcal{U}}_t}_{(\varepsilon_u, \delta_u)} \longrightarrow \underbrace{\hat{\mathcal{X}}_t^\square}_{(\varepsilon, \delta_x)} \longrightarrow \underbrace{\hat{\mathcal{R}}_t}_{(\varepsilon_\rho, \delta_\rho)} \longrightarrow \underbrace{\hat{Z}_t}_{(\varepsilon_z, \delta_z)}.$$

With $e_j^+(s) = e_j^-(s) = 0$ for all $j \in \llbracket m \rrbracket$ and $s \in [0, t]$, we know that for each trajectory $\mathbf{u}(s) \in \hat{\mathcal{U}}_s$, we have

$$\exp(t\mathbf{A})x_0 + \int_0^t \exp((t-s)\mathbf{A})\mathbf{b}\mathbf{u}(s)ds \in \hat{\mathcal{X}}_t^\square(\mathcal{U}_s).$$

However there might exist $\mathbf{x}(t) \in \hat{\mathcal{X}}_t$ resulting from $\mathbf{u}(s) \notin \hat{\mathcal{U}}_s$ for $s \in [0, t]$, therefore,

$$\mathbb{P}(\mathbf{x}(s) \in \hat{\mathcal{X}}_s^\square) \geq \mathbb{P}_u(\mathbf{u}(s) \in \hat{\mathcal{U}}(s),) \geq 1 - \varepsilon_u.$$

with probability of confidence $1 - \delta_u$.

Since the transformations $\hat{\mathcal{X}}_t \rightarrow \hat{\mathcal{R}}_t$ and $\hat{\mathcal{R}}_t \rightarrow \hat{\mathcal{Z}}_t$ are deterministic, we can conclude

$$(\varepsilon_u, \delta_u) = (\varepsilon_x, \delta_x) = (\varepsilon_\rho, \delta_\rho) = (\varepsilon_z, \delta_z).$$

In the following, we will employ the proposed learning approach to estimate the reach sets of example cases of static state feedback linearizable systems.

4.3.3 Example: learning the reach set of a static state feedback linearizable system with $\mathbf{r} = (3, 2)^\top$

Consider the 5-dimensional ($d = 5$) static state feedback linearizable system (1.11) with the relative degree vector $\mathbf{r} = (3, 2)^\top$, starting from the initial conditions $\mathbf{z}_0 = [0 \ 0 \ 0 \ 0 \ 0]$, and subject to convex compact set-valued uncertainty \mathcal{V} , a hyperrectangle defined as $\mathcal{V} := \begin{bmatrix} [-0.6, 0.6] & [-0.5, 0.5] \end{bmatrix}^\top$

We will follow the steps delineated in Ch.4.3 to learn the reach set of this system at $t = 1$ s and $t = 2$ s.

Setting $(\varepsilon, \delta) = (10^{-2}, 10^{-3})$ and $n_\theta = 2m = 4$ in (4.51), we get the required number of sample input trajectories, $N = 1410$.

Employing a constrained Gaussian process, we then generate the corresponding N random trajectory samples $\{\mathbf{v}^{(i)}(t)\}_{i=1}^N$ such that each $\{\mathbf{v}^{(i)}(t) \in \mathcal{V} \subset \mathbb{R}^2$. Since \mathcal{V} is a hyperrectangle in \mathbb{R}^2 , we use the truncated multivariate Gaussian distributions for generating the constrained Gaussian process samples to ensure that the range of the sampled functions remain in \mathcal{V} . We used the Metropolis-Hastings Markov Chain

Monte Carlo (MCMC) for sampling from the truncated multivariate Gaussian distributions.

We used MATLAB's ode45 to generate N trajectories $\{\mathbf{z}^{(i)}(t)\}_{i=1}^N$ in the original coordinates. Using input map τ_u described in (1.14b), we get the inputs in the integrator coordinates, $\{u_1^{(i)}(t)\}_{i=1}^N$ and $\{u_2^{(i)}(t)\}_{i=1}^N$.

The trajectories for the parameters $(\hat{\alpha}_j(t), \hat{\beta}_j(t))$, for $j = 1, 2$ are given by

$$\hat{\alpha}_j(t) := \min_{i=1, \dots, N} u_j^{(i)}(t), \quad \hat{\beta}_j(t) := \max_{i=1, \dots, N} u_j^{(i)}(t).$$

The initial condition in the integrator coordinates is given by $\mathbf{x}_0 = \tau(\mathbf{z}_0)$, where τ is defined in (1.14a).

Employing the parametric equation of the integrator boundary (3.9), we get the sample points $\{\mathbf{x}_1^{(i)}(t) \in \mathbb{R}^3\}_{i=1}^{391}$ and $\{\mathbf{x}_2^{(i)}(t) \in \mathbb{R}^2\}_{i=1}^{40}$ from $\partial\hat{\mathcal{X}}_1$ and $\partial\hat{\mathcal{X}}_2$, respectively. After taking the Minkowski sum of these sets we end up with $\{\mathbf{z}^{(i)}(t) \in \mathbb{R}^5\}_{i=1}^{15640}$ from $\partial\hat{\mathcal{X}}_t^\square$. Finally, we take the inverse mapping τ^{-1} given in (1.14a) to get $\{\hat{\mathbf{z}}^{(i)}(t) \in \mathbb{R}^5\}_{i=1}^{15640}$ from $\partial\hat{\mathcal{Z}}_t$. The reach set $\partial\hat{\mathcal{Z}}_t$ along with $\{\mathbf{z}^{(i)}(t)\}_{i=1}^N$ are shown in Fig. 4.5 using the projection of the 5-dimensional space onto each pair of coordinate axes at $t = 1$ s and $t = 2$ s. The serial computational time for estimating $\hat{\mathcal{Z}}_t$ was 0.94s and 1.13 for $t = 1$ s and $t = 2$ s, respectively, prior to optimization and parallelization. Faster-than-real-time computation is possible by parallelizing the learning algorithm in Ch. 4.3, which we cover in Ch. 4.6.

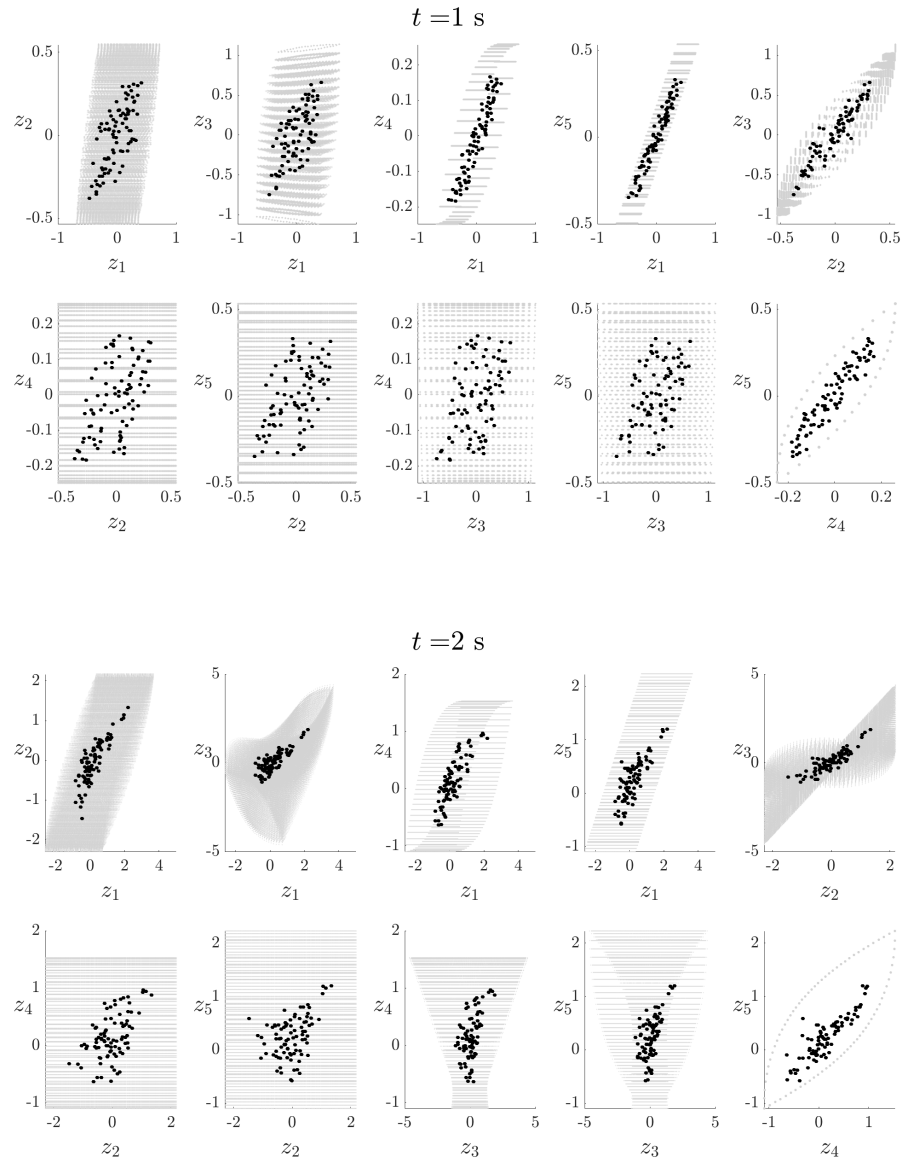


Figure 4.5: The reach set $\partial \hat{\mathcal{Z}}_t$ of system (sampled in blue) (1.11) along with sample trajectories, $\{\mathbf{z}^{(i)}(t)\}_{i=1}^N$ (black) are shown using the projection of the five-dimensional space onto each pair of coordinate axes at $t = 1$ s (top) and $t = 2$ s (bottom) (see (4.3.3) for more details).

4.3.4 Example: learning the reach set of a static state feedback linearizable system with $r = d = 4$

Following the exact same steps as (4.3.3), here, we estimate the reach set of the 4-dimensional nonlinear system given in (1.15) with initial condition $\mathbf{z}_0 = [0 \ 0 \ 0 \ 0]$ and input set $\mathcal{V} = [-1, 1]$. To satisfy the accuracy and confidence, $(\varepsilon, \delta) = (10^{-2}, 10^{-3})$, we generate the $N = 1410$ input trajectories, $\{v^{(i)}(t)\}_{i=1}^N$, using the constrained Gaussian process such that each $\{v^{(i)}(t) \in \mathcal{V} \subset \mathbb{R}\}$ (refer to Ch. 4.3.1 and Ch. 4.3.3 for more details). These input trajectories are fed to MATLAB's ode45 to propagate the state trajectories in the original coordinates, $\{\mathbf{z}^{(i)}(t)\}_{i=1}^N$. Using the input map τ_u described in (1.17b), we get the input trajectories in the integrator coordinates, $\{u^{(i)}(t)\}_{i=1}^N$. The trajectories for the parameters $(\hat{\alpha}(t), \hat{\beta}(t))$, are given by

$$\hat{\alpha}(t) := \min_{i=1, \dots, N} u^{(i)}(t), \quad \hat{\beta}(t) := \max_{i=1, \dots, N} u^{(i)}(t).$$

The initial condition in the integrator coordinates is given by $\mathbf{x}_0 = \tau(\mathbf{z}_0)$, where τ is defined in (1.17a). Employing the parametric equation of the integrator boundary (3.9), we get the sample points $\{\mathbf{x}^{(i)}(t) \in \mathbb{R}^4\}_{i=1}^{346}$ from $\partial \hat{\mathcal{X}}_t$.

In the last step, we take the inverse mapping τ^{-1} given in (1.17a) to obtain $\{\hat{\mathbf{z}}^{(i)}(t) \in \mathbb{R}^5\}_{i=1}^{346}$ from $\partial \hat{\mathcal{Z}}_t$. The reach set $\partial \hat{\mathcal{Z}}_t$ along with $\{\mathbf{z}^{(i)}(t)\}_{i=1}^N$ are shown in Fig. 4.6 using the projection of the five-dimensional space onto each pair of coordinate axes at $t = 1$ s and $t = 2$ s. The serial computational time for estimating $\hat{\mathcal{Z}}_t$ was 1.13 s and 1.20 s for $t = 1$ s and $t = 2$ s, respectively, prior to optimization and parallelization. We show in Ch. 4.6 how to parallelize the learning algorithm in Ch. 4.3 to achieve faster-than-real-time computation. In the following, we will employ the proposed learning approach to estimate the reach sets of dynamic feedback linearizable systems given in (1.18). In doing so, however, we will encounter the problem of calculating the bound-

ary of the Minkowski sum of $\mathcal{X}_t^\square(\{x_0\}) \subset \mathbb{R}^d$ with a line segment ℓ embedded in \mathbb{R}^d , which deserves its own separate subsection, which we now provide.

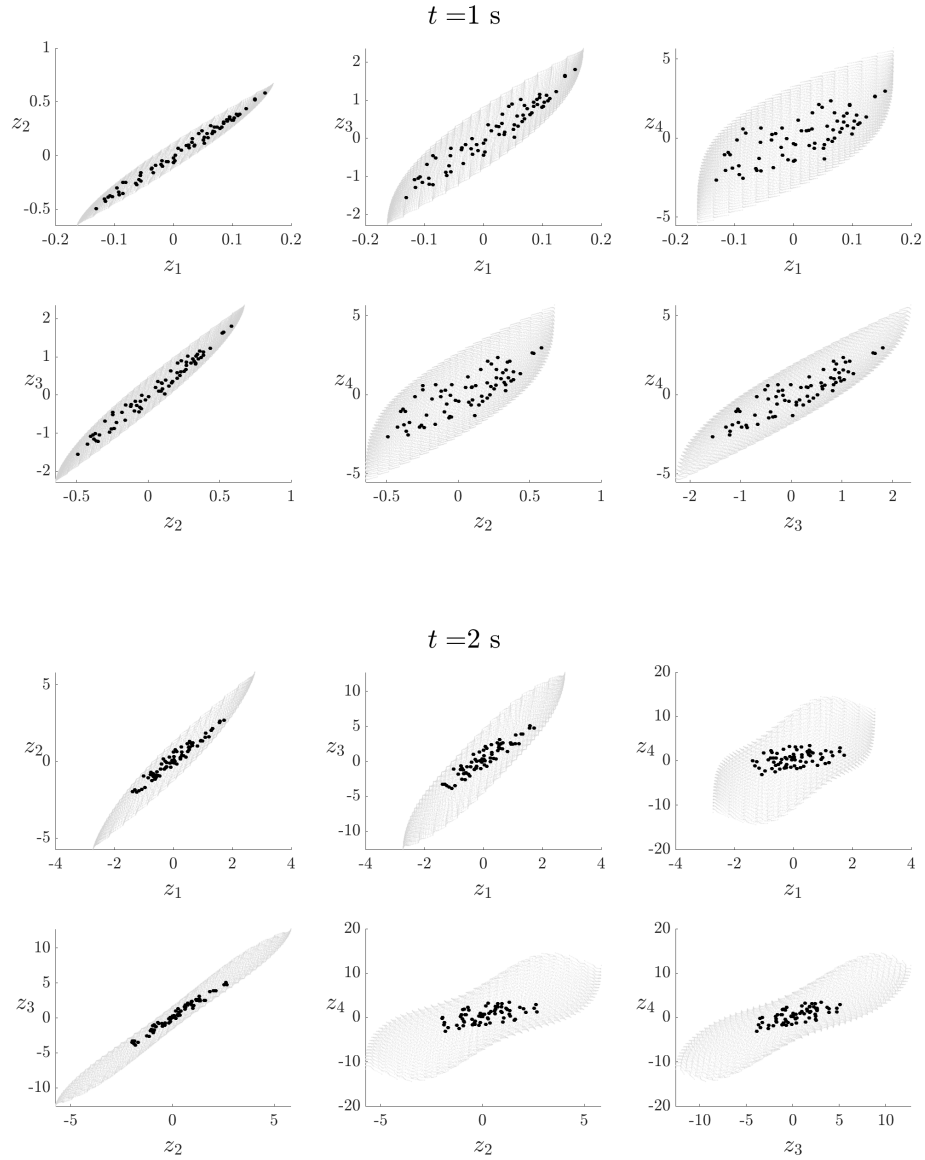


Figure 4.6: The reach set $\partial \hat{\mathcal{Z}}_t$ of system (sampled in blue) (1.15) along with sample trajectories, $\{z^{(i)}(t)\}_{i=1}^N$ (black) are shown using the projection of the five-dimensional space onto each pair of coordinate axes at $t = 1 \text{ s}$ (top) and $t = 2 \text{ s}$ (bottom) (see (4.3.4) for more details).

4.4 Boundary of the Minkowski Sum of \mathcal{X}_t^\square with a Line Segment

At a given time t , we want to investigate the boundary of the Minkowski sum of the set $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ with a line segment ℓ embedded in \mathbb{R}^d . Without loss of generality, for the purpose of computing this boundary, we consider a single input ($m = 1$) integrator reach set $\mathcal{X}_t \subset \mathbb{R}^d$.

Consider the parametric representation of the line segment

$$\ell(c) = \ell_0 + (\ell_1 - \ell_0)c, \quad 0 \leq c \leq 1, \quad (4.55)$$

where $\ell_0, \ell_1 \in \mathbb{R}^d$ represent the start and the end points of ℓ , respectively. Also, consider the unit vector

$$\widehat{\ell} := (\ell_1 - \ell_0) / \|\ell_1 - \ell_0\| = (\ell_1, \ell_2, \dots, \ell_d). \quad (4.56)$$

Let us define the parametric surface $\mathcal{S}(\boldsymbol{\sigma})$ as

$$\mathcal{S}(\boldsymbol{\sigma}) := \langle \widehat{\mathbf{n}}(\boldsymbol{\sigma}), \widehat{\ell} \rangle = 0, \quad \text{for } \boldsymbol{\sigma} \in \mathcal{W}_t, \quad (4.57)$$

where $\widehat{\mathbf{n}}$ represents the outward unit normal vector on $\partial\mathcal{X}_t$, \mathcal{W}_t is the Weyl chamber defined in (2.15) and \mathcal{W}_t^k is the degenerate set of parameter vectors defined in Remark 3.3. In the following, we derive the explicit form of $\mathcal{S}(\boldsymbol{\sigma}) = 0$. For this purpose, let the Levi-Civita permutation symbol

$$\varepsilon_{i_1, i_2, \dots, i_n}^{1, 2, \dots, n} := \begin{cases} 1 & \text{if } \{i_1, i_2, \dots, i_n\} \text{ is an even permutation of } \{1, 2, \dots, n\}, \\ -1 & \text{if } \{i_1, i_2, \dots, i_n\} \text{ is an odd permutation of } \{1, 2, \dots, n\}, \\ 0 & \text{otherwise.} \end{cases} \quad (4.58)$$

Also, for $r \in \mathbb{N}_0$, let p_r denote the r th elementary symmetric polynomials [136, Ch. 2.2] in variables $\sigma_1, \sigma_2, \dots, \sigma_{d-1}$, with the convention $e_0 \equiv 1$.

Theorem 4.6. *The parametric surface $\mathcal{S}(\boldsymbol{\sigma})$ defined in (4.57), is given by*

$$\{\boldsymbol{\sigma} \in \mathcal{W}_t \mid \sum_{i=1}^d (d-i)! (-1)^{d-i} \ell_i \mathbf{p}_{i-1} = 0\}. \quad (4.59)$$

The second summation in the RHS is over all of the different permutations of indices i_1, i_2, \dots, i_{d-1} .

The proof of Theorem 4.6 in Appendix D.6

Remark 4.6. *Theorem 4.6 tells that $\mathcal{S}(\boldsymbol{\sigma})$ is the intersection of $\mathcal{W}_t \setminus \mathcal{W}_t^{\text{dgen}}$ with the set*

$$\left\{ \boldsymbol{\sigma} \in \mathbb{R}^{d-1} \mid \sum_{i=1}^d (d-i)! (-1)^{d-i} \ell_i e_{i-1} = 0 \right\}, \quad (4.60)$$

and hence \mathcal{S} defines a semialgebraic set. In other words, \mathcal{S} is the intersection of half-spaces $0 \leq \sigma_1 < \sigma_2 < \dots < \sigma_{d-1} \leq t$ with the affine variety (4.60), the latter given by the set of zeros of certain linear combination of the elementary symmetric polynomials $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{d-1}$. For instance, when $d = 3$, (4.60) reduces to $\{(\sigma_1, \sigma_2) \in \mathbb{R}^2 \mid 2\ell_1 - \ell_2(\sigma_1 + \sigma_2) + \ell_3\sigma_1\sigma_2 = 0\}$. Likewise, when $d = 4$, (4.60) reduces to $\{(\sigma_1, \sigma_2, \sigma_3) \in \mathbb{R}^3 \mid -6\ell_1 + 2\ell_2(\sigma_1 + \sigma_2 + \sigma_3) - \ell_3(\sigma_1\sigma_2 + \sigma_2\sigma_3 + \sigma_3\sigma_1) + \ell_4\sigma_1\sigma_2\sigma_3 = 0\}$.

A consequence of Theorem 4.6 is that the parametric surface $\mathcal{S}(\boldsymbol{\sigma})$ divides the parameter space \mathcal{W}_t into two parts: \mathcal{S}_0 and \mathcal{S}_1 , such that

$$\mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S} = \mathcal{W}_t. \quad (4.61)$$

We can identify these two sets by assuming \mathcal{S}_0 is the part that contains

$$\{\boldsymbol{\sigma} \in \mathcal{W}_t \mid \sigma_1 = \sigma_2 = \dots = \sigma_{d-1}\}.$$

Let us denote the boundaries of \mathcal{S}_0 and \mathcal{S}_1 by $\partial\mathcal{S}_0$ and $\partial\mathcal{S}_1$, respectively. Let Δ denote the symmetric difference of two sets. Recall from chapter 3 that each $\boldsymbol{\sigma} \in \mathcal{W}_t$ assigns a pair of points $(\mathbf{x}^{\text{upper}}, \mathbf{x}^{\text{lower}})$ on $\partial\mathcal{X}_t$, one on each bounding hypersurfaces. Now

introduce

$$\partial\mathcal{X}_{\text{cut}} := \{\mathbf{x}^{\text{upper}}(\boldsymbol{\sigma}) \mid \boldsymbol{\sigma} \in \partial\mathcal{S}_1\} \Delta \{\mathbf{x}^{\text{lower}}(\boldsymbol{\sigma}) \mid \boldsymbol{\sigma} \in \partial\mathcal{S}_0\}. \quad (4.62)$$

The set $\partial\mathcal{X}_{\text{cut}}$ divides the integrator boundary $\partial\mathcal{X}$, into two sets: $\partial\mathcal{X}_\geq$ and $\partial\mathcal{X}_\leq$, i.e.,

$$\partial\mathcal{X}_\geq \cup \partial\mathcal{X}_\leq \cup \partial\mathcal{X}_{\text{cut}} = \partial\mathcal{X}_t, \quad (4.63)$$

where

$$\partial\mathcal{X}_\geq := \{\mathbf{x}^{\text{bdy}}(\boldsymbol{\sigma}) \in \partial\mathcal{X}_t \mid \langle \widehat{\mathbf{n}}(\boldsymbol{\sigma}), \widehat{\boldsymbol{\ell}} \rangle \geq 0\}, \quad (4.64a)$$

$$\partial\mathcal{X}_\leq := \{\mathbf{x}^{\text{bdy}}(\boldsymbol{\sigma}) \in \partial\mathcal{X}_t \mid \langle \widehat{\mathbf{n}}(\boldsymbol{\sigma}), \widehat{\boldsymbol{\ell}} \rangle \leq 0\}. \quad (4.64b)$$

Remark 4.7. *If there exist no $\boldsymbol{\sigma} \in \mathcal{W}_t$ such that (4.57) holds, i.e., if*

$$\{\boldsymbol{\sigma} \in \mathcal{W}_t \mid \langle \widehat{\mathbf{n}}(\boldsymbol{\sigma}), \widehat{\boldsymbol{\ell}} \rangle = 0\} = \emptyset,$$

then \mathcal{S}_1 is also an empty set. In this case, the sets $\partial\mathcal{X}_\geq$ and $\partial\mathcal{X}_\leq$ are simply $\partial\mathcal{X}^{\text{upper}}$ and $\partial\mathcal{X}^{\text{lower}}$ respectively, or vice versa, depending on the criteria (4.64).

In the following Theorem, we derive a parametric formula for the boundary of the Minkowski sum of the reach set \mathcal{X}_t at time t with a line segment $\boldsymbol{\ell}$ given by (4.55). In other words, we deduce an expression for the boundary $\partial(\boldsymbol{\ell} + \mathcal{X}_t)(c, \boldsymbol{\sigma})$ in terms of the parameter tuple $(c, \boldsymbol{\sigma}) \in [0, 1] \times \mathcal{W}_t$, which depends on time t because \mathcal{W}_t depends on the same, by definition (see (2.15)).

Theorem 4.7. *For a compact input set $\mathcal{U}(t) \subset \mathbb{R}$, consider the d dimensional integrator reach set boundary $\partial\mathcal{X}_t$ at time t , with single input $u(t) \in \mathcal{U}(t)$ and initial condition $\mathbf{x}_0 \in \mathbb{R}^d$. At any time t , the boundary of the Minkowski sum of the reach set \mathcal{X} with the line segment $\boldsymbol{\ell}$ in (4.55), is given by*

$$\partial(\boldsymbol{\ell} + \mathcal{X}_t)(c, \boldsymbol{\sigma}) = \{\boldsymbol{\ell}(c) + \partial\mathcal{X}_{\text{cut}}(\boldsymbol{\sigma})\} \cup \{\boldsymbol{\ell}_1 + \partial\mathcal{X}_\geq\} \cup \{\boldsymbol{\ell}_0 + \partial\mathcal{X}_\leq\}, \quad (4.65)$$

where $0 \leq c \leq 1$, and ℓ_0 and ℓ_1 are the start and the end points of ℓ , respectively. The parameter vector $\sigma \in \mathbb{R}^{d-1}$ belongs to the set (2.15). The sets $\partial\mathcal{X}_{\text{cut}}$, $\partial\mathcal{X}_\geq$, $\partial\mathcal{X}_\leq$ are defined in (4.62) and (4.64), respectively.

The proof for Theorem 4.7 is provide in Appendix D.7.

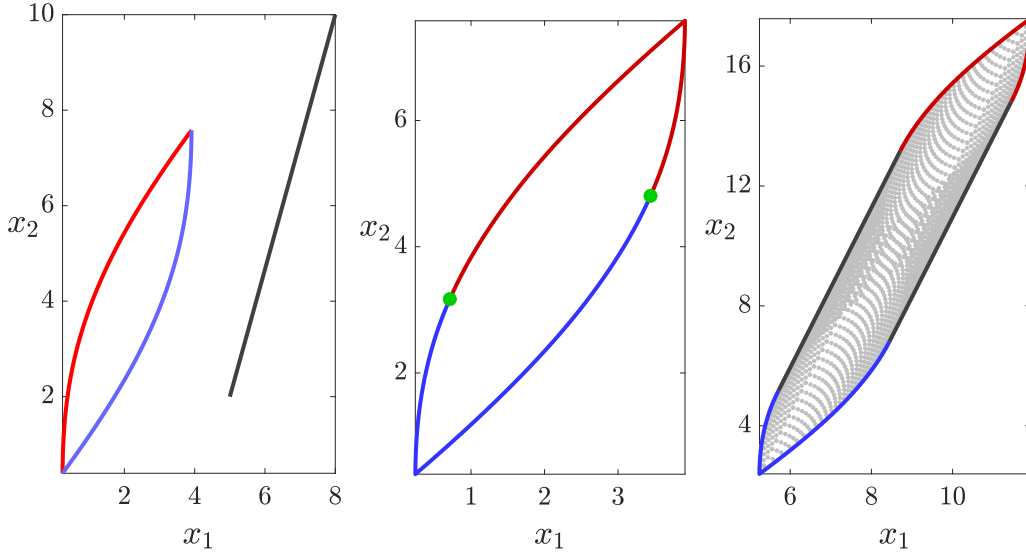


Figure 4.7: (left): This figure shows the integrator reach set $\mathcal{X} \in \mathbb{R}^2$ given in (4.66) along with the line segment (solid dark grey) described in (4.67). The upper surface, $\partial\mathcal{X}^{\text{upper}}$ and the lower surface, $\partial\mathcal{X}^{\text{lower}}$ are shown in light red and light blue, respectively. (middle) The integrator surface, $\partial\mathcal{X}$ is divided into 3 subsets: $\partial\mathcal{X}_\geq$ (red), $\partial\mathcal{X}_\leq$ (blue) and $\partial\mathcal{X}_{\text{cut}}$ (dotted green), such that $\partial\mathcal{X}_\geq \cup \partial\mathcal{X}_\leq \cup \partial\mathcal{X}_{\text{cut}} = \partial\mathcal{X}$. (right): The Minkowski sum of the integrator reach set \mathcal{X} in (4.66) and ℓ in (4.67) using the algorithm provided in (4.65): $\partial(\ell + \mathcal{X})(c, \sigma) = \{\ell(c) + \partial\mathcal{X}_{\text{cut}}(\sigma)\} \cup \{\ell_1 + \partial\mathcal{X}_\geq\} \cup \{\ell_0 + \partial\mathcal{X}_\leq\}$, while $\{\ell(c) + \partial\mathcal{X}_{\text{cut}}(\sigma)\}$ is shown by solid grey, $\{\ell_1 + \partial\mathcal{X}_\geq\}$ in red and $\{\ell_0 + \partial\mathcal{X}_\leq\}$ in blue. For verification, the sample based Mikowski sum of $\partial\mathcal{X}$ and ℓ is shown in dotted light grey.

4.4.1 Example: boundary of the Minkowski sum of $\mathcal{X}_t(\{x_0\})$ with a line segment embedded in 2D space

We exhibit the application of the proposed algorithm in Ch. 4.4 by finding the boundary of the Minkowski sum of a line segment ℓ with a single input integrator reach set $\mathcal{X}_t \subset \mathbb{R}^2$ at $t = 1$ s, starting from initial states $x_0 = [0 \ 0]$. The parametric formula given

in (3.9) provides the parametric equations for the integrator's upper surface, $\partial\mathcal{X}^{\text{upper}}$, and the lower surface, $\partial\mathcal{X}^{\text{lower}}$ as follows:

$$\begin{aligned}\mathbf{x}^{\text{upper}}(\sigma) &= \int_0^t \nu(s)\boldsymbol{\xi}(t-s)ds + \int_0^\sigma \mu(s)\boldsymbol{\xi}(t-s)ds - \int_\sigma^t \mu(s)\boldsymbol{\xi}(t-s)ds \quad (4.66) \\ \mathbf{x}^{\text{lower}}(\sigma) &= \int_0^t \nu(s)\boldsymbol{\xi}(t-s)ds - \int_0^\sigma \mu(s)\boldsymbol{\xi}(t-s)ds + \int_\sigma^t \mu(s)\boldsymbol{\xi}(t-s)ds\end{aligned}$$

where $\boldsymbol{\xi}(s) = \begin{bmatrix} s & 1 \end{bmatrix}^\top$.

The line segment ℓ is defined in (4.55), while the parameters ℓ_0, ℓ_1 and the tangent vector $\widehat{\ell}$ are given by

$$\ell_0 = \begin{bmatrix} 5 & 8 \end{bmatrix}, \quad \ell_1 = \begin{bmatrix} 2 & 10 \end{bmatrix}, \quad \widehat{\ell} = \begin{bmatrix} 3 & 8 \end{bmatrix}. \quad (4.67)$$

From (4.59), we obtain the parametric equation of the set σ as

$$\mathcal{S}(\sigma) := \left\{ \sigma \in \mathcal{W}_t \subset \mathbb{R} \mid \sigma\ell_2 - \ell_1 = 0 \right\}. \quad (4.68)$$

So for the 2D integrator, the set σ is either singleton, or empty, such that

$$\mathcal{S}(\sigma) := \left\{ \frac{\ell_1}{\ell_2} \right\}, \quad \text{if } 0 \leq \frac{\ell_1}{\ell_2} \leq t, \quad (4.69)$$

$$\mathcal{S}(\sigma) := \emptyset, \quad \text{otherwise.}$$

From (4.67) and (4.69) we get $\sigma = \{0.375\}$, $\mathcal{S}_0 := [0, 0.375)$ and $\mathcal{S}_1 := (0.375, t]$.

Having \mathcal{S}_0 and \mathcal{S}_1 , we are now able to construct the sets $\{\mathbf{x}^{\text{upper}}(\sigma) \mid \sigma \in \partial\mathcal{S}_1\}$, $\{\mathbf{x}^{\text{lower}}(\sigma) \mid \sigma \in \partial\mathcal{S}_0\}$ and $\partial\mathcal{X}_{\text{cut}}$ given in (4.62), as delineated in Fig. 4.7.

Since $\{\mathbf{x}^{\text{upper}}(\sigma) \mid \sigma \in \partial\mathcal{S}_1\} \cap \{\mathbf{x}^{\text{lower}}(\sigma) \mid \sigma \in \partial\mathcal{S}_0\} = \emptyset$, we have $\partial\mathcal{X}_\geq = \{\mathbf{x}^{\text{upper}}(\sigma) \mid \sigma \in \partial\mathcal{S}_1\}$ and $\partial\mathcal{X}_\leq = \{\mathbf{x}^{\text{lower}}(\sigma) \mid \sigma \in \partial\mathcal{S}_0\}$, or vice versa, depending on criteria (4.64).

Next, identifying $\partial\mathcal{X}_\geq, \partial\mathcal{X}_\leq$ and $\partial\mathcal{X}_{\text{cut}}$ using criteria (4.64), we can utilize Theorem 4.7 to get the boundary of the Minkowski sum of \mathcal{X} in (4.66) with ℓ in (4.67). The resulting set, $\partial(\ell + \mathcal{X})$ along with $\{\ell + \mathcal{X}_t\}$ is depicted in Fig. 4.7.

4.4.2 Example: Boundary of the Minkowski sum of $\mathcal{X}_t(\{\mathbf{x}_0\})$ with a line segment embedded in 3D space

We employ our proposed algorithm in Ch. 4.4 to find the boundary of the Minkowski sum of a line segment ℓ with a single input integrator reach set $\mathcal{X}_t \subset \mathbb{R}^3$ at $t = 1$ s, starting from initial states $\mathbf{x}_0 = [0 \ 0 \ 0]$, with $u(t) \in [-0.5, 1.5]$. From (3.9), we write the parametric equations for the integrator's upper surface, $\partial\mathcal{X}^{\text{upper}}$, and the lower surface, $\partial\mathcal{X}^{\text{lower}}$, as follows:

$$\mathbf{x}^{\text{upper}}(\boldsymbol{\sigma}) = \int_0^t \nu(s) \boldsymbol{\xi}(t-s) ds + \int_0^{\sigma_1} \mu(s) \boldsymbol{\xi}(t-s) ds - \int_{\sigma_1}^{\sigma_2} \mu(s) \boldsymbol{\xi}(t-s) ds + \int_{\sigma_2}^t \mu(s) \boldsymbol{\xi}(t-s) ds, \quad (4.70)$$

$$\mathbf{x}^{\text{lower}}(\boldsymbol{\sigma}) = \int_0^t \nu(s) \boldsymbol{\xi}(t-s) ds - \int_0^{\sigma_1} \mu(s) \boldsymbol{\xi}(t-s) ds + \int_{\sigma_1}^{\sigma_2} \mu(s) \boldsymbol{\xi}(t-s) ds - \int_{\sigma_2}^t \mu(s) \boldsymbol{\xi}(t-s) ds,$$

where $\boldsymbol{\xi}(s) = \begin{bmatrix} \frac{s^2}{2} & s & 1 \end{bmatrix}^\top$. Let us define the line segment ℓ as in (4.55), with parameters ℓ_0 and ℓ_1 and the tangent vector $\widehat{\ell}$ given by

$$\ell_0 = \begin{bmatrix} -0.5 & -0.54 & -0.80 \end{bmatrix}, \ell_1 = \begin{bmatrix} -0.24 & 0.22 & 0.67 \end{bmatrix}, \widehat{\ell} = \begin{bmatrix} 0.16 & 0.45 & 0.88 \end{bmatrix}. \quad (4.71)$$

From (4.59), we obtain the parametric equation of surface \mathcal{S} as

$$\mathcal{S}(\boldsymbol{\sigma}) := \left\{ \boldsymbol{\sigma} \in \mathcal{W}_t \subset \mathbb{R}^2 \mid (\sigma_1 - \sigma_2) \ell_1 - \left(\frac{\sigma_1^2}{2} - \frac{\sigma_2^2}{2} \right) \ell_2 + \left(\frac{\sigma_1^2 \sigma_2}{2} - \frac{\sigma_2^2 \sigma_1}{2} \right) \ell_3 = 0 \right\}. \quad (4.72)$$

In Fig. 4.8, we depict the wedge-shape parameter space \mathcal{W}_t given in (2.15) and observe that $\mathcal{S}(\boldsymbol{\sigma})$ generates two subsets in \mathcal{S} : \mathcal{S}_0 and \mathcal{S}_1 , such that $\mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S} = \mathcal{W}_t$.

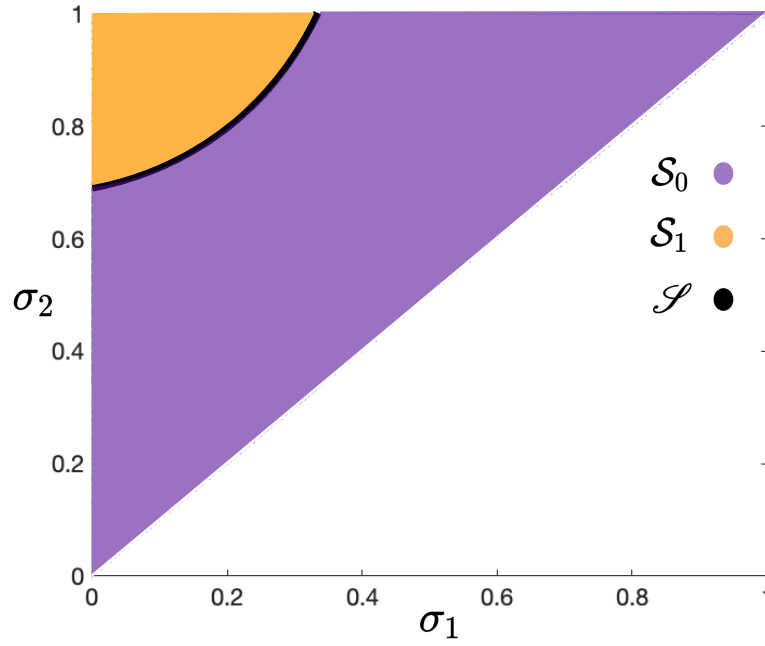


Figure 4.8: The parameter space (2.15) for $\mathcal{X}_t \in \mathbb{R}^3$ and $\ell \in \mathbb{R}^2$, given in (4.70) and (4.71), respectively. The parametric set $\mathcal{S}(\sigma)$, given in (4.72), is shown by a *dotted black* line, which divides the parameter space of \mathcal{X} into two parts: \mathcal{S}_0 (*dotted purple*) and \mathcal{S}_1 (*dotted orange*), such that $\mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S} = \mathcal{S}$.

Having \mathcal{S}_0 and \mathcal{S}_1 , we are now able to construct the sets $\{\mathbf{x}^{\text{upper}}(\sigma) \mid \sigma \in \partial\mathcal{S}_1\}$ and $\{\mathbf{x}^{\text{lower}}(\sigma) \mid \sigma \in \partial\mathcal{S}_0\}$ as delineated in Fig. 4.9. The next step is finding \mathcal{X}_{cut} using (4.62) as demonstrated in Fig. 4.10. As pointed out in (4.63), the set \mathcal{X}_{cut} , divides the integrator boundary into two disjointed parts, such that $\partial\mathcal{X}_{\geq} \cup \partial\mathcal{X}_{\leq} \cup \partial\mathcal{X}_{\text{cut}} = \partial\mathcal{X}_t$ (see Fig. 4.10a).

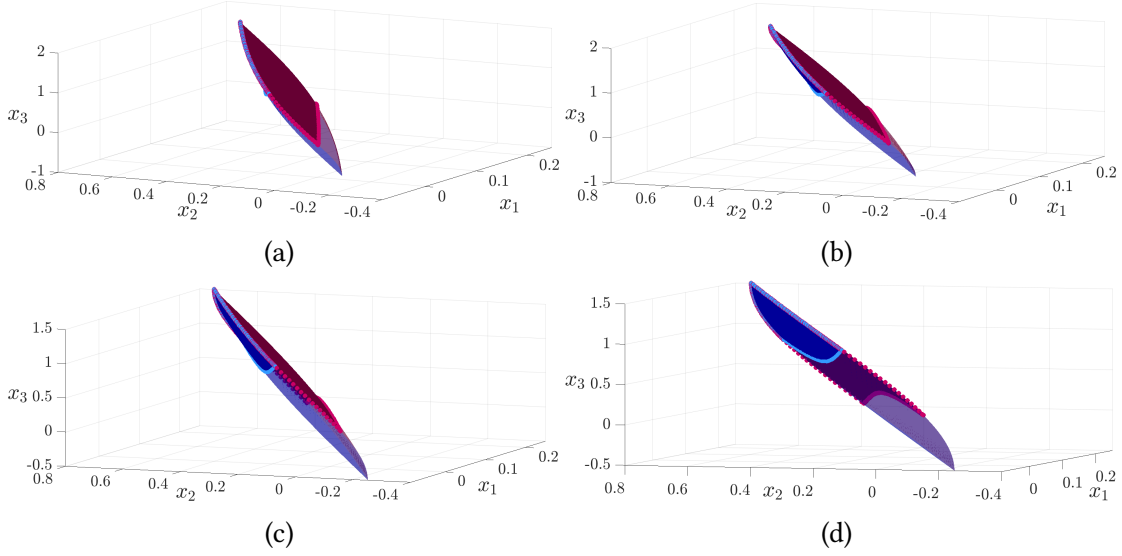


Figure 4.9: Four different views of the sets $\{\mathbf{x}^{\text{upper}}(\boldsymbol{\sigma}) \mid \boldsymbol{\sigma} \in \partial\mathcal{S}_1\}$ in *red*, and $\{\mathbf{x}^{\text{lower}}(\boldsymbol{\sigma}) \mid \boldsymbol{\sigma} \in \partial\mathcal{S}_0\}$ in *blue*. Each view is slightly rotated relative to the previous shot to provide a consistent 3D spatial view. The boundaries of these sets, $\partial(\{\mathbf{x}^{\text{upper}}(\boldsymbol{\sigma}) \mid \boldsymbol{\sigma} \in \partial\mathcal{S}_1\})$ and $\partial(\{\mathbf{x}^{\text{lower}}(\boldsymbol{\sigma}) \mid \boldsymbol{\sigma} \in \partial\mathcal{S}_0\})$ are depicted in *dotted light red* and *dotted light blue*, respectively. To provide more geometric intuition on the relative location of these sets, $\mathcal{X}^{\text{upper}}$ and $\mathcal{X}^{\text{lower}}$ are also shown in *transparent red* and *transparent blue*, respectively.

Finally, identifying $\partial\mathcal{X}_\geq$, $\partial\mathcal{X}_\leq$ and $\partial\mathcal{X}_{\text{cut}}$, we can utilize Theorem 4.7 to get the boundary of the Minkowski sum of \mathcal{X} in (4.70) with ℓ in (4.71). The resulting set is depicted in Fig. 4.10b.

4.4.3 Example: learning the reach set of a dynamic state feedback linearizable system

Consider the 4-dimensional ($d = 4$) dynamic state feedback linearizable system (1.18), starting from the initial conditions $\mathbf{z}_0 = [5 \ 0 \ 0 \ 0]$, and subject to convex compact set-valued uncertainty \mathcal{V} , a hyperrectangle defined as $\mathcal{V} := \begin{bmatrix} [-1, 1] & [-2, 2] \end{bmatrix}^\top$.

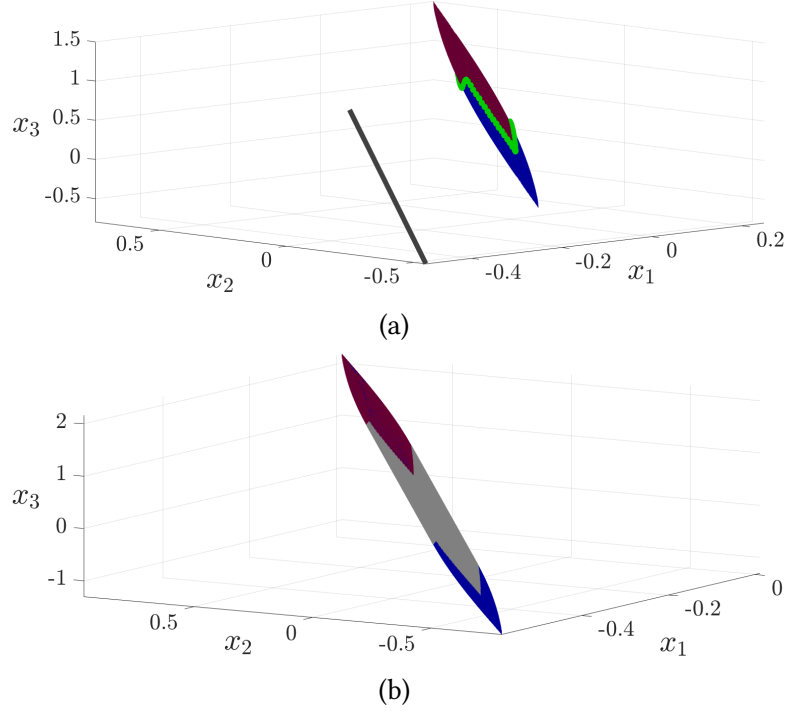


Figure 4.10: (a): This figure shows the integrator reach set $\mathcal{X} \in \mathbb{R}^3$ given in (4.70) along with the line segment (solid dark grey) described in (4.71). The integrator surface, $\partial\mathcal{X}$ is divided into 3 subsets: $\partial\mathcal{X}_\geq$ (red), $\partial\mathcal{X}_\leq$ (blue) and $\partial\mathcal{X}_{\text{cut}}$ (dotted green), such that $\partial\mathcal{X}_\geq \cup \partial\mathcal{X}_\leq \cup \partial\mathcal{X}_{\text{cut}} = \partial\mathcal{X}$ (b): The Minkowski sum of the integrator reach set \mathcal{X} in (4.70) and ℓ in (4.71) using the algorithm provided in (4.65): $\partial(\ell + \mathcal{X})(c, \sigma) = \{\ell(c) + \partial\mathcal{X}_{\text{cut}}(\sigma)\} \cup \{\ell_1 + \partial\mathcal{X}_\geq\} \cup \{\ell_0 + \partial\mathcal{X}_\leq\}$, while $\{\ell(c) + \partial\mathcal{X}_{\text{cut}}(\sigma)\}$ is shown by solid grey, $\{\ell_1 + \partial\mathcal{X}_\geq\}$ in red and $\{\ell_0 + \partial\mathcal{X}_\leq\}$ in blue.

We will follow the steps presented in Ch.4.3 to learn the reach set of this system at $t = 1$ s and $t = 2$ s.

Setting $(\varepsilon, \delta) = (10^{-2}, 10^{-3})$ and $n_\theta = 2m = 4$ in (4.51), we get the required number of sample input trajectories, $N = 1410$.

Employing a constrained Gaussian process, we then generate the corresponding N random trajectory samples $\{\mathbf{v}^{(i)}(t)\}_{i=1}^N$ such that each $\{\mathbf{v}^{(i)}(t) \in \mathcal{V} \subset \mathbb{R}^2\}$. Since \mathcal{V} is a hyperrectangle in \mathbb{R}^2 , we use the truncated multivariate Gaussian distributions for generating the constrained Gaussian process samples to ensure that the range of the sampled functions remain in \mathcal{V} . We used the Metropolis-Hastings Markov Chain Monte Carlo (MCMC) for sampling from the truncated multivariate Gaussian distri-

butions.

We used MATLAB's ode45 to generate N trajectories $\{z^{(i)}(t)\}_{i=1}^N$ in the original coordinates. Using input map $\tau_u(z, w)$ described in (1.14b), we get the inputs in the integrator coordinates, $\{u_1^{(i)}(t)\}_{i=1}^N$ and $\{u_2^{(i)}(t)\}_{i=1}^N$, where w is the compensator variable (see Ch. 1.6.1 and Ch. 1.6.1).

The trajectories for the parameters $(\hat{\alpha}_j(t), \hat{\beta}_j(t))$, for $j = 1, 2$ are estimated by

$$\hat{\alpha}_j(t) := \min_{i=1, \dots, N} u_j^{(i)}(t), \quad \hat{\beta}_j(t) := \max_{i=1, \dots, N} u_j^{(i)}(t).$$

The state mapping $\tau(z, w)$ in (1.18) is a function of the compensator variable w , while w is a function of the set valued uncertainty, \mathcal{V} , a rectangular. As a consequence, for $j = 1, 2$, the initial conditions in the integrator coordinates must be an interval instead of a single point, i.e., \mathcal{X}_{01} and \mathcal{X}_{02} are

$$\mathcal{X}_{01} = \begin{bmatrix} z_{01} \\ w \end{bmatrix} = \begin{bmatrix} z_{01} \\ z_{02} - [-1, 1] \end{bmatrix}, \quad \mathcal{X}_{02} = \begin{bmatrix} z_{01} + z_{03} \\ z_{02} \\ z_{04}(z_{02} - w) \end{bmatrix} = \begin{bmatrix} z_{01} + z_{03} \\ z_{02} \\ z_{04}[-2, 2] \end{bmatrix}. \quad (4.73)$$

As a result, the set valued matrix-vector product $\exp(\mathbf{A}_j t) \mathcal{X}_{j0}$ will return a tilted line segment, ℓ_j , embedded in \mathbb{R}^{r_j} for $j = 1, 2$, given by

$$\ell_1 = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_{01} \\ z_{02} - [-2, 2] \end{bmatrix} = \begin{bmatrix} z_{01} + tz_{02} \\ z_{02} \end{bmatrix} + \begin{bmatrix} -t \\ -1 \end{bmatrix} [-1, 1], \quad (4.74)$$

$$\ell_2 = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_{01} + z_{03} \\ z_{02} \\ z_{04}[-2, 2] \end{bmatrix} = \begin{bmatrix} z_{01} + z_{03} + tz_{02} \\ z_{02} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}t^2 z_{04} \\ tz_{04} \\ z_{04} \end{bmatrix} [-2, 2].$$

Assuming zero initial condition (see Remark 2.1), we first use the parametric equation

of the integrator boundary (3.9) to get $\mathcal{X}_{1t}(\{\mathbf{0}\})$ and $\mathcal{X}_{2t}(\{\mathbf{0}\})$. Then, utilizing algorithm (4.65) detailed in Ch. 4.4, we obtain the boundary of the Minkowski summation of $\mathcal{X}_{jt}(\{\mathbf{0}\})$ with ℓ_j , $\partial(\ell_j \dot{+} \mathcal{X}_{jt}(\{\mathbf{0}\}))$, for $j = 1, 2$.

In this example, the sets of points $\{\mathbf{x}_1^{(i)}(t) \in \mathbb{R}^2\}_{i=1}^{126}$ and $\{\mathbf{x}_2^{(i)}(t) \in \mathbb{R}^3\}_{i=1}^{3561}$ are sampled from $\partial\hat{\mathcal{X}}_1$ and $\partial\hat{\mathcal{X}}_2$, respectively.

After taking the Minkowski sum of these sets, we end up with $\{\mathbf{x}^{(i)}(t) \in \mathbb{R}^5\}_{i=1}^{448686}$ from $\partial\hat{\mathcal{X}}_t$.

Finally, we take the inverse mapping τ^{-1} given in (1.21a) to obtain $\{\hat{\mathbf{z}}^{(i)}(t) \in \mathbb{R}^5\}_{i=1}^{15640} \in \partial\hat{\mathcal{Z}}_t$. The estimated reach set in the original coordinate, $\partial\hat{\mathcal{Z}}_t$ along with $\{\mathbf{z}^{(i)}(t)\}_{i=1}^N$ are shown in Fig. 4.11 using the projection of the five-dimensional space onto each pair of coordinate axes at $t = 1$ s and $t = 2$ s. The serial computational time for estimating $\hat{\mathcal{Z}}_t$ was 1.45 s and 3.55 s for $t = 1$ s and $t = 2$ s, respectively, prior to optimization and parallelization. We will show in Ch. 4.6 that our algorithm is highly parallelizable, promising faster-than-real-time computation.

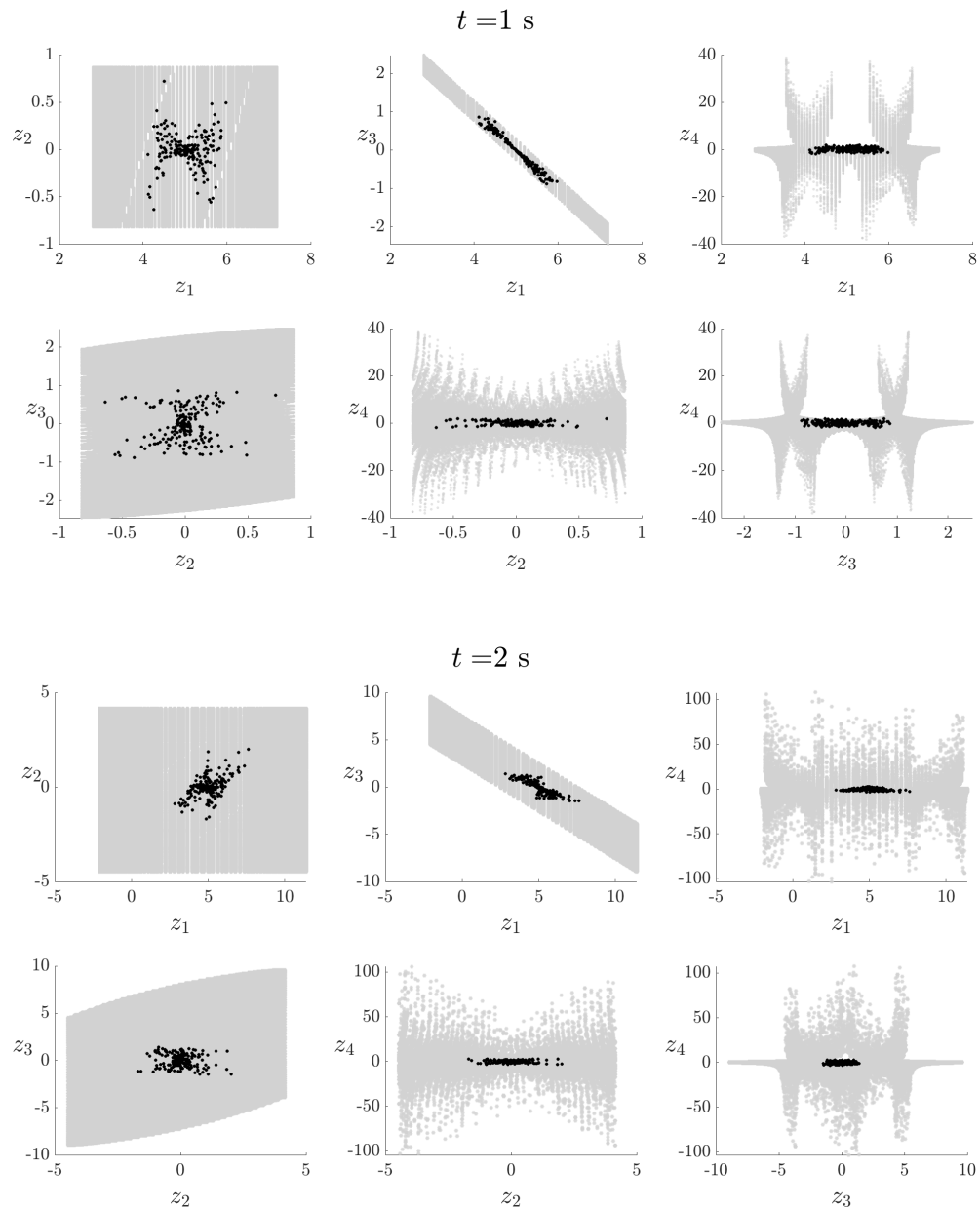


Figure 4.11: The reach set $\partial\hat{\mathcal{Z}}_t$ of system (1.18) (sampled in blue) along with sample trajectories, $\{z^{(i)}(t)\}_{i=1}^N$ (black) are shown using the projection of the five-dimensional space onto each pair of coordinate axes at $t = 1$ s (top) and $t = 2$ s (bottom) (see (4.3.3) for more details).

4.5 A Note on Monte Carlo vs Feedback Linearization

The objective of this section is to highlight the advantages of utilizing the feedback linearizable property for estimating the boundary of the reach set \mathcal{Z}_t compared to the direct Monte Carlo method [38]. Firstly, we demonstrate how the former enables the capture of the nonconvex characteristics inherent in the reach set. Furthermore, we discuss how the implementation of the feedback linearizable property offers the advantage of generating a broader range of training data for a data-driven support function learning algorithm that has been introduced in [137].

Monte Carlo

In this approach, using sample input trajectories $\{\mathbf{v}_i(t)\}_{i=1}^N$, we propagate the dynamics for $\{\mathbf{z}_i(s)\}_{i=1}^N$ using (1.5). Next we need to estimate the boundary points of \mathcal{Z}_t . One may achieve this by simply finding the vertices of the convex hull of $\{\mathbf{z}_i^{\text{bdy}}(s)\}_{i=1}^{N_y}$. However, without prior knowledge on sample distribution, vertex representation of the convex hull can have unbounded complexity [138]. Instead, by employing the support function formula (1.23), the boundary points can be simply represented as the support of the convex hull of the set $\{\mathbf{z}_i(s)\}_{i=1}^{N_y}$ in arbitrary unit directions $\{\mathbf{y}_i\}_{i=1}^{N_y} \in \mathbb{S}^{d-1}$

$$\mathbf{z}_i^{\text{bdy}} = \underset{\mathbf{z} \in \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N\}}{\operatorname{argmax}} \langle \mathbf{y}_i, \mathbf{z} \rangle, \quad i \in \llbracket N_y \rrbracket. \quad (4.75)$$

Feedback linearization property

Here we generate the boundary points by taking the following steps.

Step 1: Employing state diffeomorphism $\boldsymbol{\tau}$ we generate $\{\mathbf{x}_i(t)\}_{i=1}^N$.

Step 2: As in Ch. 4.5 using sample unit directions, $\{\mathbf{y}_i\}_{i=1}^{N_y} \in \mathbb{S}^{d-1}$, the support of the convex hull of $\{\mathbf{x}_i(t)\}_{i=1}^N$ is given by

$$\mathbf{x}_i^{\text{bdy}} = \underset{\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}}{\operatorname{argmax}} \langle \mathbf{y}_i, \mathbf{x} \rangle, \quad i \in \llbracket N_y \rrbracket. \quad (4.76)$$

Step 3: The estimated boundary points of \mathcal{Z}_t are achieved using the inverse diffeomorphism $\mathbf{z}_i^{\text{bdy}} = \boldsymbol{\tau}^{-1}(\mathbf{x}_i^{\text{bdy}})$, for $i \in \llbracket N_y \rrbracket$.

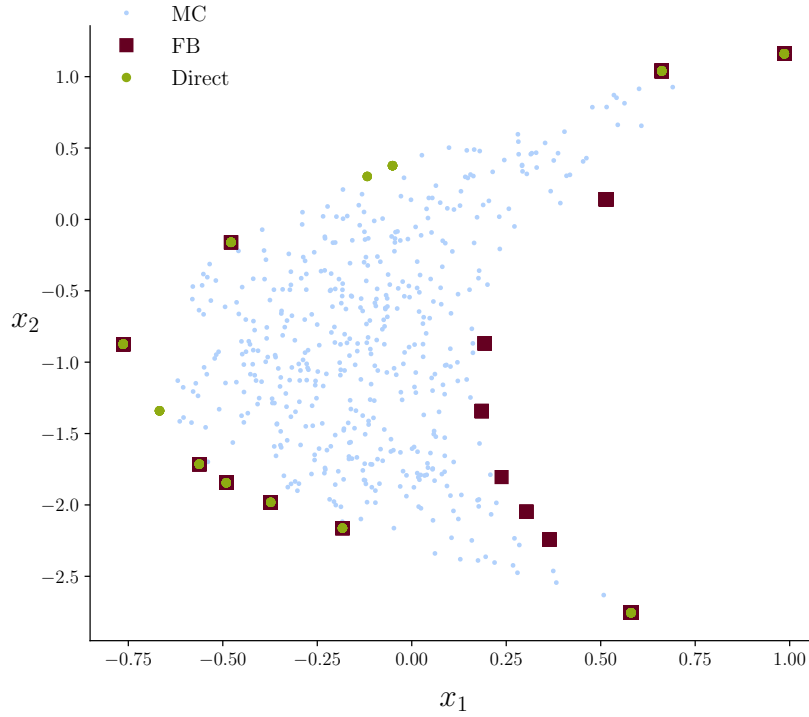


Figure 4.12: A comparison between reach set boundary of system (4.77) achieved using feedback linearization (*dark red*) vs Monte Carlo (*green*).

A simple demonstration of the above procedure is given in figure 4.12. The *light blue* dots represents Monte Carlo sample data of a 2D reach set of the feedback linearizable dynamic,

$$\dot{x}_1 = x_2/2 + x_2 v_1, \quad \dot{x}_2 = 2v_1, \quad (4.77)$$

at $t = 2$ seconds. The boundary points $\{\mathbf{z}_i^{\text{bdy}}(s)\}_{i=1}^{N_y}$ generated using feedback lin-

erization property (*dark red square*) capture the nonconvex structure of the reach set as opposed to the support of the of the convex hull of $\{z_i(s)\}_{i=1}^N$ (*light green circle*).

4.6 Parallelization

The learning strategy introduced in Section 4.3 to obtain the reach sets of full-state feedback linearizable systems allows multiple layers of parallel computation:

- [i] (Computing the components of $\partial\mathcal{X}_{jt}$)** The k th component x_k^{bdy} of each boundary point $x_j^{\text{bdy}} \in \partial\mathcal{X}_{jt}$, for $k \in \llbracket r_j \rrbracket$ and $j \in \llbracket m \rrbracket$ can be computed in parallel. In other words, parallelization is possible between the coordinates of an integrator boundary point belonging to the same block. This allows parallelization across coordinates within a given block.
- [ii] (Computing $\partial\mathcal{X}_{jt}$)** Each block of the integrator reach set $\partial\mathcal{X}_{jt}$ can be computed in parallel for $j \in \llbracket m \rrbracket$. This allows parallelization across blocks.
- [iii] (Minkowski sum of integrator blocks)** The elements of the Minkowski sum, $\partial\mathcal{X}_t = \partial\mathcal{X}_{1t} \dot{+} \partial\mathcal{X}_{2t} \dot{+} \dots \dot{+} \partial\mathcal{X}_{mt}$, can be computed in parallel independent of other points.
- [iv] (Transforming $\partial\mathcal{X}_t$ back to original coordinates $\partial\mathcal{Z}_t$)** The inverse mapping $\tau^{-1}(x^{\text{bdy}})$ can be performed component-wise in parallel, resulting in the components of $z \in \partial\mathcal{Z}_t$.

The first opportunity for parallelization occurs in item **[i]**, when computing the components of x_j^{bdy} of each $\partial\mathcal{X}_{jt}$ for $j \in \llbracket m \rrbracket$. In the following, we investigate the cost of such computation for the d dimensional single input ($m = 1$) integrator.

Consider x_k^{bdy} as the k th component of the integrator boundary given by (3.9)

$$\begin{aligned} x_k^{\text{bdy}} &= \exp(t\mathbf{A})x_{k0} + \int_0^t \nu(s)\xi_k(t-s)ds \pm \int_0^{\sigma_1} \mu(s)\xi_k(t-s)ds \\ &\mp \int_{\sigma_1}^{\sigma_2} \mu(s)\xi_k(t-s)ds \pm \dots \pm (-1)^{d+1} \int_{\sigma_{d-1}}^t \mu(s)\xi_k(t-s)ds. \end{aligned} \quad (4.78)$$

Using definitions of $\mu(s)$ and $\nu(s)$ given in (3.4), we can break the bounds of the first integral in the RHS (4.78) into d intervals and rewrite (4.78) as

$$\begin{aligned} x_k^{\text{bdy}} &= \exp(t\mathbf{A})x_{k0} + \int_0^{\sigma_1} \beta(s)\xi_k(t-s)ds + \int_{\sigma_1}^{\sigma_2} \alpha(s)\xi_k(t-s)ds + \dots \\ &+ \int_{\sigma_{d-1}}^t (\mathbf{1}_{d\%2=1}\beta(s) + \mathbf{1}_{d\%2=0}\alpha(s))\xi_k(t-s)ds, \quad \text{for } \mathbf{x}^{\text{bdy}} \in \mathcal{X}^{\text{upper}}, \end{aligned} \quad (4.79a)$$

$$\begin{aligned} x_k^{\text{bdy}} &= \exp(t\mathbf{A})x_{k0} + \int_0^{\sigma_1} \alpha(s)\xi_k(t-s)ds + \int_{\sigma_1}^{\sigma_2} \beta(s)\xi_k(t-s)ds + \dots \\ &+ \int_{\sigma_{d-1}}^t (\mathbf{1}_{d\%2=1}\alpha(s) + \mathbf{1}_{d\%2=0}\beta(s))\xi_k(t-s)ds, \quad \text{for } \mathbf{x}^{\text{bdy}} \in \mathcal{X}^{\text{lower}}, \end{aligned} \quad (4.79b)$$

where $\%$ represents the modulo operator and $\mathbf{1}_{d\%2=1}$ is the indicator function defined as

$$\begin{cases} \mathbf{1}_{d\%2=1} := 1 & \text{if } d\%2 = 1 \\ \mathbf{1}_{d\%2=1} := 0 & \text{otherwise.} \end{cases}$$

The same argument can be used for $\mathbf{1}_{d\%2=0}$.

The parameter vector $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \dots, \sigma_{d-1})$ belongs to the parameter space (2.15), and ξ_k represents the k th component of vector $\boldsymbol{\xi}$ given by (2.5).

In the following, we will perform complexity analysis to find the number of floating point operations (FLOPS) for each term in the RHS of (4.79).

[i-i] (# of FLOPS for the k th component of $\exp(\mathbf{A}t)\mathbf{x}_0$) We can employ the specific upper-diagonal structure of matrix \mathbf{A} to find the number of FLOPS associ-

ated with the matrix-vector product $\exp(\mathbf{A}s)\mathbf{x}_0$. In particular, we can write

$$\mathbf{x}_0^{\exp} = \exp(\mathbf{A}s)\mathbf{x}_0 = \begin{bmatrix} 1 & s & \frac{s^2}{2} & \cdots & \frac{s^{d-1}}{(d-1)!} \\ 0 & 1 & s & \cdots & \frac{s^{d-2}}{(d-2)!} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_{01} \\ x_{02} \\ \vdots \\ x_{0d} \end{bmatrix}. \quad (4.80)$$

The k th component of the vector \mathbf{x}_0^{\exp} , x_{0k}^{\exp} , obtained by the matrix-vector product (4.80), contains $2 \sum_{r=k}^{d-2} (d-r-1)$ FLOPS corresponding to the number of operations needed to build the k th row of $\exp(\mathbf{A}t)$ and $(d-k)$ FLOPS corresponding to the inner product of the k th row of $\exp(\mathbf{A}t)$ with \mathbf{x}_0 . Alternatively stated,

$$\# \text{ of FLOPS for } x_{0k}^{\exp} = 2 \sum_{r=k}^{d-2} (d-r-1) + (d-k). \quad (4.81)$$

[i-ii] (# of flops for ξ_k) The k th component of vector $\boldsymbol{\xi}(s)$, $\xi_k(s)$ is given by

$$\xi_k(s) = \frac{s^{d-k}}{(d-k)!},$$

which requires $(d-k-1)$ scalar multiplications in the numerator, $(d-k-2)$ scalar multiplications in the denominator. It requires 1 division if $k < d-1$, and zero additional FLOPS for $\xi_d = 1$ and $\xi_{d-1} = \sigma$, corresponding to $k = d$ and $k = d-1$, respectively. In other words,

$$\# \text{ of flop for } \xi_k(s) = \mathbf{1}_{k < d-1} ((d-k-1) + (d-k-2) + 1) = \mathbf{1}_{k < d-1} 2(d-k-1), \quad (4.82)$$

where $\mathbf{1}_{k \leq d-1}$ is the indicator function defined as

$$\begin{cases} \mathbf{1}_{k < d-1} := 1 & \text{for } k < d-1, \\ \mathbf{1}_{k < d-1} := 0 & \text{otherwise.} \end{cases}$$

[i-iii] (# of FLOPS for trapezoidal integration) Each integral in the RHS of (4.79) can be computed numerically using the variable-step size trapezoidal method such that

$$\int_{\tau_1}^{\tau_2} f(\tau) d\tau \approx \sum_{k=1}^{\substack{\# \text{ of discretizations} \\ \text{in } [\tau_1, \tau_2]^{-1}}} \frac{f(\tau_{k+1}) - f(\tau_k)}{2} \Delta\tau_k, \quad (4.83)$$

which has 3 floating point operations (FLOPS), corresponding to each discrete point τ_k , viz. 2 multiplications and 1 summation. Also, we need to account for the number of operations needed to build function f . Therefore,

$$\begin{aligned} \# \text{ of FLOPS for } \int_{\tau_1}^{\tau_2} f(\tau) d\tau &= (3 + \# \text{ of FLOPS for building } f) \\ &(\# \text{ of discretizations}). \end{aligned} \quad (4.84)$$

Suppose each component of the parameter space is discretized into N_s values, i.e.,

$$\begin{aligned} \sigma_i^{\text{disc}} &:= [s_{i_1} \ s_{i_2} \ \cdots \ s_{i_{N_s}}] \text{ such that } s_{i_n} \in [0, t] \\ &\text{for each } i \in \llbracket d-1 \rrbracket, \text{ and } n \in \llbracket N_s \rrbracket \end{aligned} \quad (4.85)$$

is the discretization vector for i^{th} coordinate of the parameter space $\mathcal{W} \subset \mathbb{R}^{d-1}$. Hence, from (4.81), (4.82), (4.84) and (4.85), the number of floating point operations incurred in computing each component of the boundary given in (4.79), is

$$\begin{aligned} \# \text{ of FLOPS for } x_k^{\text{bdy}}(\boldsymbol{\sigma}) : n_{x_k^{\text{bdy}}}^{\text{FLOPS}} &= 2 \underbrace{\sum_{r=k}^{d-2} (d-r-1) + (d-k)}_{\# \text{ of lops for } x_{0k}^{\text{exp}}} \\ &+ \underbrace{(4 + \mathbf{1}_{k < d-1} 2(d-k-1)) N_s}_{\# \text{ of FLOPS for integrals in RHS of (4.78)}}. \end{aligned} \quad (4.86)$$

When N_s is large enough, from (4.86) we get

$$n_{x_k^{\text{bdy}}}^{\text{FLOPS}} = \mathcal{O}((4 + \mathbf{1}_{k < d-1} 2(d-k-1)) N_s). \quad (4.87)$$

Hence, the number of FLOPS for computing the vector \mathbf{x}^{bdy} is

$$n_{\mathbf{x}^{\text{bdy}}}^{\text{FLOPS}} = \mathcal{O}\left(\sum_{k=1}^d [4 + \mathbf{1}_{k < d-1} 2(d-k-1)] N_s\right). \quad (4.88)$$

Notice that in the single input case, computing the k th coordinate of a boundary point in $\partial\mathcal{X}$ can be achieved using $d-1$ nested for loops, as outlined in Algorithm 1.

Algorithm 1 Integrator Boundary

```

1: for  $i_1 \leftarrow 1$  to  $N_s$  do
2:   for  $i_2 \leftarrow i_1$  to  $N_s$  do
3:      $\vdots$ 
4:     for  $i_{d-1} \leftarrow i_{d-2}$  to  $N_s$  do
5:       Return  $x_k(s_{1_{i_1}}, s_{2_{i_1}}, \dots, s_{d-1_{i_1}})$ 
6:     end for
7:      $\vdots$ 
8:   end for
9: end for

```

Let T_h denote the Tetrahedral number given by

$$T_h := \sum_{i_1=1}^{N_s} \sum_{i_2=i_1}^{N_s} \dots \sum_{i_{d-1}=i_{d-2}}^{N_s} 1 = \frac{1}{(d-1)!} N_s(N_s+1)\dots(N_s+(d-1)-1) = \mathcal{O}\left(\frac{N_s^{d-1}}{(d-1)!}\right). \quad (4.89)$$

Replacing the symbol d by r_j for the multi-input integrator, the number of floating point operations in computing the k th components of the j th block of the integrator surface, $\partial\mathcal{X}_j$, given in (3.9), is

$$T_h\left(2n_{x_k^{\text{bdy}}}^{\text{FLOPS}}\right) = \mathcal{O}\left(\frac{2(4 + \mathbf{1}_{k < r_j-1} 2(r_j - k - 1)) N_s^{r_j}}{(r_j - 1)!}\right). \quad (4.90)$$

We have multiplied $n_{x_k^{\text{bdy}}}^{\text{flops}}$ by 2 in order to account for the number of FLOPS of x_k^{bdy} in computing both the upper surface, $\mathcal{X}_j^{\text{upper}}$ and the lower surface, $\mathcal{X}_j^{\text{lower}}$, for $j \in \llbracket m \rrbracket$.

Accordingly, the number of flops for computing the integrator boundary of the j th block, $\partial\mathcal{X}_j$ is

$$\mathcal{O}\left(2 \sum_{k=1}^{r_j} \left[\frac{4 + \mathbf{1}_{k < r_j - 1} 2(r_j - k - 1)}{(r_j - 1)!} \right] N_s^{r_j}\right).$$

Remark 4.8. *In the third layer of parallelization, i.e., item **iii**, we must work with N_s^{d-m} concatenations of each combination of vectors from m blocks, i.e., if $\mathbf{x}_j^{\text{bdy},n} \in \mathbb{R}^{r_j}$ represents the n th sample point of the j th block boundary $\partial\mathcal{X}_j$, the Minkowski sum of all the m integrator blocks is numerically obtained as*

$$\mathbf{x}^{\text{bdy},n} = \begin{bmatrix} \mathbf{x}_1^{\text{bdy},n_1} & \mathbf{x}_2^{\text{bdy},n_2} & \dots & \mathbf{x}_m^{\text{bdy},n_m} \end{bmatrix}, \quad (4.91)$$

for each N_s^{d-m} combination of (n_1, n_2, \dots, n_m) such that $n_j \in \llbracket N_s^{r_j - 1} \rrbracket$ for $j \in \llbracket m \rrbracket$. Therefore, to form a new data structure with each such concatenation, we may expect (N_s^{d-m}) operations, but an efficient, lazy, implementation based on different views of a static data structure is essentially "free", and we do not count such view construction towards the total number of FLOPS.

Remark 4.9. *In the case of a full-state dynamic feedback-linearizable system, the initial condition will be a set instead of a single point (see Ch. (4.4.3)). Instead of $\exp(\mathbf{A}t)\mathbf{x}_0$, we have a tilted line segment for which we need to calculate each end point. Therefore, twice the number flops are needed vs. the case of static feedback to compute $\exp(\mathbf{A}t)\mathbf{x}_0$.*

5 | The Hausdorff Distance between Norm Balls and their Linear Maps

In the interest of computational tractability, so far we have made an assumption that the input uncertainty for the integrator reach are box-valued, characterized by the ℓ_∞ norm ball. This is despite the possibility that the true nature of these sets could adhere to an ℓ_p norm ball, where $0 < p < \infty$. Such approximations in defining the input uncertainty sets result in an over-estimation of the reach sets, as elaborated in Chapter 2. In this context, measuring the degree of over-approximation is tantamount to calculating the Hausdorff distance between the respective reach sets. Driven by this understanding, Chapter 5 focuses on estimating the Hausdorff distance between the reach set of linear time-invariant systems, which include the integrator, emerging from different norm-valued input uncertainties. The rest of this paragraph sheds light on the novel contributions stemming from this particular avenue of our research. This chapter highlights the innovative findings that are byproducts of this specific part of our research.

In control theory and formal verification literature, it is of interest to investigate how controlled dynamical systems evolve relative to each other subject to different set-valued input uncertainties [1–3, 5, 6, 111, 112, 115, 120]. For example, if the controlled dynamical systems model vehicles driving on road, then one practical question is whether the set of states reachable by one vehicle at a specific time, can intersect the other set, possibly resulting in a collision. The different set-valued inputs in the vehicle context, represent respective actuation uncertainties. Then, a natural way to quantify safety or the lack of it, is by computing the distance between such sets in terms of the Hausdorff metric.

Given compact $\mathcal{K}_1, \mathcal{K}_2 \subset \mathbb{R}^d$, the two sided Hausdorff distance δ between them is

a mapping $\delta : \mathcal{K}_1 \times \mathcal{K}_2 \mapsto \mathbb{R}_{\geq 0}$ defined as

$$\delta(\mathcal{K}_1, \mathcal{K}_2) := \max \left\{ \sup_{\mathbf{x} \in \mathcal{K}_1} \inf_{\mathbf{y} \in \mathcal{K}_2} \|\mathbf{x} - \mathbf{y}\|_2, \sup_{\mathbf{y} \in \mathcal{K}_2} \inf_{\mathbf{x} \in \mathcal{K}_1} \|\mathbf{x} - \mathbf{y}\|_2 \right\}, \quad (5.1)$$

where $\|\cdot\|_2$ is the Euclidean norm with the associated scalar product $\langle \cdot, \cdot \rangle$. Denoting the unit 2 norm ball in \mathbb{R}^d as \mathbb{B}_2^d , an equivalent definition of the Hausdorff distance is

$$\delta(\mathcal{K}_1, \mathcal{K}_2) := \inf \{ \lambda \geq 0 \mid \mathcal{K}_1 \subset \mathcal{K}_2 \dot{+} \lambda \mathbb{B}_2^d, \mathcal{K}_2 \subset \mathcal{K}_1 \dot{+} \lambda \mathbb{B}_2^d \}, \quad (5.2)$$

where $\dot{+}$ denotes the Minkowski sum. As is well-known [47, p. 60-61], $\delta \geq 0$ is a metric. The two-sided Hausdorff distance (5.1) between a pair of convex compact sets \mathcal{K}_1 and \mathcal{K}_2 in \mathbb{R}^d can be expressed in terms of their respective support functions $h_1(\cdot), h_2(\cdot)$ as

$$\delta(\mathcal{K}_1, \mathcal{K}_2) = \sup_{\mathbf{y} \in \mathbb{S}^{d-1}} |h_1(\mathbf{y}) - h_2(\mathbf{y})|, \quad (5.3)$$

where the absolute value in the objective can be dispensed if one set is included in another¹. Thus, computing δ leads to an optimization problem over all unit vectors $\mathbf{y} \in \mathbb{S}^{d-1}$. The support function, by definition, is positive homogeneous of degree one. Therefore, the unit sphere constraint $\|\mathbf{y}\|_2 = 1$ in (5.3) admits a lossless relaxation to the unit ball constraint $\|\mathbf{y}\|_2 \leq 1$. Even so, problem (5.3) is nonconvex because its objective is nonconvex in general. Please refer to Ch. 1.6.3 for more details on support functions.

When the controlled dynamical systems are linear, as it is the case for our integrator dynamics, it turns out that the corresponding Hausdorff distance (5.3) takes the form

$$\sup_{\|\mathbf{y}\|_2=1} \left(\int_0^t \|\mathbf{T}(\tau)\mathbf{y}\|_{q_2} - \|\mathbf{T}(\tau)\mathbf{y}\|_{q_1} \right) d\tau, \quad 1 \leq q_2 < q_1 \leq \infty,$$

which is what we investigate in Ch. 5.3 in this chapter.

¹This is because $\mathcal{K}_1 \subseteq \mathcal{K}_2$ if and only if $h_1(\mathbf{y}) \leq h_2(\mathbf{y})$ for all $\mathbf{y} \in \mathbb{S}^{d-1}$.

We also provide an application Example in Ch. 5.3, where the different reach sets result from the motion of a satellite subject to ℓ_2 and ℓ_∞ norm ball-valued uncertain input sets. In this application, the input components denote the radial and tangential thrusts, and depending on the actuators installed (e.g., gas jets, reaction wheel), two different scenarios may arise: one where there are hard bounds on the magnitude of the thrust components (i.e., ℓ_∞ norm ball), and another in which there is bounded thrust magnitude (i.e., ℓ_2 norm ball). So from an engineering perspective, it is natural to quantify the Hausdorff distance between the reach sets resulting from two different types of actuation uncertainties.

The Hausdorff distance was introduced by Hausdorff in 1914 [139, p. 293ff], and can be considered more generally on the set of nonempty closed and bounded subsets of a metric space $(\mathcal{M}, \text{dist})$ by replacing the Euclidean distance $\|\cdot\|_2$ in (5.1) with $\text{dist}(\cdot, \cdot)$. The Hausdorff distance and the associated topology, have found widespread applications in mathematical economics [140], stochastic geometry [141], set-valued analysis [142], image processing [143] and pattern recognition [144]. The distance δ has several useful properties with respect to set operations, see e.g., [145, Lemma 2.2], [146, Lemma A2].

In this study, we consider computing (5.3) for the case when the sets $\mathcal{K}_1, \mathcal{K}_2$ are different unit norm balls and more generally, linear maps of such norm balls in an Euclidean space. This can be viewed as quantifying the conservatism in approximating a norm ball by another in terms of the Hausdorff distance. We show that computing the associated Hausdorff distances lead to optimizing the difference between norms over the unit sphere or ellipsoid. While bounds on the difference of norms over the unit cube have been studied before [147], the optimization problems arising here seem new, and the techniques in [147] do not apply in our setting.

Related works: There have been several works on designing approximation algo-

rithms for computing the Hausdorff distance between convex polygons [148], curves [149], images [143], meshes [150] or point cloud data [151]; see also [152–156]. There are relatively few [157] known exact formula for the Hausdorff distance between sets. To the best of the authors’ knowledge, analysis of the Hausdorff distance between norm balls and their linear maps as pursued here, did not appear in prior literature.

5.1 Hausdorff Distance between Unit Norm Balls

Let us consider the case when in (5.3), the sets $\mathcal{K}_1 \equiv \mathbb{B}_{p_1}^d, \mathcal{K}_2 \equiv \mathbb{B}_{p_2}^d$, the unit ℓ_{p_1} and ℓ_{p_2} norm balls in \mathbb{R}^d , $d \geq 2$, for $1 \leq p_1 < p_2 \leq \infty$. Clearly, the Hausdorff distance $\delta = 0$ for $p_1 = p_2$, and $\delta > 0$ otherwise. Then the corresponding support functions $h_1(\cdot), h_2(\cdot)$ are the respective dual norms, i.e.,

$$h_1(\mathbf{y}) = \|\mathbf{y}\|_{q_1}, \quad h_2(\mathbf{y}) = \|\mathbf{y}\|_{q_2}, \quad \frac{1}{p_1} + \frac{1}{q_1} = 1, \quad \frac{1}{p_2} + \frac{1}{q_2} = 1,$$

for $1 \leq q_2 < q_1 \leq \infty$. By monotonicity of the norm function, we know that $\|\cdot\|_{q_1} \leq \|\cdot\|_{q_2}$. Therefore, the Hausdorff distance (5.3) in this case becomes

$$\delta(\mathcal{K}_1, \mathcal{K}_2) = \delta(\mathbb{B}_{p_1}^d, \mathbb{B}_{p_2}^d) = \sup_{\|\mathbf{y}\|_2=1} (\|\mathbf{y}\|_{q_2} - \|\mathbf{y}\|_{q_1}) \quad (5.4)$$

which has a difference of convex (DC) objective. In fact, the objective is nonconvex (the difference of convex functions may or may not be convex in general) because it admits multiple global maximizers and minimizers.

The objective in (5.4) is invariant under the plus-minus sign permutations among the components of the unit vector \mathbf{y} . There are 2^d such permutations feasible in \mathbb{R}^d which implies that the landscape of the objective in (5.4) has 2^d fold symmetry. In other words, the feasible set is subdivided into 2^d sub-domains as per the sign permutations among the components of \mathbf{y} , and the “sub-landscapes” for these sub-domains are identical.

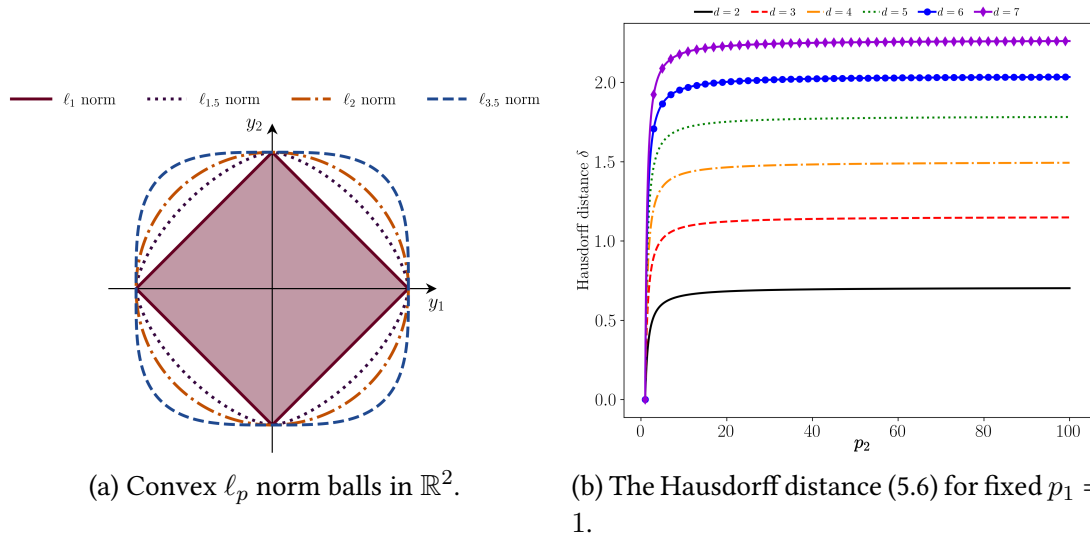


Figure 5.1: Understanding the Hausdorff distance δ between the unit ℓ_{p_1} and ℓ_{p_2} norm balls in \mathbb{R}^d , $d \geq 2$, for $1 \leq p_1 < p_2 \leq \infty$.

Since $\|\mathbf{y}\|_{q_1} \leq \|\mathbf{y}\|_{q_2}$ for $1 \leq q_2 < q_1 \leq \infty$, hence $0 \leq \delta$. The *global minimum* value of the objective in (5.4) is zero, which is achieved by any scaled basis vector, i.e., by $\mathbf{y}^{\min} := \alpha \mathbf{e}_k \in \mathbb{R}^d$ for any $k \in \llbracket d \rrbracket$ and arbitrary $\alpha \in \mathbb{R} \setminus \{0\}$. These \mathbf{y}^{\min} comprise uncountably many global minimizers for (5.4).

We can compute the *global maximum* value achieved in (5.4) using the norm inequality

$$\|\cdot\|_{q_2} \leq d^{\frac{1}{q_2} - \frac{1}{q_1}} \|\cdot\|_{q_1}, \quad 1 \leq q_2 < q_1 \leq \infty, \quad (5.5)$$

which follows from the Hölder's inequality:

$$\sum_{i=1}^d |a_i b_i| \leq \left(\sum_{i=1}^d |a_i|^r \right)^{\frac{1}{r}} \left(\sum_{i=1}^d |b_i|^{\frac{r}{r-1}} \right)^{1-\frac{1}{r}} \quad \mathbf{a}, \mathbf{b} \in \mathbb{R}^d, \quad 1 \leq r \leq \infty,$$

where the exponents r and $\frac{r}{r-1}$ are Hölder conjugates. Applying this inequality with $|a_i| = |x_i|^{q_2}$, $|b_i| = 1$, $r = q_1/q_2 > 1$, results in (5.5).

In \mathbb{R}^d , the constant $d^{1/q_2-1/q_1}$ is sharp because the equality in (5.5) is achieved by any vector in $\{-1, 1\}^d$. Since (5.4) has constraint $\|\mathbf{y}\|_2 = 1$, the corresponding global

maximum will be achieved by

$$\mathbf{y}^{\max} \in \mathcal{Y}^{\max} := \{\mathbf{u} \in \mathbb{S}^{d-1} \mid \mathbf{u} = \rho \mathbf{v}, \mathbf{v} \in \{-1, 1\}^d, \rho > 0\}.$$

The scalar ρ is determined by the normalization constraint $\|\mathbf{y}^{\max}\|_2 = 1$ as $\rho = 1/\sqrt{d}$.

Thus, we obtain

$$\begin{aligned} \delta &= \sup_{\|\mathbf{y}\|_2=1} (\|\mathbf{y}\|_{q_2} - \|\mathbf{y}\|_{q_1}), \quad 1 \leq q_2 < q_1 \leq \infty, \\ &= \left(d^{\frac{1}{q_2} - \frac{1}{q_1}} - 1 \right) \underbrace{\|\mathbf{y}^{\max}\|_{q_1}}_{=\rho d^{\frac{1}{q_1}}} = d^{-\frac{1}{2}} \left(d^{\frac{1}{q_2}} - d^{\frac{1}{q_1}} \right), \end{aligned} \quad (5.6)$$

where in the last line we substituted $\rho = 1/\sqrt{d}$. The cardinality of \mathcal{Y}^{\max} equals 2^d , i.e., there are 2^d global maximizers $\mathbf{y}^{\max} \in \mathbb{S}^{d-1}$ achieving the value (5.6).

We summarize the above in the following Proposition.

Proposition 5.1. *For $1 \leq p_1 < p_2 \leq \infty$, we have*

$$\delta(\mathbb{B}_{p_1}^d, \mathbb{B}_{p_2}^d) = d^{-\frac{1}{2}} \left(d^{\frac{1}{q_2}} - d^{\frac{1}{q_1}} \right), \quad (5.7)$$

where q_i denotes the Hölder conjugate of p_i for $i \in \{1, 2\}$.

Remark 5.1. *As the intuition suggests, for a fixed p_1 , larger p_2 results in a larger δ in a given dimension $d \geq 2$; see Fig. 5.1.*

Fig. 5.2 shows the contour plot of $\|\mathbf{y}\|_1 - \|\mathbf{y}\|_2$ in the spherical coordinates for $d = 3$, i.e., $\mathbf{y} \in \mathbb{S}^2$. As predicted by (5.6), in this case, there are eight maximizers achieving the global maximum value $\sqrt{3} - 1 \approx 0.7321$. The symmetric sub-landscapes mentioned earlier are also evident in Fig. 5.2.

5.1.1 Hausdorff distance between polyhedral D -norm balls

We next show that similar arguments as above can be used to derive the Hausdorff distance between other type of norm balls such as the D -norm balls which are certain

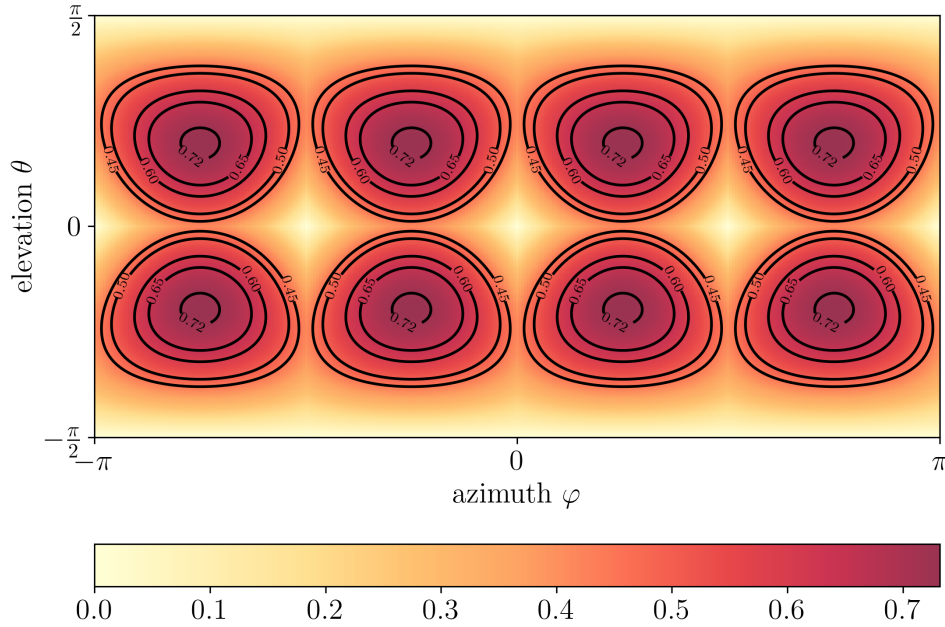


Figure 5.2: The landscape of the objective in (5.4) for $d = 3$, $q_1 = 2$, and $q_2 = 1$.

polyhedral norm balls. The D -norms and norm balls arise naturally in robust optimization, see e.g., [158, Sec. 2.2], [159]. The D -norm in \mathbb{R}^d is parameterized by k , where $1 \leq k \leq d$, as defined next.

Definition 5.1. (D -norm) For $1 \leq k \leq d$, the D -norm of $\mathbf{x} \in \mathbb{R}^d$ is

$$\|\mathbf{x}\|_k^D := \max_{\{S \cup \{t\} | S \subseteq \llbracket d \rrbracket, \text{card}(S) \leq [k], t \in \llbracket d \rrbracket \setminus S\}} \left\{ \sum_{i \in S} |x_i| + (k - [k]) |x_t| \right\}. \quad (5.8)$$

For $k = 1$, the norm (5.8) reduces to the ℓ_∞ norm, i.e., $\|\mathbf{x}\|_1^D = \|\mathbf{x}\|_\infty$. For $k = d$, the norm (5.8) reduces to the ℓ_1 norm, i.e., $\|\mathbf{x}\|_d^D = \|\mathbf{x}\|_1$. For $1 < k < d$, the norm (5.8) can be thought of as a polyhedral interpolation between the ℓ_∞ and the ℓ_1 norms. For a plot of the unit D -norm balls in \mathbb{R}^2 , we refer the readers to [158, Fig. 2].

A special case of (5.8) is when the parameter k is restricted to be a natural number, i.e., $k \in \llbracket d \rrbracket$. Then the D -norm reduces to the so-called k largest magnitude norm, defined next.

Definition 5.2. For $k \in \llbracket d \rrbracket$, the k largest magnitude norm of $\mathbf{x} \in \mathbb{R}^d$ is

$$\|\mathbf{x}\|_{[k]} := |x_{i_1}| + |x_{i_2}| + \cdots + |x_{i_k}|, \quad (5.9)$$

where the inequality $|x_{i_1}| \geq |x_{i_2}| \geq \cdots \geq |x_{i_d}|$ denotes the ordering of the magnitudes of the entries in \mathbf{x} .

It is easy to verify that (5.8) (and thus its special case (5.9)) is indeed a norm, and its dual norm equals [159, Prop. 2]

$$\left(\|\cdot\|_k^D\right)^*(\mathbf{y}) = \max \left\{ \frac{1}{k} \|\mathbf{y}\|_1, \|\mathbf{y}\|_\infty \right\}.$$

For a comparison of the D -norm and its dual with the Euclidean norm, see [159, Prop. 3]. We have the following result (proof in Appendix E.1).

Proposition 5.2. Let $1 \leq k_1 < k_2 \leq d$, and let $\mathcal{K}_1, \mathcal{K}_2 \subset \mathbb{R}^d$ denote the unit $\|\cdot\|_{k_1}^D$ and $\|\cdot\|_{k_2}^D$ norm balls in \mathbb{R}^d , i.e., $\mathcal{K}_1 \equiv \mathbb{B}_{\|\cdot\|_{k_1}^D}^d, \mathcal{K}_2 \equiv \mathbb{B}_{\|\cdot\|_{k_2}^D}^d$. Also denote the support functions of $\mathcal{K}_1, \mathcal{K}_2$ as $h_1(\cdot), h_2(\cdot)$, respectively. Using the definition of dual norm, we have

$$h_1(\mathbf{y}) = \max \left\{ \frac{1}{k_1} \|\mathbf{y}\|_1, \|\mathbf{y}\|_\infty \right\}, \quad h_2(\mathbf{y}) = \max \left\{ \frac{1}{k_2} \|\mathbf{y}\|_1, \|\mathbf{y}\|_\infty \right\}. \quad (5.10)$$

Then

$$\delta(\mathcal{K}_1, \mathcal{K}_2) = \delta\left(\mathbb{B}_{\|\cdot\|_{k_1}^D}^d, \mathbb{B}_{\|\cdot\|_{k_2}^D}^d\right) = \left(\frac{1}{k_1} - \frac{1}{k_2}\right) \sqrt{d}. \quad (5.11)$$

Fig. 5.3 shows the landscape of the objective for computing the Hausdorff distance between the unit D -norm balls with $k_1 = 1.7$ and $k_2 = 2.9$ in $d = 3$ dimensions, and as explained in the proof of the proposition above, there are eight global maximizers given by $\mathbf{v}/\sqrt{3}$ for all $\mathbf{v} \in \{-1, 1\}^3$. In this case, the formula (5.11) gives $\delta = 120\sqrt{3}/493 \approx 0.421594517055305$ while the direct numerical estimate of δ from the contours yields 0.421577951149235.

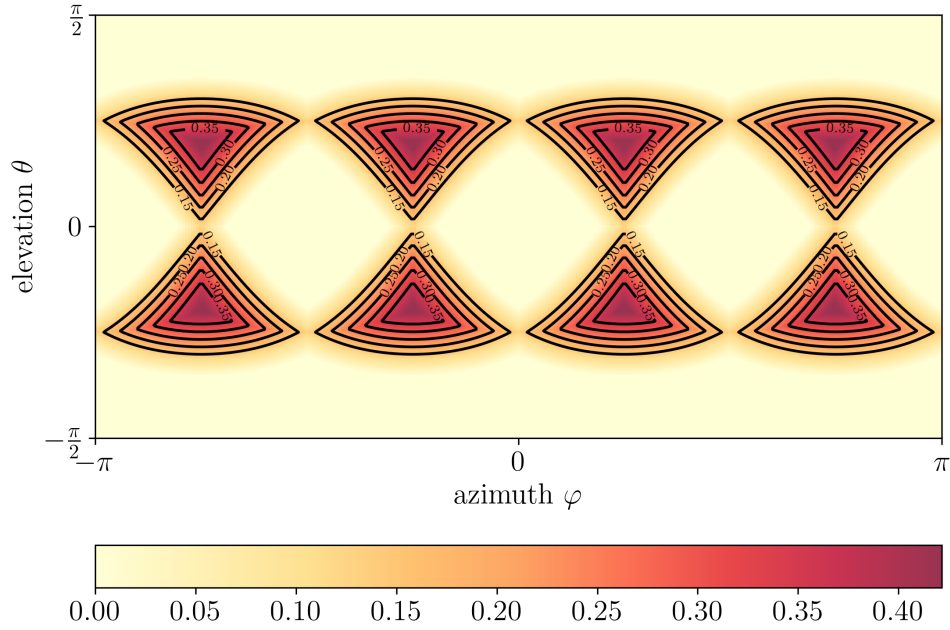


Figure 5.3: The landscape of the objective in (5.3) with h_1, h_2 as in (5.10) for $d = 3$, $k_1 = 1.7$, and $k_2 = 2.9$.

5.2 Composition with a Linear Map

We next consider a generalized version of (5.4) given by

$$\delta(\mathcal{K}_1, \mathcal{K}_2) = \sup_{\|\mathbf{y}\|_2=1} (\|\mathbf{T}\mathbf{y}\|_{q_2} - \|\mathbf{T}\mathbf{y}\|_{q_1}), \quad 1 \leq q_2 < q_1 \leq \infty, \quad (5.12)$$

where the matrix $\mathbf{T} \in \mathbb{R}^{m \times d}$, $m \leq d$, has full row rank m . Using Property 1.6.3 of support functions, we can interpret (5.12) as follows. As before, let p_1, p_2 denote the Hölder conjugates of q_1, q_2 , respectively. Then (5.12) computes the Hausdorff distance between two compact convex sets in \mathbb{R}^d obtained as the linear transformations of the m -dimensional ℓ_{p_1} and ℓ_{p_2} unit norm balls via $\mathbf{T}^\top \in \mathbb{R}^{d \times m}$, i.e.,

$$\mathcal{K}_1 \equiv \mathbf{T}^\top \mathbb{B}_{p_1}^d, \quad \mathcal{K}_2 \equiv \mathbf{T}^\top \mathbb{B}_{p_2}^d.$$

Since the right pseudo-inverse $\mathbf{T}^\dagger = \mathbf{T}^\top (\mathbf{T}\mathbf{T}^\top)^{-1}$, one can equivalently view (5.12) as that of maximizing the difference between the ℓ_{p_1} and ℓ_{p_2} norms over the m -dimensional origin-centered ellipsoid with shape matrix $\mathbf{T}\mathbf{T}^\top$.

As was the case in (5.4), problem (5.12) is a DC programming problem with non-convex objective. However, unlike (5.4), now there is no obvious symmetry in the objective's landscape that can be leveraged because the number and locations of the local maxima or saddles have sensitive dependence on the matrix parameter \mathbf{T} ; see the first column of Table 5.1. Thus, directly using off-the-shelf solvers such as [160, 161] or nonconvex search algorithms become difficult for solving (5.12) in practice as the iterative search may get stuck in a local stationary point.

Remark 5.2. *We can also consider the Hausdorff distance between the common linear transforms of different polyhedral D norm balls discussed earlier. Specifically, if $\mathcal{K}_1, \mathcal{K}_2$ are the unit $\|\cdot\|_{k_1}^D, \|\cdot\|_{k_2}^D$ norm balls for $1 \leq k_1 < k_2 \leq d$, then following Property 1.6.3 of support functions, and the steps in the proof of Proposition 5.2, the Hausdorff distance δ between the sets $\mathbf{T}\mathcal{K}_1, \mathbf{T}\mathcal{K}_2$ equals*

$$\delta(\mathbf{T}\mathcal{K}_1, \mathbf{T}\mathcal{K}_2) = \left(\frac{1}{k_1} - \frac{1}{k_2}\right) \underbrace{\sup_{\|\mathbf{y}\|_2=1} \|\mathbf{T}^\top \mathbf{y}\|_1}_{=:\|\mathbf{T}^\top\|_{2 \rightarrow 1}} = \left(\frac{1}{k_1} - \frac{1}{k_2}\right) \|\mathbf{T}\|_{\infty \rightarrow 2}, \quad (5.13)$$

where the last equality follows from the relation between the induced norm of an operator and that of its adjoint.

5.2.1 Estimates for arbitrary \mathbf{T}

We next provide an upper bound for (5.12) in terms of the operator norm $\|\mathbf{T}\|_{2 \rightarrow q_1}$ (proof in Appendix E.2).

Proposition 5.3. *Let $\mathbf{T} \in \mathbb{R}^{m \times d}$. Then for $1 \leq q_2 < q_1 \leq \infty$, we have*

$$\sup_{\|\mathbf{y}\|_2=1} (\|\mathbf{T}\mathbf{y}\|_{q_2} - \|\mathbf{T}\mathbf{y}\|_{q_1}) \leq \left(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1\right) \|\mathbf{T}\|_{2 \rightarrow q_1}. \quad (5.14)$$

Recall that $1 < q_1 \leq \infty$. When $1 \leq q_2 < q_1 \leq 2$, the operator norm $\|\mathbf{T}\|_{2 \rightarrow q_1}$ is, in general, NP hard to compute [162–164] except in the well-known case $q_1 = 2$ for which

$\|\mathbf{T}\|_{2 \rightarrow 2} = \sigma_{\max}(\mathbf{T})$, the maximum singular value of \mathbf{T} . When $1 \leq q_2 \leq 2 < q_1 \leq \infty$, the norm $\|\mathbf{T}\|_{2 \rightarrow q_1}$ is often referred to as hypercontractive [165], and its computation for generic $\mathbf{T} \in \mathbb{R}^{m \times d}$ is relatively less explored (see e.g., [165, 166]) except for the case $q_1 = \infty$ for which $\mathbf{T}_{2 \rightarrow \infty} = \max_{i=1, \dots, m} \|\mathbf{T}(i, :)\|_2$ (maximum ℓ_2 norm of a row). Hypercontractive norms and related inequalities find applications in establishing rapid mixing of random walks as well as several problems of interest in theoretical computer science [165, 167–169].

Table 5.1 dissertations our numerical experiments to estimate (5.12) with $q_1 = 2, q_2 = 1$, for five random realizations of $\mathbf{T} \in \mathbb{R}^{3 \times 3}$, arranged as the rows of Table 5.1. For visual clarity, the contour plots in the first column of Table 5.1 depict only four high-magnitude contour levels. These results suggest that the landscape of the nonconvex objective in (5.12) has sensitive dependence on the mapping \mathbf{T} .

We can say more for specific classes of \mathbf{T} . For example, notice from (5.14) that if the mapping $\mathbf{T} : \ell_q(\mathbb{R}^d) \mapsto \ell_q(\mathbb{R}^m)$ is an isometry, i.e., $\|\mathbf{T}\mathbf{y}\|_q = \|\mathbf{y}\|_q$, then the upper bound is achieved by any $\mathbf{y} \in \mathbb{R}^d$ such that $\sqrt{d}\mathbf{y} \in \{-1, 1\}^d$ as in Ch. 5.1, and we recover the exact formula (5.7). We can characterize these isometric maps as follows.

Proposition 5.4. *Consider a linear mapping given by $\mathbf{T} : \ell_q(\mathbb{R}^d) \mapsto \ell_q(\mathbb{R}^m)$.*

(i) (See e.g., [170, Remark 3.1]) *For $q = 2$, the mapping $\mathbf{T} \in \mathbb{R}^{m \times d}$ is an isometry if and only if $\mathbf{T}^\top \mathbf{T} = \mathbf{I}_d$, i.e., \mathbf{T} is a column-orthonormal matrix.*

(ii) ([170, Thm. 3.2]) *For $q \in [1, \infty) \setminus \{2\}$, the mapping $\mathbf{T} \in \mathbb{R}^{m \times d}$ is an isometry if and only if there exists a permutation matrix $\mathbf{P} \in \mathbb{R}^{m \times m}$ such that $\mathbf{P}\mathbf{T} = \text{diag}(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_d)$ and $\|\mathbf{r}_j\|_q = 1$ for all $j \in [d]$. In particular, when $d = m$ and $q \in [1, \infty) \setminus \{2\}$, the mapping \mathbf{T} is isometry if and only if it is a signed permutation matrix [171, 172], i.e. a permutation*

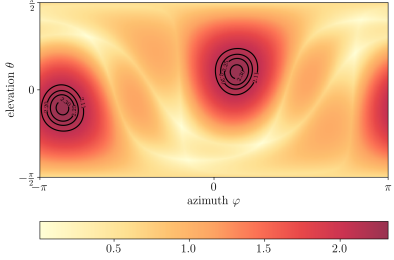
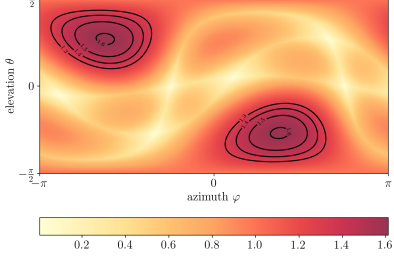
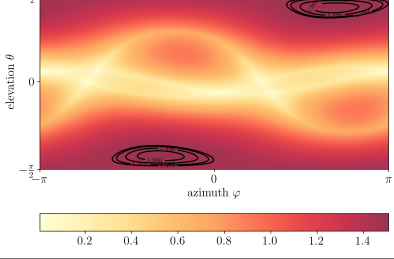
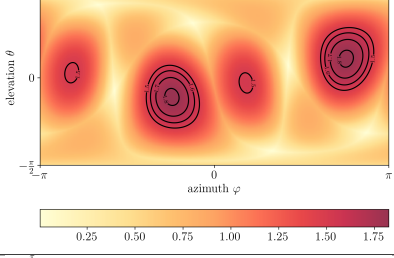
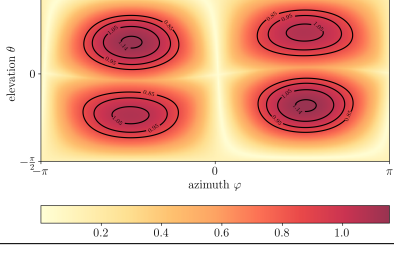
Landscape	Estimated max.	Upper bound (5.14)
	2.318732079842	2.384527280099
	1.611342375325	1.722680031455
	1.507938701982	1.583409927359
	1.824182821725	2.157801577048
	1.154650461995	1.303457527919

Table 5.1: Landscapes of $\|\mathbf{T}\mathbf{y}\|_{q_2} - \|\mathbf{T}\mathbf{y}\|_{q_1}$ for $q_1 = 2, q_2 = 1, \mathbf{y} \in \mathbb{S}^2$ in spherical coordinates for five randomly generated $\mathbf{T} \in \mathbb{R}^{3 \times 3}$ with independent standard Gaussian entries. The middle column dissertations the numerically estimated global maxima from the respective contour data, i.e., the estimated Hausdorff distance (5.12). The last column shows the corresponding bounds (5.14).

matrix whose nonzero entries are either all +1 or all -1 or some +1 and the rest -1.

The following is an immediate consequence of this characterization.

Corollary 5.1. *For \mathbf{T} as in Proposition 5.4, the Hausdorff distance δ in (5.12) equals (5.7).*

An instance in which $\|\mathbf{T}\|_{2 \rightarrow q_1}$ and hence the bound (5.14) is efficiently computable, occurs when $\mathbf{T} \in \mathbb{R}^{m \times d}$ is elementwise nonnegative and $1 \leq q_1 < 2$. In this case, the operator norm $\|\mathbf{T}\|_{2 \rightarrow q_1}$ is known [163, Thm. 3.3] to be equal to the optimal value of the following convex optimization problem:

$$\begin{aligned} \text{OPT} &:= \max_{\mathbf{X} \geq \mathbf{0}} \sqrt{\|\text{dg}(\mathbf{T}\mathbf{X}\mathbf{T}^\top)\|_{\frac{q_1}{2}}} \\ &\text{subject to } \|\text{dg}(\mathbf{X})\|_1 \leq 1, \end{aligned} \tag{5.15}$$

where $\text{dg}(\cdot)$ takes a square matrix as its argument and returns the vector comprising of the diagonal entries of that matrix. To see why problem (5.15) is convex, notice that $\mathbf{X} \geq \mathbf{0}$ has unique (principal) square root, so $\mathbf{T}\mathbf{X}\mathbf{T}^\top = \mathbf{T}\mathbf{X}^{\frac{1}{2}}(\mathbf{T}\mathbf{X}^{\frac{1}{2}})^\top \geq \mathbf{0}$ which implies $\text{dg}(\mathbf{T}\mathbf{X}\mathbf{T}^\top)$ has nonnegative entries. Consequently, the objective in (5.15) is concave for $1 \leq q_1 < 2$. The non-empty feasible set $\{\mathbf{X} \in \mathbb{R}^{d \times d} \mid \mathbf{X} \geq \mathbf{0}, \|\text{dg}(\mathbf{X})\|_1 = \sum_{i=1}^d X_{ii} \leq 1\}$ is the intersection of the positive semidefinite cone with a linear inequality, hence convex (in fact a spectrahedron).

Then, the right hand side of (5.14) equals $(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1) \times \text{OPT}$. For example, when

$$q_1 = 1.5, \quad q_2 = 1, \quad \mathbf{T} = \begin{bmatrix} 2 & 6 & 0 \\ 5 & 0 & 1 \end{bmatrix}, \tag{5.16}$$

a numerical solution of (5.15) via `cvx` [110] gives $\text{OPT} \approx 7.425702405524379$. As in Table 1, a direct numerical search over the nonconvex landscape (Fig. 5.4) for this example returns the estimated Hausdorff distance $\approx 1.888517738190415$ while using the numerically computed OPT, we find the upper bound (5.14) $\approx 1.930096365450782$.

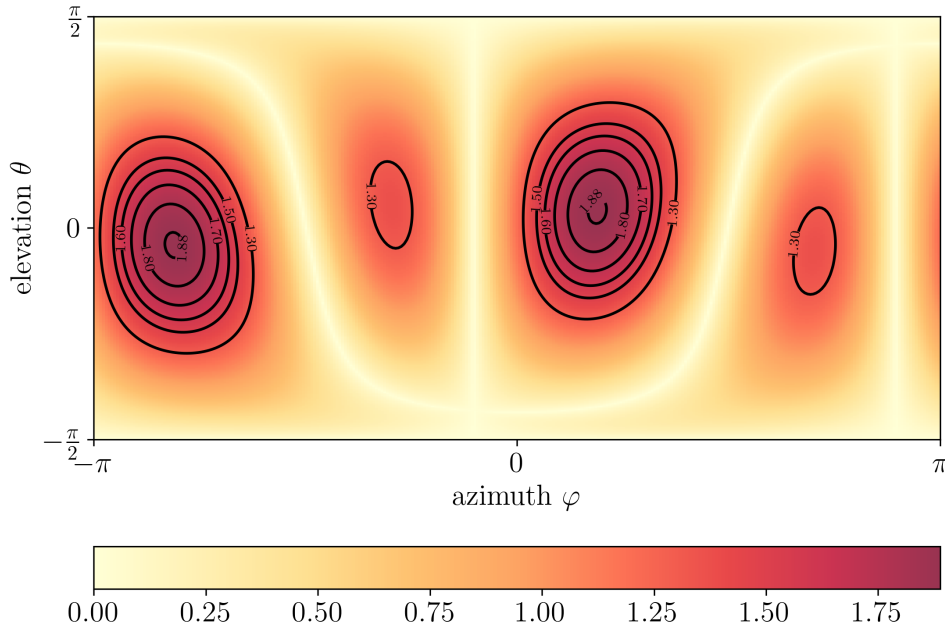


Figure 5.4: The landscape of the objective in (5.12) depicted in spherical coordinates for the problem data given in (5.16).

Remark 5.3. We clarify here that for (5.15) to be used in the upper bound (5.14), the range of q_1 is $1 < q_1 < 2$. That $\|\mathbf{T}\|_{2 \rightarrow q_1}$ equals to (5.15) holds also for the case $q_1 = 1$. Indeed, this implies we can compute (5.13) for elementwise nonnegative \mathbf{T} by computing $\|\mathbf{T}^\top\|_{2 \rightarrow 1}$ via convex optimization.

5.2.2 Estimates for random \mathbf{T}

for random linear maps $\mathbf{T} : \ell_q(\mathbb{R}^d) \mapsto \ell_q(\mathbb{R}^m)$, it is possible to bound the expected Hausdorff distance (5.12). We collect two such results in the following proposition (proof in Appendix E.3).

Proposition 5.5. Let $2 \leq q_1 < \infty$.

(i) Let $\mathbf{T} = (\theta_{ij})_{i,j=1}^{m,d}$ have independent (not necessarily identically distributed) mean-zero

entries with $|\theta_{ij}| \leq 1$ for all index pair (i, j) . Then the Hausdorff distance (5.12) satisfies

$$\mathbb{E} \delta \leq \left(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1 \right) C_{q_1} \max\{m^{\frac{1}{q_1}}, \sqrt{d}\} \quad (5.17)$$

where the pre-factor C_{q_1} depends only on q_1 .

(ii) Let $\mathbf{T} = (\theta_{ij})_{i,j=1}^{m,d}$ have independent standard Gaussian entries. Then the Hausdorff distance (5.12) satisfies

$$\mathbb{E} \delta \leq C \left(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1 \right) 2^{5/q_1} (\log m)^{1/q_1} \left(\gamma_2 + \gamma_{q_1} \mathbb{E} \max_{i,j} |\theta_{ij}| \right) + 2^{1/q_1} \gamma_{q_1} \quad (5.18)$$

where $C > 0$ is a constant, and $\gamma_r := (\mathbb{E}|X|^r)^{1/r}$, $r \geq 1$, is the L_r norm of a standard Gaussian random variable X . In particular, $\gamma_r \asymp \sqrt{r}$, i.e., there exist positive constants c_1, c_2 such that $c_1\sqrt{r} \leq \gamma_r \leq c_2\sqrt{r}$ for all $r \geq 1$.

5.3 Integral Version and Application

We now consider a further generalization of (5.12) given by

$$\delta(\mathcal{K}_1, \mathcal{K}_2) = \sup_{\|\mathbf{y}\|_2=1} \int_0^t (\|\mathbf{T}(\tau)\mathbf{y}\|_{q_2} - \|\mathbf{T}(\tau)\mathbf{y}\|_{q_1}) d\tau, \quad 1 \leq q_2 < q_1 \leq \infty, \quad (5.19)$$

where for each $\tau \in [0, t]$, the matrix $\mathbf{T}(\tau) \in \mathbb{R}^{m \times d}$, $m \leq d$, is smooth in τ and has full row rank m .

As before, let p_1, p_2 denote the Hölder conjugates of q_1, q_2 , respectively. We can interpret (5.19) as computing the Hausdorff distance between two compact convex sets in \mathbb{R}^d obtained by first taking linear transformations of the m -dimensional p_1 and p_2 unit norm balls via $\mathbf{T}^\top(\tau) \in \mathbb{R}^{d \times m}$ for fixed $\tau \in [0, t]$, and then taking respective Minkowski sums for varying τ and finally passing to the limit. In particular, if we let $\mathcal{P}_i := \{\mathbf{v} \in \mathbb{R}^m \mid \|\mathbf{v}\|_{p_i} \leq 1\}$ for $i \in \{1, 2\}$, then (5.19) computes the Hausdorff distance

between the d dimensional compact convex sets

$$\mathcal{K}_1 \equiv \int_0^t \mathbf{T}^\top(\tau) \mathcal{P}_1 d\tau := \lim_{\Delta \downarrow 0} \sum_{i=0}^{\lfloor t/\Delta \rfloor} \Delta \mathbf{T}^\top(i\Delta) \mathcal{P}_1, \quad (5.20a)$$

$$\mathcal{K}_2 \equiv \int_0^t \mathbf{T}^\top(\tau) \mathcal{P}_2 d\tau := \lim_{\Delta \downarrow 0} \sum_{i=0}^{\lfloor t/\Delta \rfloor} \Delta \mathbf{T}^\top(i\Delta) \mathcal{P}_2, \quad (5.20b)$$

i.e., the sets under consideration are set-valued Aumann integrals [173] and the symbol Σ denotes the Minkowski sum. That the sets in (5.20) are convex is a consequence of the Lyapunov convexity theorem [48, 49].

Notice that in this case, (5.14) directly yields

$$\delta(\mathcal{K}_1, \mathcal{K}_2) \leq \int_0^t \sup_{\|\mathbf{y}\|_2=1} (\|\mathbf{T}(\tau)\mathbf{y}\|_{q_2} - \|\mathbf{T}(\tau)\mathbf{y}\|_{q_1}) d\tau \leq \left(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1\right) \int_0^t \|\mathbf{T}(\tau)\|_{2 \rightarrow q_1} d\tau. \quad (5.21)$$

A different way to deduce (5.21) is to utilize the definitions (5.20), and then combine the Hausdorff distance property in [145, Lemma 2.2(ii)] with a limiting argument. This gives

$$\delta\left(\int_0^t \mathbf{T}^\top(\tau) \mathcal{P}_1 d\tau, \int_0^t \mathbf{T}^\top(\tau) \mathcal{P}_2 d\tau\right) \leq \int_0^t \delta(\mathbf{T}^\top(\tau) \mathcal{P}_1, \mathbf{T}^\top(\tau) \mathcal{P}_2) d\tau. \quad (5.22)$$

For a fixed $\tau \in [0, t]$, the integrand in the right hand side of (5.22) is precisely (5.12), hence using Proposition 5.3 we again arrive at (5.21).

As a motivating application, consider two controlled linear dynamical agents with identical dynamics given by the ordinary differential equation

$$\dot{\mathbf{x}}^i(t) = \mathbf{A}(t)\mathbf{x}^i(t) + \mathbf{B}(t)\mathbf{u}^i(t), \quad i \in \{1, 2\}, \quad (5.23)$$

where $\mathbf{x}^i(t) \in \mathbb{R}^d$ is the state and $\mathbf{u}^i(t) \in \mathbb{R}^m$ is the control input for the i th agent at time t . Suppose that the system matrices $\mathbf{A}(t)$, $\mathbf{B}(t)$ are smooth measurable functions of t , and that the initial conditions for the two agents have the same compact convex set valued uncertainty, i.e., $\mathbf{x}^i(t=0) \in \text{compact convex } \mathcal{X}_0 \subset \mathbb{R}^d$. Furthermore, suppose

that the input uncertainty sets for the two systems are given by different unit norm balls

$$\mathcal{U}^i := \{\mathbf{u}^i(\tau) \in \mathbb{R}^m \mid \|\mathbf{u}^i(\tau)\|_{p_i} \leq 1 \text{ for all } \tau \in [0, t]\}, \quad i \in \{1, 2\}, \quad (5.24)$$

such that $1 \leq p_1 < p_2 \leq \infty$. Given these set-valued uncertainties, the “reach sets” \mathcal{X}_t^i , $i \in \{1, 2\}$, are defined as the respective set of states each agent may reach at a given time $t > 0$. Specifically, for $i \in \{1, 2\}$, and \mathcal{U}^i given by (5.24), the reach sets are

$$\begin{aligned} \mathcal{X}_t^i := & \bigcup_{\text{measurable } \mathbf{u}^i(\cdot) \in \mathcal{U}^i} \{\mathbf{x}^i(t) \in \mathbb{R}^d \mid \dot{\mathbf{x}}^i(t) = \mathbf{A}(t)\mathbf{x}^i(t) + \mathbf{B}(t)\mathbf{u}^i(t), \quad i \in \{1, 2\}, \\ & \mathbf{x}^i(t=0) \in \text{compact convex } \mathcal{X}_0, \quad \mathbf{u}^i(\tau) \in \mathcal{U}^i \text{ for all } 0 \leq \tau \leq t\}. \end{aligned} \quad (5.25)$$

As such, there exists a vast literature [1–3, 5, 6, 111, 112, 115, 120] on reach sets and their numerical approximations. In practice, these sets are of interest because their separation or intersection often imply safety or the lack of it. It is natural to quantify the distance between reach sets or their approximations in terms of the Hausdorff distance [105, 174, 175], and in our context, this amounts to estimating $\delta(\mathcal{X}_t^1, \mathcal{X}_t^2)$.

Since $1 \leq p_1 < p_2 \leq \infty$, we have the norm ball inclusion $\mathcal{U}^1 \subset \mathcal{U}^2$, and consequently $\mathcal{X}_t^1 \subset \mathcal{X}_t^2$. We next show that $\delta(\mathcal{X}_t^1, \mathcal{X}_t^2)$ is exactly of the form (5.19) (proof in Appendix E.4).

Theorem 5.2. *Consider the reach sets (5.25) with input set valued uncertainty (5.24). For $\tau \leq t$, let $\Phi(t, \tau)$ be the state transition matrix (see e.g., [176, Ch. 1.3]) associated with (5.23). Denote the Hölder conjugate of p_1 as q_1 , and that of p_2 as q_2 , i.e., $1/p_1 + 1/q_1 = 1$ and $1/p_2 + 1/q_2 = 1$. Then $1 \leq q_2 < q_1 \leq \infty$, and the Hausdorff distance*

$$\delta(\mathcal{X}_t^1, \mathcal{X}_t^2) = \sup_{\|\mathbf{y}\|_2=1} \int_0^t (\|(\Phi(t, \tau)\mathbf{B}(\tau))^\top \mathbf{y}\|_{q_2} - \|(\Phi(t, \tau)\mathbf{B}(\tau))^\top \mathbf{y}\|_{q_1}) \, d\tau. \quad (5.26)$$

Corollary 5.3. *Using the same notations of Theorem 5.2, we have*

$$\delta(\mathcal{X}_t^1, \mathcal{X}_t^2) \leq \left(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1\right) \int_0^t \|\Phi(t, \tau) \mathbf{B}(\tau)\|_{p_1 \rightarrow 2} d\tau. \quad (5.27)$$

The proof for Corollary 5.3 is given in Appendix E.5.

Remark 5.4. *In the special case of a linear time invariant dynamics, the matrices \mathbf{A} , \mathbf{B} in (5.23) are constants and $\Phi(t, \tau) = \exp((t - \tau)\mathbf{A})$. In that case, Theorem 5.2 and Corollary 5.3 apply with these additional simplifications.*

Remark 5.5. *As t increases, we expect the bound (5.21) to become more conservative. Likewise, the gap between (5.26) and (5.27) is expected to increase with t .*

Example. Consider the linearized equation of motion of a satellite [176, p. 14-15] of the form (5.23) with four states, two control inputs, and constant system matrices

$$\mathbf{A}(t) \equiv \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3\omega^2 & 0 & 0 & 2\omega \\ 0 & 0 & 0 & 1 \\ 0 & -2\omega & 0 & 0 \end{bmatrix}, \quad \mathbf{B}(t) \equiv \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad (5.28)$$

for some fixed parameter ω . The input components denote the radial and tangential thrusts, respectively. We consider two cases: the inputs have set-valued uncertainty of the form (5.24) with $p_1 = 2$ (unit Euclidean norm-bounded thrust) and with $p_2 = \infty$

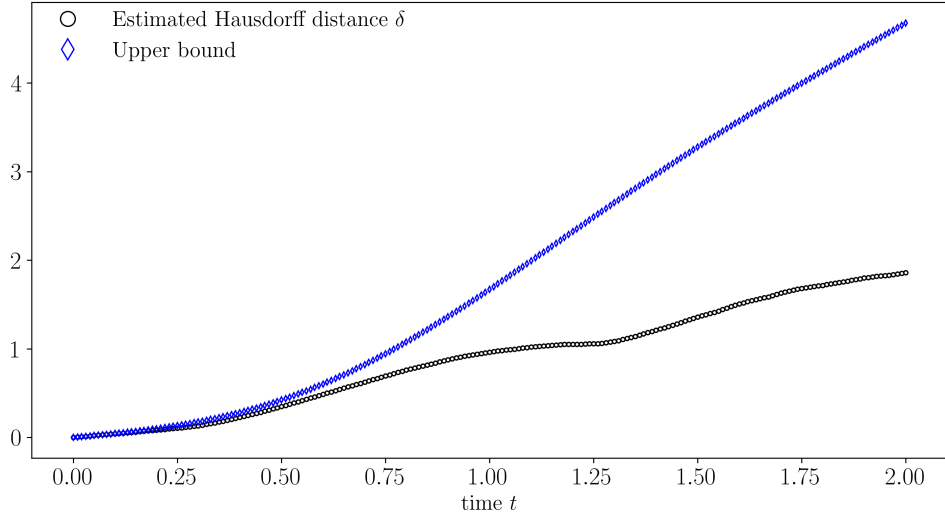


Figure 5.5: The numerically estimated Hausdorff distance (5.26) and the upper bound (5.27) for the four state, two input linear system given in (5.28).

(unit box-valued thrust). We have [176, p. 41]

$$\Phi(t, \tau)B = \begin{bmatrix} \frac{\sin(\omega(t-\tau))}{\omega} & \frac{2(1-\cos(\omega(t-\tau)))}{\omega} \\ \cos(\omega(t-\tau)) & 2\sin(\omega(t-\tau)) \\ \frac{2(1-\cos(\omega(t-\tau)))}{\omega} & \frac{-3\omega(t-\tau) + 4\sin(\omega(t-\tau))}{\omega} \\ -2\sin(\omega(t-\tau)) & -3 + 4\sin(\omega(t-\tau)) \end{bmatrix},$$

for $0 \leq \tau < t$, and the integrand in the right hand side of (5.27) equals to the maximum singular value of the above matrix. For $\omega = 3$ and $t \in [0, 2]$, Fig. 5.5 shows the time evolution of the numerically estimated Hausdorff distance (5.26) and the upper bound (5.27) between the reach sets given by (5.25) with the same compact convex initial set $\mathcal{X}_0 \subset \mathbb{R}^4$, i.e., between \mathcal{X}_t^1 and \mathcal{X}_t^2 resulting from the unit $p_1 = 2$ and $p_2 = \infty$ norm ball input sets, respectively.

6 | Support Function Representation Learning of Compact Sets

In this chapter, motivated by data-driven systems-control applications, we propose learning the support function of an arbitrary compact set up to closure of convexification. We argue that learning the support function representation can be attractive compared to other set representation alternatives, such as over-approximation by a geometric primitive fixed a priori, or representing the set of interest as the sub-level set of a suitable (Lyapunov or value) function.

From Ch. 1.6.3, we know that the support function $h_{\mathcal{X}}(\cdot)$ is both convex and positive homogeneous of degree one, and hence a sublinear function. Conversely, any sublinear function can be viewed as support function of a compact set. The converse follows from the fact [177, Thm. 8.13] that a positive homogeneous convex function can be expressed as pointwise supremum of linear function. Thus, to learn a compact set is to learn its support function.

The main idea behind our proposed approach in this chapter is rooted in this isomorphism between the space of sublinear functions and the space of finite dimensional compact convex sets, which allows transcribing set operations of system-control interest to exact functional operations on the corresponding support functions—the latter being computationally amenable via support function calculus.

Learning a support function from data leads to *sublinear regression* as opposed to the well-known *convex regression* [178,179]. Given finite data collected from numerical simulation or from experimental measurements, we outline two learning algorithms: *QP algorithm* in Ch. 6.1.2, and *ISNN algorithm* in Ch. 6.1.3. Both these algorithms perform sublinear regression to learn the corresponding support function representation, i.e., account for the fact that the support functions are not only convex, but also

sublinear (Ch. 1.6.3).

6.1 Learning Algorithms

Following the presented Ideas, we know that learning the support function of a compact set from data results in a sublinear regression problem, i.e., a regression problem where the to-be-learnt function is constrained to be sublinear. To this end, we next detail the data generation procedure followed by two proposed algorithms for the same.

6.1.1 Data generation

For the compact set $\mathcal{X} \subset \mathbb{R}^d$ of interest, we suppose that we have access to n_x i.i.d. samples $\{\widehat{\mathbf{x}}_j\}_{j=1}^{n_x}$, which are realizations of the random vector

$$\{\widehat{\mathbf{x}}_j\}_{j=1}^{n_x} = \{\mathbf{x}_j + \boldsymbol{\nu}_j\}_{j=1}^{n_x} \quad (6.1)$$

where the deterministic but unknown $\mathbf{x} \in \mathcal{X}$, and $\boldsymbol{\nu}_j$ are i.i.d. *random* realizations of some noise vector $\boldsymbol{\nu} \in \mathbb{R}^d$ with zero mean and finite second moment. For instance, if the set \mathcal{X} of interest is a reach set, then we could obtain $\{\widehat{\mathbf{x}}_j\}_{j=1}^{n_x}$ from a numerical simulator encompassing differential and difference equations, constraints, and so forth. In that case, the noise $\boldsymbol{\nu}$ could result from numerical round-off errors. We could also obtain $\{\widehat{\mathbf{x}}_j\}_{j=1}^{n_x}$ from repeated physical experiments, and the noise $\boldsymbol{\nu}$ may stem from the measurement errors.

We proposed to learn the set \mathcal{X} via learning its support function $h_{\mathcal{X}}(\mathbf{y})$, $\mathbf{y} \in \mathbb{S}^{d-1}$, using the training pair $\{\mathbf{y}, \widehat{h}_{\mathcal{X}}(\mathbf{y})\}$ estimated by

$$\widehat{h}_{\mathcal{X}}(\mathbf{y}_i) = \sup_{\widehat{\mathbf{x}} \in \{\widehat{\mathbf{x}}_j\}_{j=1}^{n_x}} \langle \mathbf{y}_i, \widehat{\mathbf{x}} \rangle, \quad \forall i \in \llbracket n_y \rrbracket. \quad (6.2)$$

We seek a sublinear function that “well fits” the values (6.2).

6.1.2 QP algorithm

In this approach, we introduce a regression algorithm that utilizes the standard least squares method. Specifically, we address an *infinite-dimensional* variational problem by solving:

$$\arg \inf_{\{h_{n_x}: \mathbb{R}^d \rightarrow \mathbb{R} | h_{n_x}(\cdot) \text{ is sublinear}\}} \sum_{i=1}^{n_y} (\widehat{h}_{\mathcal{X}}(\mathbf{y}_i) - h_{n_x}(\mathbf{y}_i))^2. \quad (6.3)$$

The least squares problem (6.3) comes with a consistency guarantee, as stated in the following theorem (proof in Appendix F.1).

Theorem 6.1. *The minimizer of (6.3), $h_{n_x}(\cdot)$, almost surely converges to the true support function $h_{\mathcal{X}}(\cdot)$ as $n_y, n_x \rightarrow \infty$.*

The least square estimator h_{n_x} , is achieved by solving the *finite dimensional* convex QP [93, Ch. 6.5.5]

$$\begin{aligned} \arg \min_{\mathbf{g}_1, \dots, \mathbf{g}_{n_y} \in \mathbb{R}^d, \mathbf{h} \in \mathbb{R}^{n_y}} \sum_{i=1}^{n_y} (\widehat{h}_{\mathcal{X}}(\mathbf{y}_i) - h_i)^2, \quad \{\mathbf{y}_i\}_{i=1}^{n_y} \\ \text{subject to } h_j \geq h_i + \langle \mathbf{g}_i, \mathbf{y}_j - \mathbf{y}_i \rangle \quad \forall (i, j) \in \llbracket n_y \rrbracket \times \llbracket n_y \rrbracket. \end{aligned} \quad (6.4)$$

where minimization is over the decision variables $\mathbf{g}_1, \dots, \mathbf{g}_{n_y} \in \mathbb{R}^d$ and $\mathbf{h} := (h_1, \dots, h_{n_y}) \in \mathbb{R}^{n_y}$. The variables $\{\mathbf{g}_i\}_{i=1}^{n_y}$ represent the subgradients of the convex estimator h_{n_x} , and their minimizing value $\mathbf{g}_1^{\text{opt}}, \dots, \mathbf{g}_{n_y}^{\text{opt}}$ from (6.4) provides a piecewise linear (PWL) estimate

$$h^{\text{PWL}}(\cdot) = \max_{i=1, \dots, n_y} \left\{ \widehat{h}_{\mathcal{X}}(\mathbf{y}_i) + \langle \mathbf{g}_i^{\text{opt}}, \cdot - \mathbf{y}_i \rangle \right\}. \quad (6.5)$$

Equation (6.5) represents a convex and degree one positive homogeneous function, therefore $h^{\text{PWL}}(\cdot)$ is sublinear.

6.1.3 ISNN algorithm

Input sublinear neural network (ISNN), i.e a neural network whose output is a sublinear function of the network's input vector, is an alternative tool for performing sublinear regression to learn the support function of the set $\mathcal{X} \in \mathbb{R}^d$ in the noisy setting described in Ch. 6.1.1.

The ISNN is a subcategory of input convex neural network (ICNN) introduced by [180], which is guaranteed to fit any convex function [181]. In other words, the output of an ICNN is convex w.r.t. the input vector $\mathbf{y} \in \mathbb{R}^d$. For clarity, we briefly review the architect of an ℓ layers ICNN. Let the width of each layer be $\{n_1, \dots, n_\ell\}$, and let $n_\ell := 1, n_0 := d$. For all $k \in \llbracket \ell \rrbracket$, the k^{th} layer of the network with width n_k has associated weight matrices $\mathbf{W}_k^{(z)} \in \mathbb{R}_{\geq 0}^{n_k \times n_{k-1}}$, $\mathbf{W}_k^{(y)} \in \mathbb{R}^{n_k \times d}$ and bias vector $\mathbf{b}_k \in \mathbb{R}^{n_k}$. Furthermore, let $\mathbf{W}_1^{(z)} := \mathbf{0} \in \mathbb{R}^{n_1 \times d}$ (zero matrix). Then, for a given input vector $\mathbf{y} \in \mathbb{R}^d$, the computation for each layer of ICNN involves:

$$\begin{aligned} \mathbf{z}_1 &= \sigma \left(\mathbf{W}_1^{(y)} \mathbf{y} + \mathbf{b}_1 \right), \\ \mathbf{z}_{k+1} &= \sigma \left(\mathbf{W}_{k+1}^{(z)} \mathbf{z}_k + \mathbf{W}_{k+1}^{(y)} \mathbf{y} + \mathbf{b}_{k+1} \right), \quad k \in \llbracket \ell - 2 \rrbracket, \\ \mathbf{z}_\ell &= \mathbf{W}_\ell^{(z)} \mathbf{z}_{\ell-1} + \mathbf{W}_\ell^{(y)} \mathbf{y} + \mathbf{b}_\ell, \end{aligned} \tag{6.6}$$

where the vector mapping σ comprises of element-wise application of the same activation function $\sigma(\cdot)$ that is assumed to be convex and non-decreasing. The ICNN model parameters $\mathbf{W}_{2:\ell}^{(z)}$, $\mathbf{W}_{1:\ell}^{(y)}$ and $\mathbf{b}_{1:\ell}$ are determined during the network's training.

The convexity of the ICNN output \mathbf{z}_ℓ w.r.t the input \mathbf{y} follows from two facts: *first*, the summation of convex functions is convex; *second*, a function composition where the inner function is convex and the outer function is convex non-decreasing, remains convex.

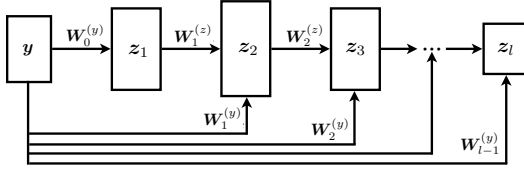


Figure 6.1: Input sublinear neural network (ISNN)

The main distinction between traditional neural networks and ICNN is that the latter necessitate the use of a convex and non-decreasing activation function in their models. The popular choice for such activation function is the rectified linear unit (ReLU) with $\sigma(x) = \max(0, x)$. Fur-

thermore, the weight matrices $\mathbf{W}_{1:\ell}^{(z)}$ that connect the *feedforward* layers must be elementwise non-negative. To compensate for the restriction on non-negative feedforward weights, supplementary *passthrough* links are integrated, linking the input layer to every hidden layer via weight matrices $\mathbf{W}_{1:\ell}^{(y)}$ that may incorporate real values. The bias vectors $\mathbf{b}_{1:\ell}$ are constituted of real entries.

Theorem 6.2. *The neural network (6.6) is an ISNN, i.e., outputs a sublinear function of a given input vector $\mathbf{y} \in \mathbb{R}^d$, if $\mathbf{b}_{1:\ell} = \mathbf{0}$, and the activation function $\sigma(\cdot)$ is convex, non-decreasing and positive homogeneous of degree one.*

The proof of Theorem 6.2 is provided in Appendix F.2. Activation functions $\sigma(\cdot)$ satisfying the conditions delineated in Theorem 6.2 encompass rectified linear units (ReLU), leaky ReLU, and parametric ReLU with a positive parameter. In the numerical simulations presented herein, ReLU activation is utilized.

For all ISNN implementations, five hidden layers with corresponding neuron counts of (5, 20, 50, 20, 5) are used, with mean squared error as the loss metric. The nonconvex training of the ISNN employs the Adam optimization algorithm [182], and the Adam updates are projected to the nonnegative orthant as follows:

$$\mathbf{W}_k^{(z)} \mapsto \text{ReLU}(\mathbf{W}_k^{(z)}) \quad \forall k \in [\ell].$$

6.2 Case Studies

In the following, we present numerical examples to illustrate our proposed approach and draw a comparison with existing algorithms in set estimation. We start with a simple demonstration in Ch. 6.2.1 to learn the support function of the p-sum of ellipsoids from noisy samples and then validate the result with ground truth. In Ch. 6.2.2 the support function of the reach set of an uncertain LTV system is estimated using a sub-linear regressions algorithm. Our results are then tested against a semi-analytical algorithm that approximates the reach set as the intersection of a finite number of ellipsoids. We show in Ch. 6.2.3 how our proposed approach can be utilized in detecting collision between dynamical agents with input uncertainties. In Ch. 6.2.4, we compare the conservatism of the sublinear regressions in estimating the output set of a neural network with the probabilistic semi-definite program that fits an ellipsoid as the output set, proposed by [183]. In our final case study, we learn the region of attraction of a nonlinear system using sublinear regression and discuss how these results differ from the Lyapunov-based region of attraction for the same system.

For all ISNN implementations, we use 5 hidden layers with the respective number of neurons (5, 20, 50, 20, 5). The training data in all of the simulations is of cardinality $(n_x, n_y) = (500, 200)$ unless otherwise is stated, and the unit random vector $\mathbf{y} \in \mathbb{R}^d$ is drawn uniformly from the unit sphere \mathbb{S}^{d-1} . All computations were performed in a MacBook Pro with 2.6 GHz 6-Core Intel i7 processor with 16 GB of memory.

6.2.1 p -Sum of ellipsoids

A nondegenerate ellipsoid of dimension d , center vector $\mathbf{q} \in \mathbb{R}^d$ and shape matrix $\mathbf{Q} \in \mathbb{S}_{++}^d$ is given by

$$\mathcal{E}(\mathbf{q}, \mathbf{Q}) := \{\mathbf{y} \in \mathbb{R}^d \mid (\mathbf{y} - \mathbf{q})^\top \mathbf{Q}^{-1} (\mathbf{y} - \mathbf{q}) \leq 1\}. \quad (6.7)$$

For simplicity, throughout the rest of this thesis, we sometimes refer to $\mathcal{E}(\mathbf{q}_i, \mathbf{Q}_i)$ by \mathcal{E}_i . Denote $\mathcal{X}_{p\text{-sum}}$, as the p -sum of two ellipsoids, \mathcal{E}_1 and \mathcal{E}_2 with centers at the origin and shape matrices $\mathbf{Q}_1 = \text{diag}(1, 14)$ and $\mathbf{Q}_2 = \text{diag}(4, 7)$, respectively. We want to estimate the support function of $\mathcal{X}_{p\text{-sum}}$ via the noisy samples $\{\widehat{\mathbf{x}}_j\}_{j=1}^{n_x}$ using linear regression algorithms introduced above, ISNN and QP. We consider $p = 1$ and $p = \infty$, which are equivalent to the Minkowski sum and the union operations, respectively, as explained in Prop. 5 (see Fig. 6.2.a). The training data, $\{(\widehat{h}_{p\text{-sum}}(\mathbf{y}_i), \mathbf{y}_i)\}_{i=1}^{n_y}$ are obtained using (6.2). We then compare the learned support functions $h_{1\text{-sum}}$ and $h_{\infty\text{-sum}}$ with the analytical solution obtained from Prop. 5.a, and Prop. 5.b, where the support function of each ellipsoid is given by

$$h_{\mathcal{E}(\mathbf{q}, \mathbf{Q})}(\mathbf{y}) = \mathbf{y}^\top \mathbf{q} + (\mathbf{y}^\top \mathbf{Q} \mathbf{y})^{1/2}. \quad (6.8)$$

This comparison is shown in Fig 6.2. We can see a slight overfitting in ISNN (with 30 epochs) which can be reduced by increasing the number of training data, while QP shows a more robust estimation. However, for the same cardinality (n_x, n_y) , the training time for QP is considerably higher (approx. 10 times) than it is for ISNN. We will investigate the training time of different algorithms in more details Ch. 6.2.4.

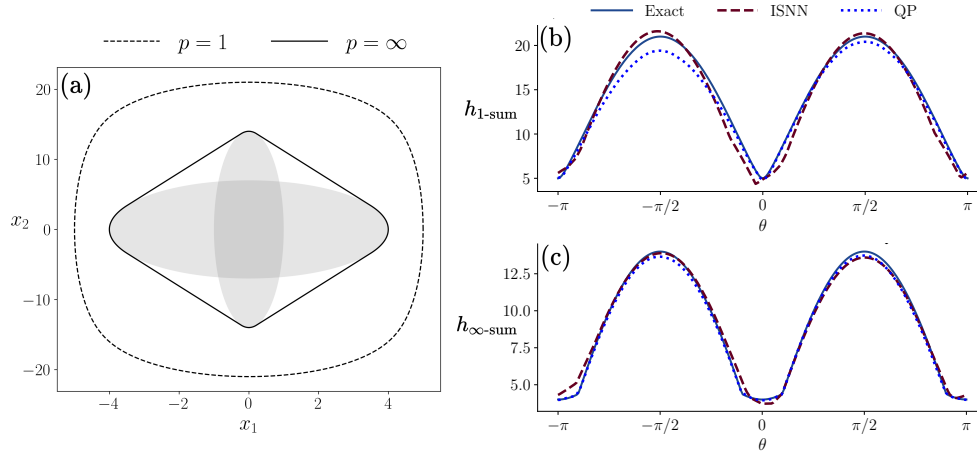


Figure 6.2: The p -sum of two ellipses \mathcal{E}_1 and \mathcal{E}_2 given in Ch. 6.2.1 (filled gray) are shown for $p = 1, \infty$. The dashed boundary is the Minkowski sum ($p = 1$), and the inner solid curve shows the convex hull of the union ($p = \infty$) of \mathcal{E}_1 and \mathcal{E}_2 . (a). The support function of the Minkowski sum ($p = 1$) (b), and the union ($p = \infty$) (c) of two ellipses \mathcal{E}_1 and \mathcal{E}_2 . The exact support function in each figure is shown by *dashed red*. The estimated support function $h_{p\text{-sum}}(\mathbf{y})$ via linear regressions, ISNN and QP are shown with *solid and dashed blue*, respectively.

6.2.2 Reach sets of linear time-varying (LTV) systems

Consider the forward reach set of a linear time-varying system at time t

$$\begin{aligned} \mathcal{X}_{\text{LTV}}(\mathbf{x}_0, t) := \{ & \mathbf{x}(t) \in \mathbb{R}^d \mid \dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x} + \mathbf{B}(t)\mathbf{u} + \mathbf{G}(t)\mathbf{w} \mid \\ & \mathbf{x}(0) \sim \mathcal{N}(\boldsymbol{\mu}_{x_0}, \boldsymbol{\Sigma}_{x_0}), \mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u), \mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w) \}, \end{aligned} \quad (6.9)$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents a normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ and the state vector $\mathbf{x} \in \mathbb{R}^d$, control input $\mathbf{u} \in \mathbb{R}^m$ and unmeasured disturbance $\mathbf{w} \in \mathbb{R}^p$. The system matrices $\mathbf{A}(t)$, $\mathbf{B}(t)$ and $\mathbf{G}(t)$ are continuous in time and of commensurate dimensions.

We want to estimate the support function of $\mathcal{X}_{\text{LTV}}(\mathbf{x}_0, t)$ via approximating the reach set by the intersection of known ellipsoids. We then compare these results with $h(\cdot)$ obtained via ISNN and QP.

Lemma 6.1. Define \mathcal{E}_p as the p -level ($p \in [0, 1]$) confidence region of a vector random

variable $\mathbf{x} \in \mathbb{R}^n$ s.t $\Pr(\mathbf{x} \in \mathcal{E}_p \geq p)$. If $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ be an d -dimensional Gaussian random variable, then

$$\mathcal{E}_p = \{\mathbf{x} \mid (\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq \chi_d^2(p)\}, \quad (6.10)$$

is an p -level confidence set of \mathbf{x} , where $\chi_d^2(p)$ is the quantile function of the chi-squared distribution with d degrees of freedom [183].

Using lemma 6.1, we construct the p -level confidence ellipsoids, of random variables \mathbf{x} , \mathbf{u} and \mathbf{w} as $\mathcal{E}_p^{(x_0)}(\boldsymbol{\mu}_{x_0}, \mathbf{X}_0)$, $\mathcal{E}_p^{(u)}(t)(\boldsymbol{\mu}_u(t), \mathbf{U}(t))$, $\mathcal{E}_p^{(w)}(t)(\boldsymbol{\mu}_w(t), \mathbf{W}(t))$ respectively.

Following the ellipsoidal outer-approximation procedure as in [35], [79, Ch. 3], we generate a family of ellipsoids $\{\mathcal{E}(\mathbf{x}_c(t), \mathbf{X}_i(t))\}_{i=1}^N$ parameterized by unit vectors $\boldsymbol{\ell}_{i0} \in \mathbb{R}^n$ where $i = 1, \dots, N$, such that for any finite $N \in \mathbb{N}$, we have

$$\mathcal{X}_{\text{LTV}}(\mathbf{x}_0, t) \subseteq \widehat{\mathcal{X}}_{\text{LTV}}^{(N)}(\mathbf{X}_0, t) := \bigcap_{i=1}^N \mathcal{E}(\mathbf{x}_c(t), \mathbf{X}_i(t)), \quad (6.11)$$

and $\bigcap_{i=1}^\infty \mathcal{E}(\mathbf{x}_c(t), \mathbf{X}_i(t)) = \mathcal{X}_{\text{LTV}}(\mathcal{X}_0, t)$. Notice that $\widehat{\mathcal{X}}_{\text{LTV}}^{(N)}$ being an intersection of ellipsoids, is guaranteed to be convex but not an ellipsoid in general. The center vector $\mathbf{x}_c(t)$ solves the initial value problem (IVP)

$$\dot{\mathbf{x}}_c = \mathbf{A}(t)\mathbf{x}_c + \mathbf{B}(t)\boldsymbol{\mu}_u + \mathbf{G}(t)\boldsymbol{\mu}_w, \quad \mathbf{x}_c(0) = \boldsymbol{\mu}_{x_0}, \quad (6.12)$$

and the shape matrices $\mathbf{X}_i(t)$ solve the IVPs

$$\begin{aligned} \dot{\mathbf{X}}_i(t) &= \mathbf{A}(t)\mathbf{X}_i(t) + \mathbf{X}_i(t)(\mathbf{A}(t))^\top + \pi_i(t)\mathbf{X}_i(t) + \frac{1}{\pi_i(t)}\mathbf{B}(t)\mathbf{U}(t)\mathbf{B}^\top(t) \\ &\quad - \mathbf{X}_i^{1/2}(t)\mathbf{S}_i(t)\mathbf{G}(t)\mathbf{W}(t)\mathbf{G}^\top(t) - \mathbf{G}(t)\mathbf{W}(t)\mathbf{G}^\top(t)\mathbf{S}_i^\top(t)\mathbf{X}_i^{1/2}(t) \\ \mathbf{X}_i(0) &= \mathbf{X}_0 \end{aligned} \quad (6.13)$$

Let $\boldsymbol{\ell}(t) := \exp(-(\mathbf{A}(t))^\top t)\boldsymbol{\ell}_{i0}$, then an $n \times n$ orthogonal matrix $\mathbf{S}_i(t)$ solves

$$\mathbf{S}_i(t)\widehat{\mathbf{v}}_{2i}(t) = \widehat{\mathbf{v}}_{1i}(t). \quad (6.14)$$

where

$$\widehat{\mathbf{v}}_{1i}(t) := \frac{\mathbf{X}_i^{1/2}(t)\boldsymbol{\ell}_i(t)}{\|\mathbf{X}_i^{1/2}(t)\boldsymbol{\ell}_i(t)\|_2}, \quad \widehat{\mathbf{v}}_{2i}(t) := \frac{\mathbf{G}(t)\mathbf{W}(t)\mathbf{G}^\top(t)\boldsymbol{\ell}_i(t)}{\|\mathbf{G}(t)\mathbf{W}(t)\mathbf{G}^\top(t)\boldsymbol{\ell}_i(t)\|_2}, \quad (6.15)$$

furthermore

$$\pi_i(t) := \left(\frac{\boldsymbol{\ell}_i^\top(t)\mathbf{B}(t)\mathbf{U}(t)\mathbf{B}^\top(t)\boldsymbol{\ell}_i(t)}{\boldsymbol{\ell}_i^\top(t)\mathbf{X}_i(t)\boldsymbol{\ell}_i(t)} \right)^{1/2}. \quad (6.16)$$

An algorithm to compute $\mathbf{S}_i(t)$ in (9) is given in [79, Thm. 4.4.4] using $O(n^2)$ operations. The IVP's (6.12) and (6.13) guarantee $\mathcal{X}_{\text{LTV}}(\mathbf{x}_0, t) \subseteq \widehat{\mathcal{X}}_{\text{LTV}}^{(N)}(\mathbf{X}_0, t)$.

A well-known method to estimate the support function of the intersection of a finite number of outer ellipsoids $\{\mathcal{E}(\mathbf{x}_c(t), \mathbf{X}_i(t))\}_{i=1}^N$ is to compute the minimum volume outer ellipsoid, a.k.a the Löwner-John ellipsoid [107, 108], [184, p. 69] containing $\widehat{\mathcal{X}}_{\text{LTV}}^{(N)}(\mathbf{X}_0, t)$ and then finding the support function of the same using (6.8). However, we adopt a more efficient approach here and find the support function of the intersection directly by utilizing the infimal convolution property, Prop. 4 and (6.8)

$$h_{\widehat{\mathcal{X}}_{\text{LTV}}^{(N)}}(\mathbf{y}) = \inf_{\mathbf{y}_1, \dots, \mathbf{y}_N} \sum_{i=1}^N \mathbf{y}_i^\top \mathbf{x}_c + (\mathbf{y}_i^\top \mathbf{X}_i \mathbf{y}_i)^{1/2}, \quad \text{such that } \mathbf{y}_1 + \dots + \mathbf{y}_N = \mathbf{y}. \quad (6.17)$$

To demonstrate the algorithm presented above, we consider the linearized model of a standard quadrotor dynamics with the 12×1 state vector $\mathbf{x} = (x, y, z, \phi, \theta, \psi, p, q, r, u, v, w)^\top$ comprises of the translational positions (x, y, z) [m], the Euler angles (ϕ, θ, ψ) [rad], the translational velocities (u, v, w) [m/s], and the rotational velocities (p, q, r) [rad/s]. Please refer to [112] for the details on model parameters and matrices \mathbf{A} , \mathbf{B} and \mathbf{G} . We implement a finite horizon LQR controller which acts on imperfect state estimate $\widehat{\mathbf{x}}(t)$ with estimation error $\boldsymbol{\xi}(t) := \mathbf{x}(t) - \widehat{\mathbf{x}}(t)$. The closed loop control is given by

$$\mathbf{u}(\cdot, t) = \mathbf{K}(t)(\cdot) + \mathbf{u}_{\text{feedforward}}(t), \quad (6.18)$$

which tracks the desired path $(x_d(t), y_d(t), z_d(t)) \equiv (\cos t, \sin t, t)$. The LQR con-

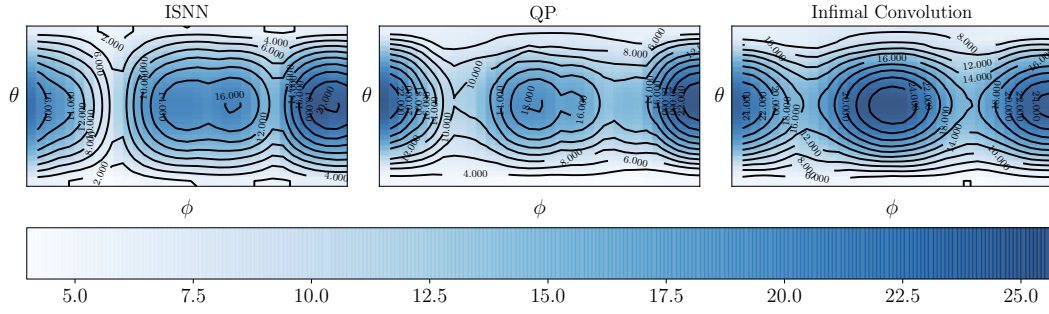


Figure 6.3: The support function of the projection of the quadrator's forward reach set in Ch. 6.2.2 into (x, y, z) coordinates over $(\phi, \theta) \in [0, \pi] \times [0, 2\pi]$ obtained via ISNN, QP and infimal convolution of the support functions associated with the over-approximating outer ellipsoids. The darker (resp. lighter) hues correspond to the higher (resp. lower) widths.

troller acts according to the state cost weight matrix $\mathbf{Q} = \text{blkdiag}(1000\mathbf{I}_3, \text{diag}(1, 1, 10), \mathbf{I}_6)$, the control cost weight matrix $\mathbf{R} = 0.1\mathbf{I}_4$, and the terminal cost weight matrix $\mathbf{M} = \text{blkdiag}(1000\mathbf{I}_3, \mathbf{I}_9)$. The feedback gain $\mathbf{K}(t) = -\mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P}(t)$ where $\mathbf{P}(t)$ solves the associated well known Riccati matrix ODE. Furthermore, $\mathbf{u}_{\text{feedforward}}(t) = \mathbf{R}^{-1}\mathbf{B}^\top\mathbf{v}(t)$ where $\mathbf{v}(t)$ solves a vector ODE with terminal conditions. The closed-loop dynamics can then be written as the linear time-varying system

$$\dot{\mathbf{x}} = \mathbf{A}_{\text{cl}}(t)\mathbf{x} + \mathbf{B}_{\text{cl}}\boldsymbol{\eta} + \mathbf{G}\mathbf{w}, \quad (6.19)$$

where $\mathbf{A}_{\text{cl}} := \mathbf{A} + \mathbf{B}\mathbf{K}(t)$, $\mathbf{B}_{\text{cl}} := \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top$, $\boldsymbol{\eta}(t) := \mathbf{P}(t)\boldsymbol{\xi}(t) + \mathbf{v}(t)$.

We suppose that the initial condition $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}_{12 \times 1}, \text{diag}(0.14, 1.91, 0.53036, 1.04, 0.28, 1.86, 1.33, 0.19, 0.14, 1.43, 1.37)^\top)$. The estimation error $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}_{12 \times 1}, \mathbf{I}_{12})$ and the disturbance $\mathbf{w}(t) \sim \mathcal{N}((\cos t, \sin t, \cos t)^\top, 0.01\mathbf{I}_3)$. Employing lemma 6.1 we construct the corresponding p -level confidence ellipsoids for random variables \mathbf{x}_0 , $\boldsymbol{\xi}$ and \mathbf{w} for $p = 0.85$. Notice that $\boldsymbol{\eta}(t) \in \mathcal{E}(\mathbf{v}(t), \mathbf{V}(t))$ with $\mathbf{V}(t) := \mathbf{P}(t)\mathbf{P}^\top(t)/\chi_d^2(p)$.

For $N = 10$ we solve the IVP's (6.12) and (6.13) for the closed loop dynamic (6.19) and obtain the over-approximating ellipsoids and their corresponding support function using (6.8). The infimal convolution (6.17) then is solved via MATLAB, CVX which

gives $h_{\widehat{\mathcal{X}}_{\text{LTV}}^{(N)}}(\cdot)$, the support function of $\widehat{\mathcal{X}}_{\text{LTV}}^{(10)}(\mathcal{X}_0, t)$. Using the affine transformation property of support functions (Property 1.6.3) we obtain the support function of the projection to the translation states (x, y, z) such that $h_{\text{Proj}(\widehat{\mathcal{X}}_{\text{LTV}}^{(N)})}(\mathbf{P}\mathbf{y}) = h_{\widehat{\mathcal{X}}_{\text{LTV}}^{(N)}}(\mathbf{y})$, where $\mathbf{y} \in \mathbb{R}^{12}$ and the projection matrix $\mathbf{P} = [\mathbf{I}_3, \mathbf{0}_{3 \times 9}] \in \mathbb{R}^{3 \times 12}$.

Now we can visually compare this result with the support function achieved via ISNN and QP for the identical setting. Notice that in this experiment, $\{\widehat{\mathbf{x}}_j\}_{j=1}^{n_x}$ in (6.2) corresponds to the states resulting from different feasible input sample paths. We propagate the trajectories of system (6.19) in 12 dimensional state space and then project the same into the first three coordinates (x, y, z) . The training data for our learning process $\{(\hat{h}_{\text{LTV}}(\mathbf{y}_i), \mathbf{y}_i)\}_{i=1}^{n_y}$ is generated using (6.2).

Fig. 6.3 shows the variation of the support function of the forward reach set projection (6.9) over $(\phi, \theta) \in [0, \pi] \times [0, 2\pi)$ for three different estimation algorithms, ISNN, QP and the infimal convolution of the support functions associated with the over-approximating ellipsoids. As expected, we observe that for the same setting, the sublinear regression algorithms provide a considerably tighter estimation of the set than of infimal convolution algorithm described in this section. Notice that the conservatism in infimal algorithm is pre-determined when choosing N , the number of outer ellipsoids, while the sublinear regression holds no assumptions on the level of conservatism.

6.2.3 Collision detection

Consider the forward reach set of two dynamical agents as $\mathcal{X}_t^A, \mathcal{X}_t^B \subset \mathbb{R}^n$. We want to falsify or certify the intersection between these reach sets at a given time t . We use an intersection oracle, which is based on the support function of the Minkowski sum of the two reach sets, given by [72]

$$\text{opt}(t) = \min_{\mathbf{y} \in \mathbb{S}^{n-1}} \{h_{\mathcal{X}_t^A}(\mathbf{y}) + h_{\mathcal{X}_t^B}(-\mathbf{y})\} \geq (<)0 \iff \mathcal{X}_t^A \cap \mathcal{X}_t^B \neq (=)\emptyset. \quad (6.20)$$

Conditions (6.20) are equivalent to verifying if the zero vector $\mathbf{0} \in \mathbb{R}^d$ belongs to the set of Minkowski difference of $\mathcal{X}_t^A, \mathcal{X}_t^B \subset \mathbb{R}^n$ [64, 65] [64, 65] (see Ch. 4.2 for more details).

In order to show the application of (6.20), we consider two quadrotors, A and B with identical simulation settings as in Ch. 6.2.2, and different initial conditions $\mathbf{x}_{A,0} \sim \mathcal{N}([15, -20, -3, \mathbf{0}_{1 \times 1} 9]^\top, \text{diag}(0.62, 0.96, 0.03, 0.27, 0.42, 0.43, 0.88, 0.62, 0.19, 0.17, 0.80, 0.24)^\top)$ and $\mathbf{x}_{B,0} \sim \mathcal{N}([-10.5, 2, 8, \mathbf{0}_{1 \times 1} 9]^\top, \text{diag}(1, 0.77, 0.28, 0.53, 0.27, 0.57, 0.20, 0.45, 0.35, 0.59, 0.63, 0.15)^\top)$. without loss of generality, we ignore the effect of the unmeasured disturbance vector \mathbf{w} in this experiment.

We want to investigate if there will be a collision between the two agents at any time during $[0, 3]$ s. To this end, we propagate the trajectories of system (6.19) in 12 dimensional state space for both agents A and B and then project the same into the first three coordinates (x, y, z) [m]. We generate the training data using (6.2) and learn the support functions $h_{\text{Proj}(\mathcal{X}_t^A)}(\cdot), h_{\text{Proj}(\mathcal{X}_t^B)}(\cdot)$ via ISNN with 50 epochs and QP and compute the minimum value of (6.20) for both algorithms at each time instant $\{t_i\}_{i=1}^{150}$ over the test unit vectors $\{\mathbf{y}_i\}_i^{n=900}$. We then also solve (6.20) for the support functions $h_{\text{Proj}(\widehat{\mathcal{X}}_{\text{LTV}}^{10,A})}(\cdot), h_{\text{Proj}(\widehat{\mathcal{X}}_{\text{LTV}}^{10,B})}(\cdot)$ estimated via the ellipsoidal over-approximation technique (followed by infimal convolution) detailed in this section, for $p = 0.75$.

Figure 6.4 reveals that the infimal convolution provides the most conservative estimate for the collision time $t = 1.04$ s. When using ISNN, the first collision between $\mathcal{X}_t^A, \mathcal{X}_t^B \subset \mathbb{R}^n$ occurs at $t = 1.22$ s, however, the optimal value keeps alternating sign afterward from negative to positive at $[1.22, 1.34, 1.48, 2.18, 2.22, 2.26, 2.32, 2.38, 2.40, 2.46, 2.54, 2.62, 2.76, 2.90, 2.980]$ s time instances, which each indi-

cates intersection following a separation state of the sets. Finally, we observe that the intersection for QP happens at $t = 2.26$ s, and the optimal value remains positive afterward. The monotone increase of the optimal value of (6.20) when using the infimal convolution and QP algorithm is due to the fact that both the quadrotors track the same desired trajectory. So as time passes, the two reach sets get closer, and the intersection region grows larger. However, this growth is considerably faster for the infimal algorithm as the uncertainty, and consequently, the volume of the outer ellipsoids grows with time. Although the ISNN and QP use the same training data, we do not observe the same monotone growth in the optimal value when using ISNN. This is due to an important distinction between the two sublinear regresses. While QP achieves a deterministic trajectory, ISNN training involves randomization both in Adam algorithm and epochs, which explains the noisy behavior of the ISNN trajectory.

Although ISNN is considerably faster, Fig. 6.4 convinces us that the QP is the best choice for detecting a collision. The high computational time of solving QP in CVX, however, motivates us to design a custom algorithm for solving QP in order to cut down the computation time in our future research.

Notice that in this example, at each time instant, we actually only need to detect the sign change of the n dimensional vector created by estimating $\text{opt}(t)$ for $\{\mathbf{y}\}_1^{n=900}$. If a sign change exists, this process requires less computational time than of finding the minimum value in (6.20), which is between $\mathcal{O}(n)$ and $\mathcal{O}(\log(n))$, depending on the level of parallelization.

6.2.4 Reach set of deep neural network

In the present section, we learn the support function of the reach set of a neural network in a probabilistic setting. In this setting, the input to the network, $\mathbf{x} \in \mathbb{R}^d$ has

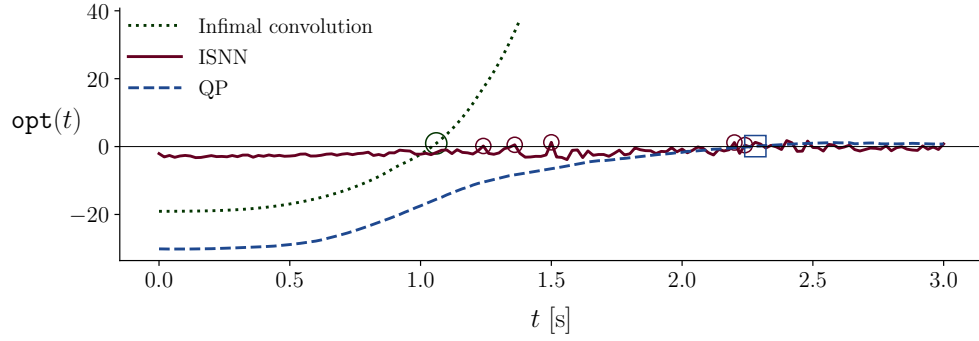


Figure 6.4: Collision detection between the quadrotors A and B in Ch. 6.2.3 over the time horizon $[0, 3]$ s. The vertical axis represents $\text{opt}(t) = \min_{\mathbf{y} \in \mathbb{S}^{n-1}} \{h_{\text{Proj}(\mathcal{X}_t^A)}(\mathbf{y}) + h_{\text{Proj}(\mathcal{X}_t^B)}(-\mathbf{y})\}$ obtained via infimal convolution (dotted dark green), ISNN (solid maroon) and QP (dashed blue).

Instance	ISNN, 30 epochs	QP	SDP
1	6.80 s	61.38 s	5.03 s
2	6.76 s	60.39 s	4.71 s
3	6.66 s	63.54 s	4.97 s
4	6.73 s	67.02 s	5.48 s
5	6.78 s	66.55 s	5.03 s
6	7.80 s	70.32 s	5.05 s
7	6.76 s	73.82 s	4.97 s
8	6.66 s	73.86 s	4.92 s
9	6.79 s	66.32 s	5.34 s
10	6.65 s	68.73 s	5.00 s

Table 6.1: Computational times [s] incurred by the sublinear regressions ISNN, QP vs the SDP solver for the example in Ch. 6.2.4 with $n_l = 30$ for 10 different random sampling instances.

random and potentially unbounded uncertainty, and we want to estimate the output set \mathcal{S} of the neural network $f(\mathbf{x})$, such that $f(\mathbf{x}) \subseteq \mathcal{S}$.

If \mathcal{E}_p is the p -level ($p \in [0, 1]$) confidence region of a vector random variable $\mathbf{x} \in \mathbb{R}^d$, then \mathcal{S} is a p -level confidence region for the random variable $f(\mathbf{x})$, i.e. $\Pr(f(\mathbf{x}) \in \mathcal{S}) \geq p$ if $f(\mathcal{E}_p) \subseteq \mathcal{S}$ [183].

Given \mathcal{E}_p as in (6.10), Fazylab et al. [183] introduced a SDP relaxation of the nonox-

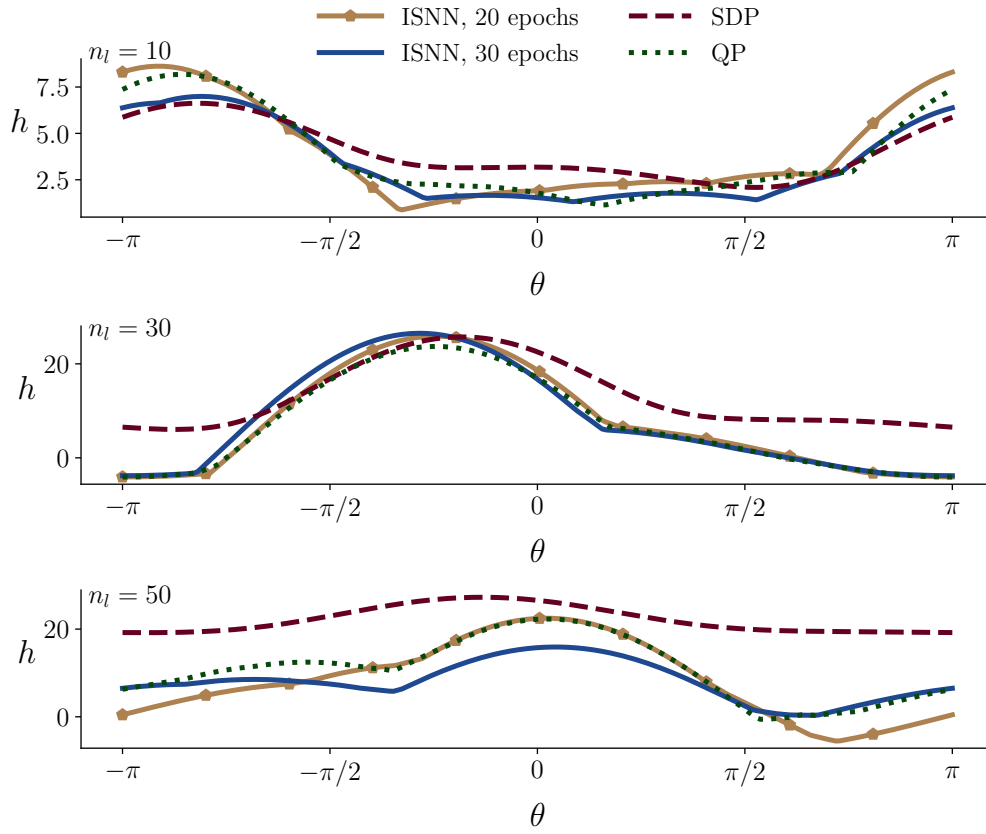


Figure 6.5: The estimated support function of the output reach set of NN (6.22) Via SDP (6.26) (dashed red), QP (dotted green), ISNN with 20 (solid mustard) and 30 (solid blue). All subfigures are plotted over the polar coordinate $\theta \in (-\pi, \pi]$. Different number of neurons for the single hidden layer are indicated in the respective subfigures.

convex optimization problem

$$\min \text{Vol}(\mathcal{S}) \quad \text{subject to} \quad f(\mathcal{E}_p) \subseteq \mathcal{S}, \quad (6.21)$$

where \mathcal{S} is the safe output ellipsoid of a n_l layer feed-forward fully-connected neural network

$$\mathbf{x}_0 = \mathbf{x} \quad (6.22a)$$

$$\mathbf{x}_{k+1} = \sigma(\mathbf{W}_k \mathbf{x}_k + \mathbf{b}_k), \quad k = 0, \dots, l-1 \quad (6.22b)$$

$$f(\mathbf{x}) = \mathbf{W}_l \mathbf{x}_l + \mathbf{b}_l, \quad (6.22c)$$

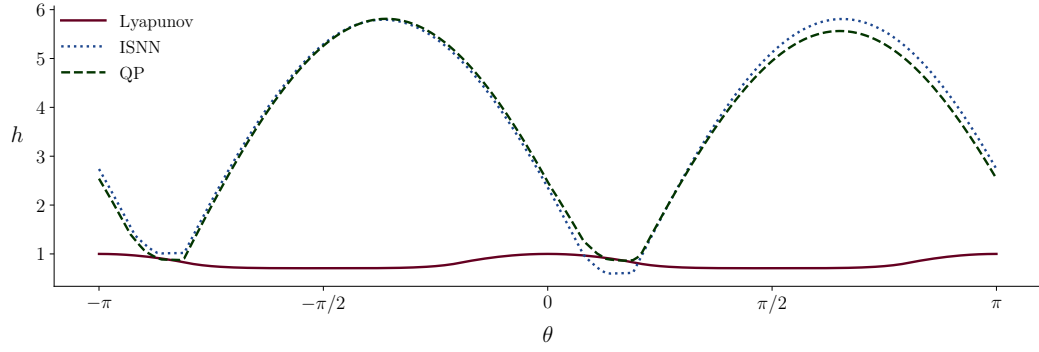


Figure 6.6: The support function of the ROA of the stable fixed point $\mathbf{x}^* = (0, 0)$ of the nonlinear system (6.28) using Lyapunov theory (solid red), ISNN (dashed blue) and QP (dashed green). The estimated ROA via sublinear regressions, ISNN and QP is considerably less conservative than the Lyapunov-Based ROA. The height of the peaks of the support function of the ROA in ISNN, and QP depends on how farther away from the origin the sample data $\{\mathbf{x}_j\}_{j=1}^{n_x}$ are taken (refer to Fig.6.7).

with $\sigma := \text{ReLU}$.

Lemma 6.2 ([183]). *The ReLU function, $\sigma(\mathbf{x}) = \max(\mathbf{0}, \mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^d$, satisfies the QC*

$$\begin{bmatrix} \mathbf{x} \\ \sigma(\mathbf{x}) \\ 1 \end{bmatrix}^\top \mathbf{Q} \begin{bmatrix} \mathbf{x} \\ \sigma(\mathbf{x}) \\ 1 \end{bmatrix} \geq 0 \quad (6.23)$$

defined by \mathbf{Q} where

$$\mathbf{Q} = \left\{ \mathbf{Q} \mid \mathbf{Q} = \begin{bmatrix} \mathbf{0} & \mathbf{T} & -\boldsymbol{\nu} \\ \mathbf{T} & -2\mathbf{T} & \boldsymbol{\nu} + \boldsymbol{\eta} \\ -\boldsymbol{\nu}^\top & \boldsymbol{\nu}^\top + \boldsymbol{\eta}^\top & \mathbf{0} \end{bmatrix} \right\} \quad (6.24)$$

Here $\boldsymbol{\eta}, \boldsymbol{\nu} \geq 0$ and $\mathbf{T} \in \mathbb{S}_+^d$ is given by

$$\mathbf{T} = \sum_{i=1}^d \lambda_i \mathbf{e}_i \mathbf{e}_i^\top + \sum_{i=1}^{d-1} \sum_{j>i}^d \lambda_{ij} (\mathbf{e}_i - \mathbf{e}_j) (\mathbf{e}_i - \mathbf{e}_j)^\top \quad (6.25)$$

where \mathbf{e}_i is the i -th basis vector in \mathbb{R}^d and $\lambda_{ij} \geq 0$.

Theorem 6.3 ([183]). *Consider a single layer neural network $f : \mathbb{R}^d \rightarrow \mathbb{R}^{n_z}$ described*

by the equation

$$z = \mathbf{W}^{(1)} \sigma(\mathbf{W}^{(0)} \mathbf{x} + \mathbf{b}^0) + \mathbf{b}^{(1)}$$

where $\sigma : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_1}$ satisfies the quadratic constraint defined by (6.24) and $\mathbf{x} \in \mathcal{E}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$. Consider the following matrix inequality

$$\begin{aligned} \mathbf{M}_1 + \mathbf{M}_2 + \mathbf{M}_3 \leq \mathbf{0}, \quad \text{where} \quad \mathbf{M}_1 = \begin{bmatrix} \mathbf{W}^{0\top} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_1} & \mathbf{0} \\ \mathbf{b}^{0\top} & \mathbf{0} & 1 \end{bmatrix} \mathbf{Q} \begin{bmatrix} \mathbf{W}^0 & \mathbf{0} & b^0 \\ \mathbf{0} & \mathbf{I}_{n_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}, \\ \mathbf{M}_2 = \begin{bmatrix} \mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{P}(\tau) \begin{bmatrix} \mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}^\top, \quad \mathbf{M}_3 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{W}^{1\top} & \mathbf{0} \\ \mathbf{b}^\top & 1 \end{bmatrix} \mathbf{S}(\mathbf{A}, \mathbf{b}) \begin{bmatrix} \mathbf{0} & \mathbf{W}^1 & \mathbf{b}^1 \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \end{aligned} \quad (6.26)$$

with

$$\mathbf{P}(\tau) = \tau \begin{bmatrix} -\boldsymbol{\Sigma}_x^{-1} & \boldsymbol{\Sigma}_x^{-1} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_x^\top \boldsymbol{\Sigma}_x^{-1} & -\boldsymbol{\mu}_x^\top \boldsymbol{\Sigma}_x^{-1} \boldsymbol{\mu}_x + 1 \end{bmatrix}, \quad \mathbf{S}(\mathbf{A}, \mathbf{b}) = \begin{bmatrix} \mathbf{A}^2 & \mathbf{A}\mathbf{b} \\ \mathbf{b}^\top \mathbf{A} & \mathbf{b}^\top \mathbf{b} - 1 \end{bmatrix}. \quad (6.27)$$

If (6.26) is feasible for some $(\tau, \mathbf{A}, \mathbf{Q}, \mathbf{b}) \in \mathbb{R}_+ \times \mathbb{S}^{n_z} \times \mathbf{Q} \times \mathbb{R}^{n_z}$, then $z \in \mathcal{E}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$ with $\boldsymbol{\mu}_z = -\mathbf{A}^{-1} \mathbf{b}$ and $\boldsymbol{\Sigma}_z = \mathbf{A}^{-2}$.

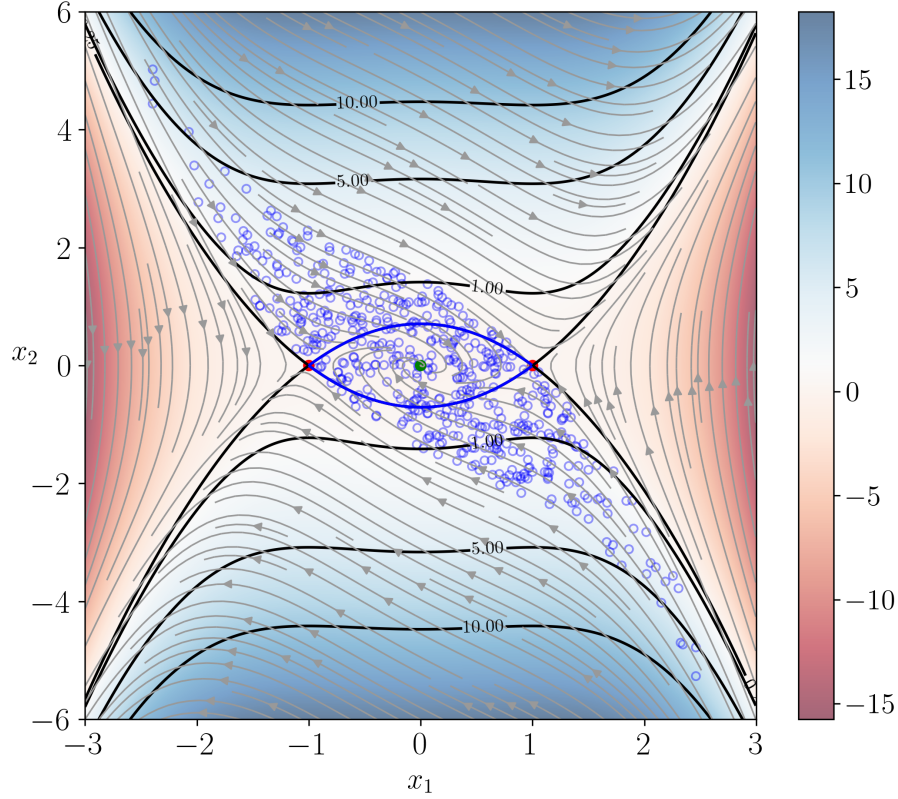


Figure 6.7: The ROA of the stable fixed point $\mathbf{x}^* = (0, 0)$ (shown with *green dot*) of the nonlinear system (6.28); the *solid black* lines shows 4 contours for positive c of the function $V(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2) - \frac{1}{4}x_1^4$, superimposed with the given vector field. *Blue (red)* color denotes region where V is $> 0 (< 0)$. The open set \mathcal{D} is the infinite vertical strip strictly inside the dashed green lines $x_1 = \pm 1$. The Lyapunov estimate for the region of attraction is $(x_1, x_2) \in \mathbb{R}^2 : V(x_1, x_2) \leq c$ where $0 < c < 1/4$ (the "eye-shaped" set inside the *solid blue* boundary) which is also the heteroclinic orbit connecting the two saddle points (shown with *red dots*). The *scattered blue* samples form the ROA which is not fully covered by the Lyapunov-based ROA.

We compare the support function of the output ellipsoid obtained by SDP (6.26) with the learned support function using regression algorithms QP and ISNN with 20 and then 30 epochs. The same simulation setting as [183] is adopted where $\boldsymbol{\mu} = (1, 1)$, $\boldsymbol{\sigma} = \text{diag}(1, 2)$ and for $p = 0.95$. The number of neurons for each layer is $\{2, n_l, 2\}$, respectively and we ran the experiment for $n_l = 10, 30, 50$. The training data is generated as (6.2) with $(n_x, n_y) = (10000, 200)$. As expected from the observation made in

Ch. 6.2.2, Fig. 6.5 reveals that the support function learned by either ISNN and QP provides a tighter estimation of the output reach set of NN (6.22) compare to the output ellipsoid given by SDP (6.26). Farther more we see that as the number of epochs for ISNN increases, the ISNN estimates approach the QP estimate of the support function. While QP provides a more robust estimate, ISNN is more time efficient. For 10 different random sampling instances with fixed cardinality and identical simulation setting, table 6.1 dissertations a higher (approx. 10 times) training time for the QP than that of ISNN. The training time for ISNN with 30 epochs is comparable of the computation time for solving SDP (6.26) while keeping all other simulation settings fixed **although we don't have probabilistic inclusion guarantee.**

6.2.5 ROA of nonlinear systems

The region of attraction (ROA) of a stable fixed point \mathbf{x}^* of a nonlinear system is defined as an invariant subset of its domain \mathcal{D} , such that all the trajectories starting inside this set will converge (asymptotically) to \mathbf{x}^* .

An estimate for the ROA follows from the Lyapunov theorem and LaSalle's invariance principle. Any sublevel sets of the lyapunov function $V(\mathbf{x})$ defined on the domain around \mathbf{x}^* is an invariant set of the form $\Omega_c = \{(\mathbf{x}) \in \mathbb{R}^d : V(\mathbf{x}) \leq c\}$, where an upper bound for the level $c > 0$ gives the largest estimate for the ROA, such that Ω_c is compact, positively invariant, and $\Omega_c \subset \mathcal{D}$.

Consider the nonlinear system

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -x_1 - x_2 + x_1^3 \quad (6.28)$$

The proposed Lyapunov function for the stable fixed point $\mathbf{x}^* = (0, 0)$ is $V(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2) - \frac{1}{4}x_1^4$ which gives $1 < c < 1/4$. In Fig. 6.6, we draw attention to the gap between the support function of the Lyapunov based ROA and those obtained via sub-

linear regressions, QP and ISNN with $(n_x, n_y) = (385, 300)$. The region of attraction learned by the sublinear regressions are considerably larger than the one obtained using Lyapunov theory. We can see a slight overfitting by the ISNN that results in crossing the Lyapunov ROA, which can be resolved by increasing the training data. The height of the peaks of the support function of the ROA in ISNN, and QP depends on how farther away from the origin the sample data $\{\mathbf{x}_j\}_{j=1}^{n_x}$ are taken such that in the limit, the two peaks in the support function will stretch out to infinity. This can be seen intuitively in Fig.6.7. In this experiment first we sample the initial states $x_1, x_2 \sim \mathcal{N}(0, 2)$ and only keep the samples for which the corresponding trajectory converge asymptotically to the origin.

6.2.6 Control invariant set

Consider a controlled dynamical system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x} \in \mathbb{R}^d, \quad \mathbf{u} \in \mathcal{U} \subset \mathbb{R}^m, \quad (6.29)$$

where \mathcal{U} is the set of admissible control. A set $\mathcal{C} \subset \mathbb{R}^d$ is a *control-invariant* (CI) for (6.29), if for any initial condition $\mathbf{x}_0 \in \mathcal{C}$, there exist a control trajectory $\{\mathbf{u}(t)\}_{t=0}^{\infty}$ such that dynamic (6.29) stays inside \mathcal{C} for all $t \geq 0$ [10].

In practical applications, the safety of the system is usually guaranteed by adhering to specific state constraints that define a safe region $\mathcal{S} \subset \mathbb{R}^d$. One popular way to achieve this goal is to identify a control-invariant set such that $\mathcal{C} \subset \mathcal{S}$.

To minimize system's conservatism, it is desirable to find the *maximum control-*

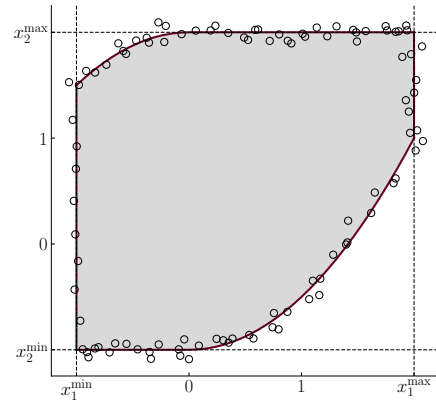


Figure 6.8: The MCI set for integrator dynamic (6.30). The boundary of the exact MCI and the its noisy samples are shown by *solid red* and *black circles*, respectively.

invariant (MCI) set $\bar{\mathcal{C}}$, that encompasses all other control-invariant sets $\mathcal{C} \subset \mathcal{S}$.

While determining maximal control-invariant sets for arbitrary systems poses an intractable challenge, a commonly used approach to represent such sets is through super-level sets of a function $g : \mathcal{R}^d \rightarrow \mathcal{R}$ such that $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^d \mid g(\mathbf{x}) \geq 0\}$. In this case study, we like to test the performance of sublinear regression algorithms by identifying the MCI of the 2-dimensional integrator system under boxed-valued input and state constraints \mathcal{U} and \mathcal{E} , given by

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = u, \quad (6.30a)$$

$$\mathcal{E}(\underline{\mathbf{x}}, \bar{\mathbf{x}}) = \{x_i \in \mathbb{R}^n : \underline{x}_i \leq x_i \leq \bar{x}_i, \quad i = 1, 2\} \quad (6.30b)$$

$$\mathcal{U}(\underline{u}, \bar{u}) = \{u \in \mathbb{R} : \underline{u} \leq u \leq \bar{u}, \quad \underline{u} \leq 0 \leq \bar{u}\} \quad (6.30c)$$

We learn the support function of the desired CI using (noisy) samples derived from simulation, and validate the result via the analytical expression of the double integrator MCI obtained by [185]. For $\mathcal{E}([-1, -1], [2, 2])$ and $\mathcal{U}(-1, 1)$ the 2-dimensional integrator MCI along with noisy samples is depicted in Fig. 6.8. Figure 6.9 shows the

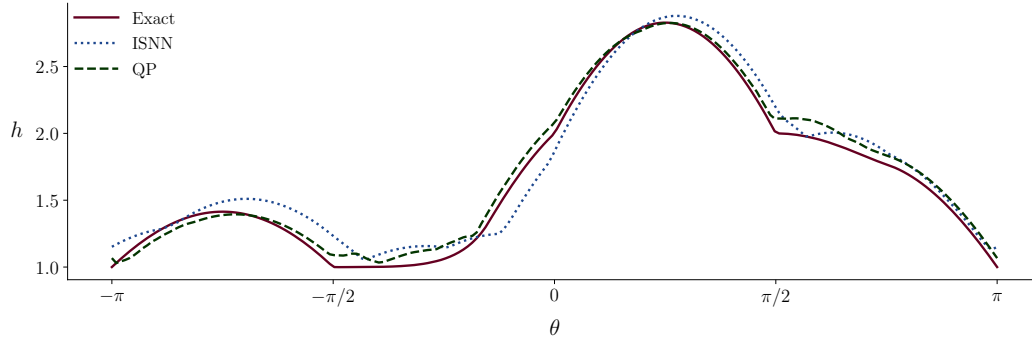


Figure 6.9: The support function of the MCI of the double integrator (6.30) with state and input constraints $\mathcal{E}([-1, -1], [2, 2])$ and $\mathcal{U}(-1, 1)$ over $\theta \in [-\pi, \pi]$. The (dashed blue) and (dashed green) curves correspond to the learned support function using ISNN and QP respectively. The (solid red) shows the support function of the exact MCI (refer to Fig.6.8).

support function of the MCI over $\theta \in [-\pi, \pi]$ via ISNN, QP and the analytical solution

for the double integrator system. We observe that, while having limited access to the sample data, the sublinear regressions provide a considerably close estimation exact support function.

7 | Summary and Future Work

7.1 Summary

Our research initiates a systematic study on learning the reach sets of full state feedback linearizable systems subject to set-valued uncertainties. These reach sets are, in general, compact but nonconvex. Our learning strategy relies on computing the reach sets in the associated integrator coordinates, and then transforming the sets back to the original coordinates via known diffeomorphisms.

To realize the aforesaid research plan, we first investigate the exact geometry of the integrator reach sets under set-valued uncertainty, which are known to be compact and convex. We show that under time-invariant set-valued input uncertainty, the integrator reach set is semialgebraic (Ch. 2.1.2), a zonoid up to translation (Ch. 2.1.1), but not a spectrahedron (Ch. 2.1.5). We derive the equation of its boundary in both parametric (Theorem 2.3) and implicit forms (Ch. 2.1.3). We also deduce the closed form formula for the volume (Ch. 2.2.1) and diameter (Ch. 2.2.2) of these reach sets. Furthermore, we derive the scaling laws (Ch. 2.2.3 and 2.2.4) for these quantities, thereby quantifying how the “size” of these sets grow under the combined dependence on time and the state space dimension. We point out that these results may be used to benchmark the performance of set over-approximation algorithms (Ch. 2.3).

In Ch. 3, we extend our results to compute integrator reach sets under time-varying compact set-valued uncertainties in the input. Such uncertainties appear in differentially flat systems. We derive the explicit formula for the support function (3.1.1) and boundaries (3.2.1) of the integrator reach sets under time-varying compact set-valued uncertainties in the inputs.

Based on the previous findings, we introduce a semi-analytical approach for cal-

culating LTI reach sets (Section 4.1). Additionally, we present an algorithm designed to either falsify or verify the intersection of multiple integrator reach sets, and consequently, the intersection of reach sets for the corresponding differentially flat systems (Section 4.2). Finally, we unveil an algorithm for learning the compact, albeit possibly nonconvex, reach sets for differentially flat systems (Section 4.3).

Finally, we provide a computational complexity analysis (Ch. 4.6) and show that by parallelizing the learning algorithm proposed in Ch. 4.3, fast numerical computation can be achieved.

In order to quantify the conservatism in our reach set estimation, we study the Hausdorff distance between the reach set of LTI systems under different norm ball-valued input uncertainties. Specifically, we study the Hausdorff distance between two different norm balls in an Euclidean space and derive closed-form formula for the same. In d dimensions, we provide results for the ℓ_p norm balls parameterized by p where $1 \leq p \leq \infty$, as well as for the polyhedral D -norm balls parameterized by k where $1 \leq k \leq d$. We then investigate a more general setting: the Hausdorff distance between two convex sets obtained by transforming two different ℓ_p norm balls via a given linear map. In this setting, while we do not have a general closed-form formula, we provide upper bounds for the Hausdorff distance or its expected value depending on whether the linear map is arbitrary or random. Our results make connections with the literature on hypercontractive operator norms, and on the norms of random linear maps. We then focus on a further generalization: the Hausdorff distance between two set-valued integrals obtained by applying a parametric family of linear maps to the unit ℓ_p norm balls, and then taking the Minkowski sums of the resulting sets in a limiting sense. As an illustrative application, we show that the problem of computing the Hausdorff distance between the reach sets of a linear time-varying dynamical system with different unit ℓ_p norm ball-valued input uncertainties, leads to this set-valued

integral setting.

In Ch. 6, we propose data-driven learning of compact sets in general, and reach sets in particular, by learning the corresponding support function representations. We point out an equivalence between the support functions and the class of sublinear functions, and propose leveraging the same for performing sublinear regression. We numerically demonstrate and compare two approaches: the *first* involves convex quadratic programming (QP), and the *second* being ISNN that involves nonconvex programming. Our numerical experiments reveal that among the two, the ISNN is numerically faster but the QP solution is more robust and comes with consistency guarantee. We empirically observe that ISNN with modest number of epochs can be a practical alternative to QP without incurring as much computational cost as the latter.

7.2 Future Work

For future work, we propose a number of useful and interesting extensions of our research listed below.

[i] (Generic dynamic state feedback linearizable systems)

In this dissertation, we have only considered a class of dynamic state feedback linearizable systems, where the compensator dynamics is affine in control. A relevant extension of the work is analyzing the reach sets of dynamic state feedback linearizable systems where the compensator dynamics may also depend on the time derivatives of the control.

[ii] (Collision avoidance applications) It is of interest to explore how intersection detection between integrator agents can be utilized in collision avoidance applications. Specifically, it carries a considerable curiosity to know how our

results in Ch. 4.2 can be exploited for collision detection in the presence of set-valued uncertainties, when these integrator dynamics correspond to the normal forms of differentially flat systems, such as in vehicular dynamics.

[iii] (Partial state feedback linearizable systems) Many control systems of practical importance are not full state feedback linearizable. However, some of them can be transformed to two interconnected subsystems [186, page 264]: a linear subsystem and a nonlinear subsystem called the zero dynamics [43]. This is known as partial state feedback linearization. Developing an efficient learning algorithm for estimating the reach sets of partial state feedback linearizable systems can be a remarkable extension of this work.

[iv] (Hausdorff distance of ℓ_p norm balls) It will be interesting to explore the qualitative properties of the nonconvex landscape of the Hausdorff distance between the ℓ_p norm balls composed with linear maps (5.12), and to design efficient algorithms in computing the Hausdorff distance for the same.

[vi] (Customized QP algorithm) In Chapter 6, we tackled the quadratic problem at hand by employing semi-definite programming (SDP) methods. It should be possible to design a customized algorithm tailored to solve the given QP in a more efficient manner.

A | Proofs for Chapter 1

A.1 Proof of Lemma 1.1

We know that $\mathcal{K}_i \rightarrow \mathcal{K}$ iff $\delta_H(\mathcal{K}_i, \mathcal{K}) \rightarrow 0$. From Property 1.6.3 of support functions, the latter is equivalent to $h_{\mathcal{K}_i}(\cdot) \rightarrow h_{\mathcal{K}}(\cdot)$.

A.2 Proof of Lemma 1.2

Follows from continuity of the volume functional [47, p. 55], and uniform convergence of the support functions of corresponding sets.

A.3 Proof of Lemma 1.3

For any point-to-set function $F(\cdot)$, we define

$$\int_0^t F(s) ds := \lim_{\Delta \downarrow 0} \sum_{i=0}^{\lfloor \frac{t}{\Delta} \rfloor} \Delta F(i\Delta), \quad (\text{A.1})$$

where the summation symbol Σ stands for the Minkowski sum, and $\lfloor \cdot \rfloor$ is the floor operator; see e.g., [3]. The, for any $\mathbf{y} \in \mathbb{R}^d$, we have

$$\begin{aligned} h_{\int_0^t F(s) ds}(\mathbf{y}) &\stackrel{(\text{A.1})}{=} h_{\lim_{\Delta \downarrow 0} \sum_{i=0}^{\lfloor \frac{t}{\Delta} \rfloor} \Delta F(i\Delta)}(\mathbf{y}) = \sup_{\mathbf{x} \in \lim_{\Delta \downarrow 0} \sum_{i=0}^{\lfloor \frac{t}{\Delta} \rfloor} \Delta F(i\Delta)} \langle \mathbf{y}, \mathbf{x} \rangle \\ &\stackrel{(\text{Lemma 1.1})}{=} \lim_{\Delta \downarrow 0} \sup_{\mathbf{x} \in \sum_{i=0}^{\lfloor \frac{t}{\Delta} \rfloor} \Delta F(i\Delta)} \langle \mathbf{y}, \mathbf{x} \rangle = \lim_{\Delta \downarrow 0} h_{\sum_{i=0}^{\lfloor \frac{t}{\Delta} \rfloor} \Delta F(i\Delta)}(\mathbf{y}) \\ &= \lim_{\Delta \downarrow 0} \sum_{i=0}^{\lfloor \frac{t}{\Delta} \rfloor} \Delta h(i\Delta, \mathbf{y}) = \int_0^t h(s, \mathbf{y}) ds, \end{aligned}$$

wherein the last but one line used the properties (1.6.3)-(1.6.3) for the support function.

A.4 Proof of Proposition 1.1

Since support function is distributive over sum, we have

$$h_{\mathcal{R}(\mathcal{X}_0, t)}(\mathbf{y}) = \sup_{\mathbf{x}_0 \in \mathcal{X}_0} \langle \mathbf{y}, \exp(t\mathbf{A})\mathbf{x}_0 \rangle + h_{\int_0^t \exp((t-s)\mathbf{A})\mathbf{B}\mathcal{U} ds}(\mathbf{y}). \quad (\text{A.2})$$

The block diagonal structure of the matrix \mathbf{A} in (1.7) implies

$$\sup_{\mathbf{x}_0 \in \mathcal{X}_0} \langle \mathbf{y}, \exp(t\mathbf{A})\mathbf{x}_0 \rangle = \sup_{\mathbf{x}_0 \in \mathcal{X}_0} \sum_{j=1}^m \langle \mathbf{y}_j, \exp(t\mathbf{A}_j)\mathbf{x}_{j0} \rangle. \quad (\text{A.3})$$

Following the definition of support function and [78, Proposition 1], we then have

$$\begin{aligned} h_{\int_0^t \exp((t-s)\mathbf{A})\mathbf{B}\mathcal{U} ds}(\mathbf{y}) &= \int_0^t h_{\exp(\mathbf{A})\mathbf{B}\mathcal{U}}(\mathbf{y}) ds = \int_0^t \sup_{\mathbf{u} \in \mathcal{U}} \langle \mathbf{y}, \exp((t-s)\mathbf{A})\mathbf{B}\mathbf{u} \rangle ds \\ &= \int_0^t \sup_{\mathbf{u} \in \text{closure}(\text{conv}(\mathcal{U}))} \sum_{j=1}^m \{ \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle u_j \} ds. \end{aligned} \quad (\text{A.4})$$

The last equality in (A.4) follows from (2.5), and from the fact [71, Prop. 6.1] that the reach set remains invariant under the closure of convexification of the input set \mathcal{U} .

Substituting (A.3) and (A.4) in (A.2) yields (1.32).

B | Proofs for Chapter 2

B.1 Proof of Theorem 2.1

Since the uncertainties in (2.1) along different input co-ordinate axes are mutually independent, the support function of the reach set is of the form (2.9). Therefore, in this case, (1.32) takes the form

$$h_{\mathcal{R}(\mathcal{X}_0, t)}(\mathbf{y}) = \sum_{j=1}^m \left\{ \sup_{\mathbf{x}_{j0} \in \mathcal{X}_{j0}} \langle \mathbf{y}_j, \exp(t\mathbf{A}_j) \mathbf{x}_{j0} \rangle + \int_0^t \sup_{u_j \in [\alpha_j, \beta_j]} \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle u_j \, ds \right\}. \quad (\text{B.1})$$

The optimizer u_j^{opt} of the integrand in the RHS of (B.1), for $j \in [m]$, can be written in terms of the Heaviside unit step function $H(\cdot)$ as

$$u_j^{\text{opt}} = \alpha_j + (\beta_j - \alpha_j) H(\langle \mathbf{y}_j, \boldsymbol{\xi}_j \rangle) = \alpha_j + (\beta_j - \alpha_j) \times \frac{1}{2} (1 + \text{sgn}(\langle \mathbf{y}_j, \boldsymbol{\xi}_j \rangle)),$$

where $\text{sgn}(\cdot)$ denotes the signum function. Therefore,

$$\sup_{u_j \in [\alpha_j, \beta_j]} \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle u_j = \nu_j \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle + \mu_j |\langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle| \quad (\text{B.2})$$

for $0 \leq s \leq t$. Substituting (B.2) back in (B.1) and integrating over s completes the proof.

B.2 Proof of Theorem 2.2

For $s \in [0, t]$, let the vector measure $\tilde{\boldsymbol{\mu}}$ be defined as $d\tilde{\boldsymbol{\mu}}(s) := \boldsymbol{\xi}(t-s)ds$ where $\boldsymbol{\xi}(s)$ is given by (2.5). Then $\int_0^t |\langle \mathbf{y}, \boldsymbol{\xi}(t-s) \rangle| ds$ is exactly in the form of a support function of a zonoid (see e.g., [74, Ch. 2]). Using the one-to-one correspondence between a compact convex set and its support function, the corresponding set is a zonoid.

From Property 1.6.3 and (2.10), $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ is the translation of a set with support

function $\int_0^t |\langle \mathbf{y}, \boldsymbol{\xi}(t-s) \rangle| ds$, i.e., the translation of a zonoid. Thus, we conclude that $\mathcal{X}_t^\square(\{\mathbf{x}_0\})$ is a zonoid.

B.3 Proof of Proposition 2.3

From Ch. 1.6.3, the supporting hyperplane at any $\mathbf{x}^{\text{bdy}} \in \partial\mathcal{X}_t(\{\mathbf{x}_0\})$ with outward normal $\mathbf{y} \in \mathbb{R}^d$ is $\langle \mathbf{y}, \mathbf{x}^{\text{bdy}} \rangle = h_{\partial\mathcal{X}_t(\{\mathbf{x}_0\})}(\mathbf{y})$, and the Legendre-Fenchel conjugate

$$h_{\partial\mathcal{X}_t(\{\mathbf{x}_0\})}^*(\mathbf{x}^{\text{bdy}}) = 0. \quad (\text{B.3})$$

For $j \in \llbracket m \rrbracket$, let \mathbf{y} comprise of subvectors $\mathbf{y}_j \in \mathbb{R}^{r_j}$. Since the Cartesian product (2.8) is equivalent to the Minkowski sum $\mathcal{X}_{1t} \dot{+} \mathcal{X}_{2t} \dot{+} \dots \dot{+} \mathcal{X}_{mt}$, and the support function of Minkowski sum is the sum of support functions of the summand sets [47, p. 48], we have

$$h_{\mathcal{X}(\{\mathbf{x}_0\}, t)}(\mathbf{y}) = \sum_{j=1}^m h_{\mathcal{X}_{jt}(\{\mathbf{x}_0\})}(\mathbf{y}_j) \Rightarrow h_{\mathcal{X}_t^\square(\{\mathbf{x}_0\})}^*(\mathbf{x}^{\text{bdy}}) = \sum_{j=1}^m h_{\mathcal{X}_{jt}(\{\mathbf{x}_0\})}^*(\mathbf{x}_j^{\text{bdy}}), \quad (\text{B.4})$$

wherein the last line follows from the property that the Legendre-Fenchel conjugate of a *separable* sum equals to the sum of the Legendre-Fenchel conjugates [93, p. 95].

Therefore, combining (B.3) and (B.4), we obtain

$$\sum_{j=1}^m \inf_{\mathbf{y}_j \in \mathbb{R}^{r_j}} \left\{ \langle -\mathbf{x}_j^{\text{bdy}} + \exp(t\mathbf{A}_j) \mathbf{x}_{j0} + \nu_j \boldsymbol{\zeta}_j(t), \mathbf{y}_j \rangle + \mu_j \int_0^t |\langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle| ds \right\} = 0. \quad (\text{B.5})$$

For $j \in \llbracket m \rrbracket$, since each objective in (B.5) involves an integral of the absolute value of a polynomial in s of degree $r_j - 1$, that polynomial can have *at most* $r_j - 1$ roots in the interval $[0, t]$, i.e., can have at most $r_j - 1$ sign changes in that interval. If all $r_j - 1$ roots of the aforesaid polynomial are in $[0, t]$, we denote these roots as $\sigma_1 \leq \sigma_2 \leq \dots \leq \sigma_{r_j-1}$,

and write

$$\begin{aligned}
 \int_0^t |\langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle| ds &= \pm \int_0^{\sigma_1} \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle ds \mp \int_{\sigma_1}^{\sigma_2} \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle ds \\
 &\quad \pm \cdots \pm (-1)^{r_j-1} \int_{\sigma_{r_j-1}}^t \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle ds \\
 &= \langle \mathbf{y}_j, \pm \boldsymbol{\zeta}_j(0, s_1) \mp \boldsymbol{\zeta}_j(\sigma_1, \sigma_2) \pm \cdots \pm (-1)^{r_j-1} \boldsymbol{\zeta}_j(\sigma_{r_j-1}, t) \rangle.
 \end{aligned} \tag{B.6}$$

Notice that even if the number of roots in $[0, t]$ is strictly less than¹ $r_j - 1$, the expression (B.6) is generic in the sense the corresponding summand integrals become zero.

Thus, combining (B.5) and (B.6), we arrive at

$$\begin{aligned}
 \sum_{j=1}^m \inf_{\mathbf{y}_j \in \mathbb{R}^{r_j}} \langle -\mathbf{x}_j^{\text{bdy}} + \exp(t\mathbf{A}_j) \mathbf{x}_{j0} + \nu_j \boldsymbol{\zeta}_j(t) \pm \mu_j \boldsymbol{\zeta}_j(0, \sigma_1) \\
 \mp \mu_j \boldsymbol{\zeta}_j(\sigma_1, \sigma_2) \pm \cdots \pm \mu_j (-1)^{r_j-1} \boldsymbol{\zeta}_j(\sigma_{r_j-1}, t), \mathbf{y}_j \rangle = 0.
 \end{aligned} \tag{B.7}$$

The left hand side of (B.7) being the sum of the infimum values of linear functions, can achieve zero if and only if each of those infimum equals to zero, i.e., if and only if

$$\begin{aligned}
 \mathbf{x}_j^{\text{bdy}} &= \exp(t\mathbf{A}_j) \mathbf{x}_{j0} + \nu_j \boldsymbol{\zeta}_j(t) \pm \mu_j \boldsymbol{\zeta}_j(0, s_1) \mp \mu_j \boldsymbol{\zeta}_j(\sigma_1, \sigma_2) \\
 &\quad \pm \cdots \pm (-1)^{r_j-1} \mu_j \boldsymbol{\zeta}_j(\sigma_{r_j-1}, t).
 \end{aligned} \tag{B.8}$$

Using (1.22), (2.5) and (2.6), we simplify (B.8) to (2.16), thereby completing the proof.

B.4 Proof of Corollary 2.4

From (2.16), we get two different parametric representations of $\mathbf{x}_j^{\text{bdy}}$ in terms of the parameter vector $\boldsymbol{\sigma}$. One parametric representation results from the choice of pos-

¹this may happen either because there are repeated roots in $[0, t]$, or because some real roots exist outside $[0, t]$, or because some roots are complex conjugates, or a combination of the previous three.

itive sign for the \pm appearing in (2.16), and another for the choice of negative sign for the same. Denoting the implicit representation corresponding to the parametric representation (2.16) with $+$ (resp. $-$) sign as $p_j^{\text{upper}}(\mathbf{x}) = 0$ (resp. $p_j^{\text{lower}}(\mathbf{x}) = 0$), the result follows.

B.5 Proof of Theorem 2.5

We notice that (2.16) gives polynomial parameterizations of the components of $\mathbf{x}_j^{\text{bdy}}$ for all $j \in \llbracket m \rrbracket$. In particular, for each $k \in \{1, \dots, r_j\}$, the right hand side of (2.16) is a homogeneous polynomial in $r_j - 1$ parameters $(\sigma_1, \sigma_2, \dots, \sigma_{r_j-1})$ of degree $r_j - k + 1$. By polynomial implicitization [63, p. 134], the corresponding implicit equations $p_j^{\text{upper}}(\mathbf{x}_j^{\text{bdy}}) = 0$ (when fixing plus sign for \pm in (2.16)) and $p_j^{\text{lower}}(\mathbf{x}_j^{\text{bdy}}) = 0$ (when fixing minus sign for \pm in (2.16)), must define affine varieties $V_{\mathbb{R}[x_1, \dots, x_{r_j}]}(p_j^{\text{upper}})$, $V_{\mathbb{R}[x_1, \dots, x_{r_j}]}(p_j^{\text{lower}})$ in $\mathbb{R}[x_1, \dots, x_d]$. Specifically, denote the right hand sides of (2.16) as $g_1^\pm, \dots, g_{r_j}^\pm$ for all $j \in \llbracket m \rrbracket$, where the superscripts indicate that either all g 's are chosen with plus signs, or all with minus signs. Then write (2.16) as

$$\mathbf{x}_{j,1}^{\text{bdy}}(\boldsymbol{\sigma}) = g_1^\pm(\boldsymbol{\sigma}), \quad \mathbf{x}_{j,2}^{\text{bdy}}(\boldsymbol{\sigma}) = g_2^\pm(\boldsymbol{\sigma}), \quad \vdots \quad \mathbf{x}_{j,r_j}^{\text{bdy}}(\boldsymbol{\sigma}) = g_{r_j}^\pm(\boldsymbol{\sigma}).$$

Now for each $j \in \llbracket m \rrbracket$, consider the ideal

$$I_j^\pm := \langle \langle \mathbf{x}_{j,1}^{\text{bdy}} - g_1^\pm, \mathbf{x}_{j,2}^{\text{bdy}} - g_2^\pm, \dots, \mathbf{x}_{j,r_j}^{\text{bdy}} - g_{r_j}^\pm \rangle \rangle \subseteq \mathbb{R}[\sigma_1, \sigma_2, \dots, \sigma_{r_j-1}, x_1, x_2, \dots, x_{r_j}],$$

and let $I_{j,r_j-1}^\pm := I_j^\pm \cap \mathbb{R}[x_1, \dots, x_{r_j}]$ be the $(r_j - 1)$ th elimination ideal of I_j^\pm . Then for each $j \in \llbracket m \rrbracket$, the variety

$$V(I_{j,r_j-1}^+) = V_{\mathbb{R}[x_1, \dots, x_{r_j}]}(p_j^{\text{upper}}).$$

Likewise, the variety $V(I_{j,r_j-1}^-) = V_{\mathbb{R}[x_1, \dots, x_{r_j}]}(p_j^{\text{lower}})$.

Thus, the algebraic boundary (i.e., the Zariski closure of the Euclidean boundary)

of \mathcal{X}_j is

$$\partial\mathcal{X}_j = V_{\mathbb{R}[x_1, \dots, x_{r_j}]}(p_j^{\text{upper}}) \cup V_{\mathbb{R}[x_1, \dots, x_{r_j}]}(p_j^{\text{lower}}).$$

Therefore, $\mathcal{X}_j := \{\mathbf{x} \in \mathbb{R}^{r_j} \mid p_j^{\text{upper}}(\mathbf{x}) \leq 0, p_j^{\text{lower}}(\mathbf{x}) \leq 0\}$ is semialgebraic for all $j \in \llbracket m \rrbracket$.

Since the Cartesian product of semialgebraic sets is semialgebraic, the statement follows from (2.8).

B.6 Proof of Theorem 2.6

We organize the proof in three steps.

Step 1: From (2.8), we have

$$\text{vol}(\mathcal{X}_t^\square(\{\mathbf{x}_0\})) = \text{vol}(\mathcal{X}_{1t} \times \mathcal{X}_{2t} \times \cdots \times \mathcal{X}_{mt}) = \prod_{j=1}^m \text{vol}(\mathcal{X}_{jt}(\{\mathbf{x}_0\})). \quad (\text{B.9})$$

Step 2: Motivated by (B.9), we focus on deriving the r_j -dimensional volume of $\mathcal{X}_{jt}(\{\mathbf{x}_0\})$.

For this purpose, we proceed as in [78] by uniformly discretizing the interval $[0, t]$ into n subintervals

$$\left[\frac{(i-1)t}{n}, \frac{it}{n} \right), \quad i = 1, 2, \dots, n,$$

with $(n+1)$ breakpoints $\{t_i\}_{i=0}^n$, where $t_i := it/n$ for $i = 0, 1, \dots, n$.

From (2.13), and the vector $\boldsymbol{\xi}_j$ defined in (2.5), we then have

$$\begin{aligned} \text{vol}(\mathcal{X}_{jt}(\{\mathbf{x}_0\})) &= \text{vol}\left(\lim_{n \rightarrow \infty} \sum_{i=0}^n \frac{t}{n} \exp(t_i \mathbf{A}_j) \mathbf{b}_j[-\mu_j, \mu_j]\right) \\ &= \lim_{n \rightarrow \infty} \left(\frac{t}{n}\right)^{r_j} \text{vol}\left(\sum_{i=0}^n \exp(t_i \mathbf{A}_j) \mathbf{b}_j[-\mu_j, \mu_j]\right) = t^{r_j} \lim_{n \rightarrow \infty} \frac{1}{n^{r_j}} \text{vol}\left(\sum_{i=0}^n \mu_j \boldsymbol{\xi}_j(t_i)[-1, 1]\right). \end{aligned} \quad (\text{B.10})$$

At this point, we recognize that the set

$$\sum_{i=0}^n \mu_j \boldsymbol{\xi}_j(t_i)[-1, 1] \quad (\text{B.11})$$

in (B.10) is a Minkowski sum of $n + 1$ intervals, each interval being rotated and scaled in \mathbb{R}^{r_j} via different linear transformations $\exp(t_i \mathbf{A}_j)$, $i = 0, 1, \dots, n$. In other words, the set (B.11) is a zonotope imbedded in \mathbb{R}^{r_j} .

Using the formula for the volume of zonotopes [53, eqn. (57)], [187, exercise 7.19], we can write (B.10) as

$$\text{vol}(\mathcal{X}_{jt}(\{\mathbf{x}_0\})) = (2\mu_j t)^{r_j} \lim_{n \rightarrow \infty} \frac{1}{n^{r_j}} \times \sum_{0 \leq i_1 < i_2 < \dots < i_{r_j} \leq n} \det(\boldsymbol{\xi}_j(t_{i_1}) | \boldsymbol{\xi}_j(t_{i_2}) | \dots | \boldsymbol{\xi}_j(t_{i_{r_j}})). \quad (\text{B.12})$$

To compute the summand determinants in (B.12), let

$$\Delta_j(i_1, i_2, \dots, i_{r_j}) := \det(\boldsymbol{\xi}_j(t_{i_1}) | \boldsymbol{\xi}_j(t_{i_2}) | \dots | \boldsymbol{\xi}_j(t_{i_{r_j}})),$$

where $0 \leq i_1 < i_2 < \dots < i_{r_j} \leq n$. In the matrix list notation, let us use the vertical bars $|\cdot|$ to denote the absolute value of determinant. From (2.5), $\Delta_j(i_1, i_2, \dots, i_{r_j})$ equals

$$\begin{vmatrix} \frac{(i_1 t/n)^{r_j-1}}{(r_j-1)!} & \frac{(i_2 t/n)^{r_j-1}}{(r_j-1)!} & \dots & \frac{(i_{r_j} t/n)^{r_j-1}}{(r_j-1)!} \\ \frac{(i_1 t/n)^{r_j-2}}{(r_j-2)!} & \frac{(i_2 t/n)^{r_j-2}}{(r_j-2)!} & \dots & \frac{(i_{r_j} t/n)^{r_j-2}}{(r_j-2)!} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 \end{vmatrix} = \frac{(t/n)^{r_j(r_j-1)/2}}{\prod_{k=1}^{r_j-1} k!} \begin{vmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ i_1^{r_j-2} & i_2^{r_j-2} & \dots & i_{r_j}^{r_j-2} \\ i_1^{r_j-1} & i_2^{r_j-1} & \dots & i_{r_j}^{r_j-1} \end{vmatrix}, \quad (\text{B.13})$$

wherein the last step used the properties of elementary row operations.

Notice that the determinant appearing in the last step of (B.13) is the well-known *Vandermonde determinant* that equals (see e.g., [124, p. 37])

$$\prod_{1 \leq a < b \leq r_j} (i_b - i_a). \quad (\text{B.14})$$

Combining (B.12), (B.13) and (B.14), we obtain

$$\text{vol}(\mathcal{X}_{jt}(\{\mathbf{x}_0\})) = \frac{(2\mu_j)^{r_j} t^{r_j(r_j+1)/2}}{\prod_{k=1}^{r_j-1} k!} \lim_{n \rightarrow \infty} \frac{1}{n^{r_j(r_j+1)/2}} \times \sum_{0 \leq i_1 < i_2 < \dots < i_{r_j} \leq n} \prod_{1 \leq a < b \leq r_j} (i_b - i_a). \quad (\text{B.15})$$

Step 3: Our next task is to simplify (B.15) by eliminating the limit and the nested sums.

Observe that the sum

$$\sum_{0 \leq i_1 < i_2 < \dots < i_{r_j} \leq n} \prod_{1 \leq a < b \leq r_j} (i_b - i_a), \quad (\text{B.16})$$

returns a polynomial in n of degree $r_j(r_j+1)/2$, and hence the limit in (B.15) is always well-defined. Specifically, the limit extracts the leading coefficient of this polynomial.

Let us denote the leading coefficient of the sum (B.16) as $c(r_j)$. By the Euler-Maclaurin formula [188], [189, Chap. II.10]:

$$c(r_j) = \int_{0 \leq y_1 < y_2 < \dots < y_{r_j} \leq 1} \prod_{1 \leq \alpha < \beta \leq r_j} (y_\alpha - y_\beta) \cdot \prod_{a=1}^{r_j} dy_a. \quad (\text{B.17})$$

One way to unpack (B.17) is to write it as a sum over the symmetric permutation group \mathfrak{S}_{r_j} of the finite set $\{1, 2, \dots, r_j\}$, i.e.,

$$c(r_j) = \sum_{\sigma \in \mathfrak{S}_{r_j}} \text{sgn}(\sigma) \frac{1}{\prod_{k=1}^{r_j} (\sigma_1 + \sigma_2 + \dots + \sigma_k)},$$

where

$$\text{sgn}(\sigma) := (-1)^\nu, \quad \nu := \{\#\{(i, j) \mid i < j, \sigma(i) > \sigma(j)\}\},$$

and # stands for “the number of”. We will now prove that

$$c(r_j) = \prod_{k=1}^{r_j-1} \frac{(k!)^2}{(2k+1)!}. \quad (\text{B.18})$$

To this end, we write $r_j! \cdot c(r_j)$ as an integral over the unit cube $[0, 1]^{r_j}$:

$$r_j! \cdot c(r_j) = \int_{[0,1]^{r_j}} \prod_{1 \leq a < b \leq r_j} |y_a - y_b| \, dy_1 \dots dy_{r_j}. \quad (\text{B.19})$$

In 1955, de Bruijn [190, see toward the end of Sec. 9] used certain Pfaffians to evaluate

$$\int_{[0,1]^{r_j}} \prod_{1 \leq a < b \leq r_j} |y_a - y_b| \, dy_1 \dots dy_{r_j} \frac{r_j! \cdot \{1! \times 2! \times \dots \times (r_j - 1)!\}^2}{1! \times 3! \times \dots \times (2r_j - 1)!}, \quad r_j = 2, 3, \dots,$$

which upon substitution in (B.19), indeed yields (B.18).

Combining (B.15) and (B.18), we arrive at

$$\text{vol}(\mathcal{X}_{jt}(\{\mathbf{x}_0\})) = \frac{(2\mu_j)^{r_j} t^{r_j(r_j+1)/2}}{\prod_{k=1}^{r_j-1} k!} c(r_j) = (2\mu_j)^{r_j} t^{r_j(r_j+1)/2} \prod_{k=1}^{r_j-1} \frac{k!}{(2k+1)!}. \quad (\text{B.20})$$

Finally, substituting (B.20) in (B.9), and recalling that $r_1 + r_2 + \dots + r_m = d$, the expression (2.31) follows.

B.7 Proof of Theorem 2.7

From (2.5), the subvector $\boldsymbol{\xi}_j(t-s)$, where $j = 1, 2, \dots, m$, is component-wise nonnegative for all $s \in [0, t]$.

Therefore, by triangle inequality, we have

$$\int_0^t |\langle \boldsymbol{\eta}, \boldsymbol{\xi}(t-s) \rangle| \, ds \leq \int_0^t \sum_{j=1}^m \langle |\boldsymbol{\eta}_j|, \mu_j \boldsymbol{\xi}_j(t-s) \rangle = \langle |\boldsymbol{\eta}|, \boldsymbol{\zeta}(t) \rangle, \quad (\text{B.21})$$

where $|\boldsymbol{\eta}_j|$ denotes the j th subvector with component-wise absolute values. Let us call $|\boldsymbol{\eta}|$ as the “absolute unit vector”.

The upper bound in (B.21) is convex in $\boldsymbol{\eta}$, and is maximized by an absolute unit

vector collinear with $\zeta(t)$, or equivalently by $\boldsymbol{\eta} \in \mathbb{S}^{d-1}$ such that

$$\boldsymbol{\eta} = \pm \frac{\zeta(t)}{\|\zeta(t)\|_2}, \quad (\text{B.22})$$

i.e., the unit vectors associated with $\zeta(t)$ up to plus-minus sign permutations among its components.

Out of the 2^d unit vectors given by (B.22), the “all plus” and “all minus” unit vectors achieve equality in (B.21), and hence must be the maximizers of (2.33). The inequality (B.21) remains strict for the remaining $2^d - 2$ unit vectors in (B.22), thus are suboptimal for (2.33).

Therefore, the maximizers in (2.34) are

$$\boldsymbol{\eta}^{\max} = \zeta(t) / \|\zeta(t)\|_2, \quad -\zeta(t) / \|\zeta(t)\|_2,$$

which upon substitution in (2.33), results in (2.35).

C | Proofs for Chapter 3

C.1 Proof of Theorem 3.1

Let us denote the summation of the first two constant terms in the right-hand-side of (3.9) as η_t . W.l.o.g., we prove our claim in a translated coordinate system with origin at η_t .

Notice that the bounding hypersurfaces in (3.9) have an antipodal property, i.e., for any fixed $\sigma \in \mathcal{W}_t$, the line segment $\ell(\sigma) := \lambda \mathbf{x}^{\text{upper}}(\sigma) + (1 - \lambda) \mathbf{x}^{\text{lower}}(\sigma)$ generated by varying $\lambda \in [0, 1]$, will pass through the origin. Since \mathcal{X}_t is convex and compact, for any $\tilde{\mathbf{x}} \in \mathcal{X}_t$, the line through $\tilde{\mathbf{x}}$ and the origin crosses the boundary at the antipodal points $\tilde{\mathbf{x}}^{\text{upper}} = \mathbf{x}^{\text{upper}}(\sigma)$ and $\tilde{\mathbf{x}}^{\text{lower}} = \mathbf{x}^{\text{lower}}(\sigma)$ for some $\sigma \in \mathcal{W}_t$. Therefore, $\tilde{\mathbf{x}} \in \ell(\sigma)$. It follows that π is surjective.

C.2 Proof of Theorem 3.2

Using the parameterization (3.12), we get

$$d \text{vol}_n (\mathcal{X}_t (\{\mathbf{0}\})) = \left| \det (D\pi) \right| d\sigma_1 \cdots d\sigma_{n-1} d\lambda. \quad (\text{C.1})$$

From (3.12), the map π is $C^1 (\mathcal{W}_t \times [0, 1])$. By Sard's theorem [191, Sec. 2], the *set of critical values* (image of the set of *critical points* in $\mathcal{W}_t \times [0, 1]$ where $\det (D\pi) = 0$) has n dimensional Lebesgue measure zero. Using (3.9) and Remark 3.1, direct computation gives

$$\left| \det (D\pi) \right| = \left| \det \left(\frac{\partial \pi(\sigma, \lambda)}{\partial \sigma} \quad \frac{\partial \pi(\sigma, \lambda)}{\partial \lambda} \right) \right| = (3.14).$$

This completes the proof.

D | Proofs for Chapter 4

D.1 Proof of Lemma 4.1

For convenience, let $\theta := s - \tau$. From (4.5), the vector $e^{\theta \mathbf{A}_{\text{con}}} \mathbf{b}$ equals to the last column of the matrix exponential of $\theta \mathbf{A}_{\text{con}}$.

On the other hand, \mathbf{A}_{con} being a companion matrix, each consecutive row of $e^{\theta \mathbf{A}_{\text{con}}}$ must be the derivative of its previous row w.r.t. θ . Therefore, it suffices to determine the top right corner (i.e., first row and last column) entry of $e^{\theta \mathbf{A}_{\text{con}}}$, which we denote as $g(\theta)$.

We have

$$\begin{aligned}
 g(\theta) &= \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \end{pmatrix} e^{\theta \mathbf{A}_{\text{con}}} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \\
 &= \sum_{r=0}^{\infty} \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \end{pmatrix} \frac{\theta^r \mathbf{A}_{\text{con}}^r}{r!} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \\
 &= \sum_{r=0}^{\infty} \frac{\theta^r}{r!} \sum_{i=1}^n \frac{\lambda_i^r}{p'(\lambda_i)}, \tag{D.1}
 \end{aligned}$$

where the second line uses the Taylor series for matrix exponential, $p'(\cdot)$ denotes the derivative of the characteristic polynomial $p(\cdot)$ in (4.6), and the third line follows from a known result due to Dan Kalman [192, equation (9)] that gives a formula for

the top right corner entry of the r th power of a companion matrix in terms of the eigenvalues. The latter result in turn comes from a connection between the powers of the companion matrix and the generalized Fibonacci sequences; see also [193, 194].

Exchanging the order of summation in (D.1), we get

$$g(\theta) = \sum_{i=1}^n \frac{1}{p'(\lambda_i)} \sum_{r=0}^{\infty} \frac{(\lambda_i \theta)^r}{r!} = \sum_{i=1}^n \frac{e^{\lambda_i \theta}}{p'(\lambda_i)}. \quad (\text{D.2})$$

It follows that

$$e^{\theta \mathbf{A}_{\text{con}}} \mathbf{b} = \begin{pmatrix} g(\theta) \\ g^{(1)}(\theta) \\ \vdots \\ g^{(n-1)}(\theta) \end{pmatrix} \quad (\text{D.3})$$

where the parenthetical superscript denotes the order of derivative w.r.t. θ .

Combining (4.11), (D.2) and (D.3), we obtain

$$\begin{aligned} \langle \mathbf{c}, e^{\theta \mathbf{A}_{\text{con}}} \mathbf{b} \rangle &= \sum_{i=1}^n \frac{(c_0 + c_1 \lambda_i + \dots + c_{n-1} \lambda_i^{n-1}) e^{\lambda_i \theta}}{p'(\lambda_i)} \\ &= \sum_{i=1}^n \frac{(p(\lambda_i) - \lambda_i^n) e^{\lambda_i \theta}}{p'(\lambda_i)}. \end{aligned} \quad (\text{D.4})$$

By definition, $p(\lambda_i) = 0$. Taking the logarithmic derivative of the characteristic polynomial $p(\lambda) = \prod_{i=1}^n (\lambda - \lambda_i)$, we have $p'(\lambda_i) = \prod_{\substack{j=1 \\ j \neq i}}^n (\lambda_i - \lambda_j)$. Substituting these back in (D.4), and recalling that $\theta = s - \tau$, the proof is complete.

D.2 Proof of Theorem 4.1

For a fixed finite $s > 0$, the function $f(\tau)$ in (4.11) is continuous and bounded in $\tau \in [0, s]$. So $v \mapsto I(v)$ is a continuous functional. Therefore, the infimum and supremum in (4.10) are achieved. Because the functional $I(v)$ in (4.11) is linear, we can determine

the respective optimal values in terms of the disjoint union

$$[0, s] = \mathcal{L}_f^- \cup \mathcal{L}_f^{++}.$$

Using (4.10), (4.11) and Lemma 4.1, we arrive at (4.14).

The recovery of the LTI reach set (4.2) from the integrator reach set (3.1) follows from the input correspondence (4.8) and the subsequent discussion in Ch. 4.1.2.

D.3 Proof of Theorem 4.2

Support function is distributive over sum, so from (2.2) and (1.23), we obtain

$$h_{\mathcal{X}_t}(\mathbf{y}) = \langle \mathbf{y}, \exp(t\mathbf{A})\mathbf{x}_0 \rangle + h_{\int_0^t \exp((t-s)\mathbf{A})\mathbf{B}\mathcal{U}(s) ds}(\mathbf{y}). \quad (\text{D.5})$$

Using [78, Proposition 1], we rewrite the RHS of (D.5)

$$\int_0^t h_{\exp((t-s)\mathbf{A})\mathbf{B}\mathcal{U}(s)}(\mathbf{y}) ds = \int_0^t \sup_{\mathbf{u}(s) \in \mathcal{U}(s)} \langle \mathbf{y}, \exp((t-s)\mathbf{A})\mathbf{B}\mathbf{u}(s) \rangle ds. \quad (\text{D.6})$$

Let $\mathbf{z}(s) := (\exp((t-s)\mathbf{A})\mathbf{B})^\top \mathbf{y}$, and $\tilde{\mathbf{u}} = \mathbf{M}\mathbf{u} + \mathbf{p}$, for all $0 \leq s \leq t$, then, we rewrite the integrand of the RHS of (D.6)

$$\begin{aligned} & \sup \langle \mathbf{z}, \mathbf{M}^{-1}(\tilde{\mathbf{u}} - \mathbf{p}) \rangle \\ & \text{subject to } \|\tilde{\mathbf{u}}\|_p \leq 1. \end{aligned} \quad (\text{D.7})$$

Recall that $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$ for $\frac{1}{p} + \frac{1}{q} = 1$. From the definition of dual norm, we have

$$\begin{aligned} \langle \mathbf{z}, \mathbf{M}^{-1}(\tilde{\mathbf{u}} - \mathbf{p}) \rangle &= \langle \mathbf{M}^{-\top} \mathbf{z}, \tilde{\mathbf{u}} \rangle - \langle \mathbf{M}^{-\top} \mathbf{z}, \mathbf{p} \rangle \leq \|\mathbf{M}^{-\top} \mathbf{z}\|_q \|\tilde{\mathbf{u}}\|_p - \langle \mathbf{M}^{-\top} \mathbf{z}, \mathbf{p} \rangle \\ &\stackrel{(\text{D.7})}{\leq} \|\mathbf{M}^{-\top} \mathbf{z}\|_q - \langle \mathbf{M}^{-\top} \mathbf{z}, \mathbf{p} \rangle \end{aligned}$$

Substituting this supremum in (D.6) and then using (D.5), we arrive at (4.32).

D.4 Proof of Theorem 4.3

(i) Since $\mathbf{0}_{(n+K+1) \times 1}$ is in the feasible set defined by (4.39b)-(4.39d), and the objective in (4.39a) is linear, we have $\tilde{p}^* \leq 0$.

(ii) Since \tilde{p}^* is the optimal value of the convex relaxation of a nonconvex problem with optimal value p^* , we must have $\tilde{p}^* \leq p^*$. Thus, $\tilde{p}^* = 0 \Rightarrow 0 \leq p^*$. That $0 \leq p^*$ is equivalent to $\mathcal{X}_t^A \cap \mathcal{X}_t^B \neq \emptyset$, was explained before in Ch. 4.2.1.

(iii) We now show that when $\tilde{p}^* < 0$, the convexification is in fact lossless, i.e., $\tilde{p}^* = p^*$. Denote the arg min for (4.39) as $\boldsymbol{\eta}^{\min}$. It suffices to prove that $\boldsymbol{\eta}^{\min}$ achieves equality in (4.39d). To this end, suppose if possible, that $\|\mathbf{N}\boldsymbol{\eta}^{\min}\|_2 =: \varepsilon < 1$, i.e., $0 < \varepsilon < 1$. Now let $\tilde{\boldsymbol{\eta}} := \boldsymbol{\eta}^{\min}/\varepsilon$, which is clearly feasible w.r.t. (4.39b)-(4.39d). However,

$$\langle \boldsymbol{\kappa}(t), \tilde{\boldsymbol{\eta}} \rangle = \underbrace{\frac{1}{\varepsilon}}_{>1} \underbrace{\langle \boldsymbol{\kappa}(t), \boldsymbol{\eta}^{\min} \rangle}_{<0} < \langle \boldsymbol{\kappa}(t), \boldsymbol{\eta}^{\min} \rangle,$$

contradicting the supposition that the arg min achieves strict inequality in (4.39d). Therefore, if $\tilde{p}^* < 0$ then $\tilde{p}^* = p^*$. That $p^* < 0$ is equivalent to $\mathcal{X}_t^A \cap \mathcal{X}_t^B = \emptyset$, was explained before in Ch. 4.2.1.

D.5 Proof of Theorem 4.5

From (4.54) we have $\alpha_j(s) \leq \hat{\alpha}_j(s) < \hat{\beta}_j(s) \leq \beta_j(s)$, for $j = 1, \dots, m$ and $s \in [0, t]$ using the support function equation in (2.10) we arrive at

$$\begin{cases} \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle < 0, & h(t, \mathbf{y}_j) = \int_0^t \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle \alpha_j(s) ds + \langle \exp(At) \mathbf{x}_{j0}, \mathbf{y}_j \rangle \\ \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle > 0, & h(t, \mathbf{y}_j) = \int_0^t \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle \beta_j(s) ds + \langle \exp(At) \mathbf{x}_{j0}, \mathbf{y}_j \rangle \\ \langle \mathbf{y}_j, \boldsymbol{\xi}_j(t-s) \rangle = 0, & h(t, \mathbf{y}_j) = \langle \exp(At) \mathbf{x}_{j0}, \mathbf{y}_j \rangle \end{cases} \quad (\text{D.8})$$

from which, we conclude

$$\begin{aligned}
 \hat{h}_j(t, \mathbf{y}_j) \leq h_j(t, \mathbf{y}_j), \quad \forall j = 1, \dots, m &\iff \sum_{j=1}^m \hat{h}_j(t, \mathbf{y}_j) \leq \sum_{j=1}^m h_j(t, \mathbf{y}_j) \\
 &\iff \hat{\mathcal{X}}_{t1}^\square(\mathbf{x}_0, \hat{\mathcal{U}}) + \dots + \hat{\mathcal{X}}_{tm}^\square(\mathbf{x}_0, \hat{\mathcal{U}}) \subseteq \mathcal{X}_{t1}(\mathbf{x}_0, \mathcal{U}) + \dots + \mathcal{X}_{tm}^\square(\mathbf{x}_0, \mathcal{U}) \\
 &\iff \hat{\mathcal{X}}_t^\square(\mathbf{x}_0, \hat{\mathcal{U}}) \subseteq \mathcal{X}_t^\square(\mathbf{x}_0, \mathcal{U}) \iff \hat{\mathcal{R}}_t(\mathbf{z}_0, \mathbf{w}_0, \mathcal{V}) \subseteq \mathcal{R}_t(\mathbf{z}_0, \mathbf{w}_0, \mathcal{V}), \\
 &\iff \hat{\mathcal{Z}}_t(\mathbf{z}_0, \mathcal{V}) \subseteq \mathcal{Z}_t(\mathbf{z}_0, \mathcal{V}),
 \end{aligned}$$

the last but one inequality follows from τ being a diffeomorphism mapping and this completes the proof.

D.6 Proof of Theorem 4.6

Consider a boundary point $\mathbf{x}(\boldsymbol{\sigma})^{\text{bdy}} = (x_1^{\text{bdy}}(\boldsymbol{\sigma}), x_2^{\text{bdy}}(\boldsymbol{\sigma}), \dots, x_d^{\text{bdy}}(\boldsymbol{\sigma})) \in \partial\mathcal{X}(\boldsymbol{\sigma})$ with parametric representation (3.9). Using this parametric representation, we rewrite (4.57) as a determinant equation (see [195, Sec. 14], and [196, 197]),

$$\begin{aligned}
 \mathcal{S}(\boldsymbol{\sigma}) = \langle \hat{\mathbf{n}}(\boldsymbol{\sigma}), \hat{\boldsymbol{\ell}} \rangle &= \begin{vmatrix} \frac{\partial x_1}{\partial \boldsymbol{\sigma}} & \frac{\partial x_2}{\partial \boldsymbol{\sigma}} & \dots & \frac{\partial x_d}{\partial \boldsymbol{\sigma}} \\ \ell_1 & \ell_2 & \dots & \ell_d \end{vmatrix} \\
 &= (-1)^{d-1} 2^{d-1} \mu(\sigma_1) \mu(\sigma_2) \dots \mu(\sigma_{d-1}) \begin{vmatrix} \frac{\sigma_1^{d-1}}{(d-1)!} & \frac{\sigma_1^{d-2}}{(d-2)!} & \dots & 1 \\ \frac{\sigma_2^{d-1}}{(d-2)!} & \frac{\sigma_2^{d-2}}{(d-2)!} & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\sigma_{d-1}^{d-1}}{(d-2)!} & -\frac{\sigma_{d-1}^{d-2}}{(d-2)!} & \dots & 1 \\ \ell_1 & \ell_2 & \dots & \ell_d \end{vmatrix} \quad (\text{D.9a})
 \end{aligned}$$

$$\begin{aligned}
 &= (-1)^{d-1} 2^{d-1} \mu(\sigma_1) \mu(\sigma_2) \cdots \mu(\sigma_{d-1}) \times \\
 &\quad \sum_{i=1}^d \ell_i (-1)^{1+i} \sum_{i_1, i_2, \dots, i_{d-1}=1}^d \varepsilon_{i_1, i_2, \dots, i_{d-1}}^{1, 2, i-1, i+1, \dots, n} \prod_{r=1}^{d-1} \frac{\sigma_r^{d-i_r}}{(d-i_r)!} = 0. \quad (\text{D.9b})
 \end{aligned}$$

In above, we use vertical bars to denote the determinant of matrix represented in array format. Canceling the prefactor in (D.9b) followed by algebraic simplification gives

$$\left(\prod_{j \neq i} (\sigma_i - \sigma_j) \right) \sum_{i=1}^d (d-i)! (-1)^{d-i} \ell_i e_{i-1} = 0. \quad (\text{D.10})$$

Following (4.57), we drop the factor $\prod_{j \neq i} (\sigma_i - \sigma_j)$ from (D.10), and arrive at (4.59)..

D.7 Proof of Theorem 4.7

Let $P_{\mathcal{X}}(\mathbf{y})$ be a supporting hyperplane of set \mathcal{X} , specified by the direction vector $\mathbf{y} \in \mathbb{S}^{d-1}$ and point $p_{\mathcal{X}}(\mathbf{y}) \in \mathcal{X}$ such that

$$p_{\mathcal{X}}(\mathbf{y}) = \operatorname{argmax}_{\mathbf{x}^{\text{bdy}} \in \partial \mathcal{X}} \langle \mathbf{x}^{\text{bdy}}, \mathbf{y} \rangle. \quad (\text{D.11})$$

Similarly, we define the supporting hyperplane, $P_{\ell}(\mathbf{y})$ on ℓ in the direction \mathbf{y} , specified by $p_{\ell}(\mathbf{y}) \in \partial \ell$ given by

$$p_{\ell}(\mathbf{y}) = \operatorname{argmax}_{\mathbf{x}^{\ell} \in \ell} \langle \mathbf{x}^{\ell}, \mathbf{y} \rangle. \quad (\text{D.12})$$

Since \mathcal{X} and ℓ are convex, the supporting hyperplane of $\partial(\ell + \mathcal{X}_t)$ in the direction \mathbf{y} , touches this set for all $\mathbf{y} \in \mathbb{S}^{d-1}$ at

$$p_{(\ell + \mathcal{X}_t)}(\mathbf{y}) = p_{\mathcal{X}}(\mathbf{y}) + p_{\ell}(\mathbf{y}) = \begin{cases} \operatorname{argmax}_{\mathbf{x}^{\text{bdy}} \in \partial \mathcal{X}} \langle \mathbf{x}^{\text{bdy}}, \widehat{\mathbf{y}} \rangle + \ell_1 & \text{if } \langle \widehat{\ell}, \mathbf{y} \rangle > 0, \\ \operatorname{argmax}_{\mathbf{x}^{\text{bdy}} \in \partial \mathcal{X}} \langle \mathbf{x}^{\text{bdy}}, \mathbf{y} \rangle + \ell_0 & \text{if } \langle \widehat{\ell}, \mathbf{y} \rangle < 0, \\ \text{any point } \in \{ \mathcal{X}_{\text{cut}} + \ell \} & \text{if } \langle \widehat{\ell}, \mathbf{y} \rangle = 0, \end{cases} \quad (\text{D.13})$$

where

$$\operatorname{argmax}_{\mathbf{x}^{\text{bdy}} \in \partial \mathcal{X}} \langle \mathbf{x}^{\text{bdy}}, \mathbf{y} \rangle \in \begin{cases} \mathcal{X}_{\geq} & \text{if } \langle \widehat{\ell}, \mathbf{y} \rangle > 0 \\ \mathcal{X}_{\leq} & \text{if } \langle \widehat{\ell}, \mathbf{y} \rangle < 0 \end{cases} \quad (\text{D.14})$$

and the last equality in (D.13) is achieved using criteria (4.64). From (D.13) we can recover (4.65) and complete the proof.

E | Proofs for Chapter 5

E.1 Proof of Proposition 5.2

Recall that δ relates to h_1, h_2 via (5.3). Since $1 \leq k_1 < k_2 \leq d$, we know that $\frac{1}{k_2} \|\mathbf{y}\|_1 < \frac{1}{k_1} \|\mathbf{y}\|_1$. Depending on the value of $\|\mathbf{y}\|_\infty$, we need to consider three subsets of unit vectors.

Specifically, for the unit vectors \mathbf{y} satisfying $\frac{1}{k_2} \|\mathbf{y}\|_1 < \frac{1}{k_1} \|\mathbf{y}\|_1 \leq \|\mathbf{y}\|_\infty$, we have $h_1(\mathbf{y}) = h_2(\mathbf{y}) = \|\mathbf{y}\|_\infty$ and $h_1(\mathbf{y}) - h_2(\mathbf{y}) = 0$.

On the other hand, for the unit vectors \mathbf{y} satisfying $\|\mathbf{y}\|_\infty \leq \frac{1}{k_2} \|\mathbf{y}\|_1 < \frac{1}{k_1} \|\mathbf{y}\|_1$, we have $h_1(\mathbf{y}) = \frac{1}{k_1} \|\mathbf{y}\|_1$, $h_2(\mathbf{y}) = \frac{1}{k_2} \|\mathbf{y}\|_1$, hence we obtain that $h_1(\mathbf{y}) - h_2(\mathbf{y}) = (1/k_1 - 1/k_2) \|\mathbf{y}\|_1$, which is always nonnegative.

Finally, for the unit vectors \mathbf{y} satisfying $\frac{1}{k_2} \|\mathbf{y}\|_1 \leq \|\mathbf{y}\|_\infty < \frac{1}{k_1} \|\mathbf{y}\|_1$, we must have $h_1(\mathbf{y}) - h_2(\mathbf{y}) = \frac{1}{k_1} \|\mathbf{y}\|_1 - \|\mathbf{y}\|_\infty < (1/k_1 - 1/k_2) \|\mathbf{y}\|_1$.

Therefore, using (5.3) we get

$$\delta(\mathcal{K}_1, \mathcal{K}_2) = \left(\frac{1}{k_1} - \frac{1}{k_2} \right) \sup_{\|\mathbf{y}\|_2=1} \|\mathbf{y}\|_1.$$

Using the same arguments as in (5.6), we obtain $\sup_{\|\mathbf{y}\|_2=1} \|\mathbf{y}\|_1 = \sqrt{d}$, which is achieved by 2^d vectors of the form \mathbf{v}/\sqrt{d} for all $\mathbf{v} \in \{-1, 1\}^d$. This completes the proof.

E.2 Proof of Proposition 5.3

Proceeding as in Ch. 5.1, for $\mathbf{y} \in \mathbb{S}^{d-1}$ we get

$$\begin{aligned}
 \|\mathbf{T}\mathbf{y}\|_{q_2} &\leq m^{\frac{1}{q_2} - \frac{1}{q_1}} \|\mathbf{T}\mathbf{y}\|_{q_1} \\
 \Rightarrow \|\mathbf{T}\mathbf{y}\|_{q_2} - \|\mathbf{T}\mathbf{y}\|_{q_1} &\leq \left(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1\right) \|\mathbf{T}\mathbf{y}\|_{q_1} \\
 &\leq \left(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1\right) \sup_{\|\mathbf{y}\|_2=1} \|\mathbf{T}\mathbf{y}\|_{q_1} = \left(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1\right) \|\mathbf{T}\|_{2 \rightarrow q_1} \\
 \Rightarrow \sup_{\|\mathbf{y}\|_2=1} (\|\mathbf{T}\mathbf{y}\|_{q_2} - \|\mathbf{T}\mathbf{y}\|_{q_1}) &\leq \left(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1\right) \|\mathbf{T}\|_{2 \rightarrow q_1}.
 \end{aligned}$$

E.3 Proof of Proposition 5.5

(i) Following [198, Thm. 1], we bound the expected operator norm as $\mathbb{E}\|\mathbf{T}\|_{2 \rightarrow q_1} \leq C_{q_1} \max\{m^{1/q_1}, \sqrt{d}\}$. Combining this with (5.14), the result follows.

(ii) The expected $2 \rightarrow q_1$ operator norm bound, in this case, follows from specializing more general bound¹ given in [199, Thm. 1.1]. Specifically, we get

$$\mathbb{E}\|\mathbf{T}\|_{2 \rightarrow q_1} \leq C 2^{5/q_1} (\log m)^{1/q_1} \left(\gamma_2 + \gamma_{q_1} \mathbb{E} \max_{i,j} |a_{ij}| \right) + 2^{1/q_1} \gamma_{q_1}, \quad (\text{E.1})$$

where $C, \gamma_2, \gamma_{q_1}$ are as in the statement. Combining (E.1) with (5.14), we obtain (5.18).

¹The operator norm bound in [199, Thm. 1.1] is more general on two counts. First, the operator norm considered there is $p^* \rightarrow q$ where $1 \leq p^* \leq 2 \leq q \leq \infty$. Second, the result therein allows nonuniform deterministic scaling of the standard Gaussian entries of \mathbf{T} .

E.4 Proof of Theorem 5.2

We have

$$\mathcal{X}_t^i = \Phi(t, 0)\mathcal{X}_0 \dot{+} \int_0^t \Phi(t, \tau)\mathbf{B}(\tau)\mathcal{U}^i \, d\tau, \quad i \in \{1, 2\}, \quad (\text{E.2})$$

where $\dot{+}$ denotes the Minkowski sum and the second summand in (E.2) is a set-valued Aumann integral.

Since the support function is distributive over the Minkowski sum, following [78, Prop. 1] and (1.23), from (E.2) we find that

$$h_i(\mathbf{y}) := h_{\mathcal{X}_t^i}(\mathbf{y}) = \left(\sup_{\mathbf{x}_0 \in \mathcal{X}_0} \langle \mathbf{y}, \Phi(t, 0)\mathbf{x}_0 \rangle \right) + \int_0^t \sup_{\mathbf{u}^i(\tau) \in \mathcal{U}^i} \langle \mathbf{y}, \Phi(t, \tau)\mathbf{B}(\tau)\mathbf{u}^i(\tau) \rangle \, d\tau, \quad (\text{E.3})$$

wherein $i \in \{1, 2\}$ and the sets \mathcal{U}^i are given by (5.24). Next, we follow the same arguments as in [72, Thm. 1] to simplify (E.3) as

$$h_i(\mathbf{y}) = \left(\sup_{\mathbf{x}_0 \in \mathcal{X}_0} \langle \mathbf{y}, \Phi(t, 0)\mathbf{x}_0 \rangle \right) + \int_0^t \| (\Phi(t, \tau)\mathbf{B}(\tau))^\top \mathbf{y} \|_{q_i} \, d\tau, \quad (\text{E.4})$$

where q_i is the Hölder conjugate of p_i . Then (5.3) together with (E.4) yield (5.26).

E.5 Proof of Corollary 5.3

From (5.21), we obtain the estimate

$$\delta(\mathcal{X}_t^1, \mathcal{X}_t^2) \leq \left(m^{\frac{1}{q_2} - \frac{1}{q_1}} - 1 \right) \int_0^t \| (\Phi(t, \tau)\mathbf{B}(\tau))^\top \|_{2 \rightarrow q_1} \, d\tau. \quad (\text{E.5})$$

Recall that the norm of a linear operator is related to the norm of its adjoint via

$$\| (\cdot)^\top \|_{\alpha \rightarrow \beta} = \| \cdot \|_{\beta^* \rightarrow \alpha^*},$$

where α^*, β^* are the Hölder conjugates of α, β , respectively. Using this fact in (E.5) completes the proof.

F | Proofs for Chapter 6

F.1 Proof of Theorem 6.1

Notice that for $i = 1, \dots, n_y$, we have

$$d_{n_x,i} := \widehat{h}_{\mathcal{X}}(\mathbf{y}_i) - h_{\overline{\text{conv}}(\{\mathbf{x}_j\}_{j=1}^{n_x})}(\mathbf{y}_i) = \mathbf{y}_i^\top \boldsymbol{\nu}. \quad (\text{F.1})$$

Therefore, $d_{n_x,i}$ is i.i.d. with $\mathbb{E}_{\mathbf{x}, \boldsymbol{\nu}}[d_{n_x,i}] = 0$ and $\mathbb{E}_{\mathbf{x}, \boldsymbol{\nu}}[d_{n_x,i}^2] = 1$. As $n_x \rightarrow \infty$, we have $h_{\overline{\text{conv}}(\{\mathbf{x}_j\}_{j=1}^{n_x})}(\cdot) \rightarrow h_{\mathcal{X}}(\cdot)$ and the least squares problem (6.3) becomes identical to the one investigated in [179], where the proof of consistency for $n_y \rightarrow \infty$ is provided.

F.2 Proof of Theorem 6.2

The computation for each layer of ISNN are represented as follows

$$\begin{aligned} \mathbf{z}_1 &= \boldsymbol{\sigma}(\mathbf{W}_1^{(y)} \mathbf{y}), \\ \mathbf{z}_{k+1} &= \boldsymbol{\sigma}(\mathbf{W}_{k+1}^{(z)} \mathbf{z}_k + \mathbf{W}_{k+1}^{(y)} \mathbf{y}), \quad k \in \llbracket \ell - 2 \rrbracket, \\ \mathbf{z}_\ell &= \mathbf{W}_\ell^{(z)} \mathbf{z}_{\ell-1} + \mathbf{W}_\ell^{(y)} \mathbf{y}, \end{aligned} \quad (\text{F.2})$$

with elementwise non-negative matrices $\mathbf{W}_{1:\ell}^{(z)}$. Each layer conducts a composition of non-negative sums of linear functions, succeeded by a convex, non-decreasing mapping. The sublinearity with respect to the input vector $\mathbf{y} \in \mathbb{R}^d$ is maintained due to the positive homogeneity of the function $\boldsymbol{\sigma}(\cdot)$.

Bibliography

- [1] T. Pecsvaradi and K. S. Narendra, “Reachable sets for linear dynamical systems,” *Information and control*, vol. 19, no. 4, pp. 319–344, 1971.
- [2] A. Chutinan and B. H. Krogh, “Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations,” in *Hybrid Systems: Computation and Control: Second International Workshop, HSCC’99 Berg en Dal, The Netherlands, March 29–31, 1999 Proceedings 2*, pp. 76–90, Springer, 1999.
- [3] P. Varaiya, “Reach set computation using optimal control,” in *Verification of Digital and Hybrid Systems*, pp. 323–331, Springer, 2000.
- [4] 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.
- [5] C. Le Guernic and A. Girard, “Reachability analysis of linear systems using support functions,” *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 250–262, 2010.
- [6] M. Althoff, G. Frehse, and A. Girard, “Set propagation techniques for reachability analysis,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 369–395, 2021.
- [7] U. Topcu, A. K. Packard, P. Seiler, and G. J. Balas, “Robust region-of-attraction estimation,” *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 137–142, 2009.
- [8] A. Chakraborty, P. Seiler, and G. J. Balas, “Nonlinear region of attraction analysis for flight control verification and validation,” *Control Engineering Practice*, vol. 19, no. 4, pp. 335–345, 2011.
- [9] D. Henrion and M. Korda, “Convex computation of the region of attraction of polynomial control systems,” *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 297–312, 2013.

- [10] M. Nagumo, “Über die lage der integralkurven gewöhnlicher differentialgleichungen,” *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 24, pp. 551–559, 1942.
- [11] J. N. Maidens, S. Kaynama, I. M. Mitchell, M. M. Oishi, and G. A. Dumont, “Lagrangian methods for approximating the viability kernel in high-dimensional systems,” *Automatica*, vol. 49, no. 7, pp. 2017–2029, 2013.
- [12] M. Korda, D. Henrion, and C. N. Jones, “Convex computation of the maximum controlled invariant set for polynomial control systems,” *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 2944–2969, 2014.
- [13] W. Esterhuizen, T. Aschenbruck, and S. Streif, “On maximal robust positively invariant sets in constrained nonlinear systems,” *Automatica*, vol. 119, p. 109044, 2020.
- [14] T. Anevlavis, Z. Liu, N. Ozay, and P. Tabuada, “An enhanced hierarchy for (robust) controlled invariance,” in *2021 American Control Conference (ACC)*, pp. 4860–4865, IEEE, 2021.
- [15] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, “Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 4661–4666, IEEE, 2016.
- [16] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado, “Learning region of attraction for nonlinear systems,” in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 6477–6484, IEEE, 2021.
- [17] S. Bansal and C. J. Tomlin, “Deepreach: A deep learning approach to high-dimensional reachability,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1817–1824, IEEE, 2021.
- [18] A. Devonport, F. Yang, L. El Ghaoui, and M. Arcak, “Data-driven reachability analysis with christoffel functions,” in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 5067–5072, IEEE, 2021.
- [19] A. J. Thorpe, K. R. Ortiz, and M. M. Oishi, “Learning approximate forward reachable sets using separating kernels,” in *Learning for Dynamics and Control*, pp. 201–212, PMLR, 2021.
- [20] S. Haddad and A. Halder, “Convex and nonconvex sublinear regression with application to data-driven learning of reach sets,” in *2023 American Control Conference (ACC)*, pp. 4581–4586, IEEE, 2023.

- [21] J. L. Prince and A. S. Willsky, "Reconstructing convex sets from support line measurements," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 4, pp. 377–389, 1990.
- [22] J. L. Prince and A. S. Willsky, "Convex set reconstruction using prior shape information," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 5, pp. 413–427, 1991.
- [23] N. I. Fisher, P. Hall, B. A. Turlach, and G. S. Watson, "On the estimation of a convex set from noisy data on its support function," *Journal of the American Statistical Association*, vol. 92, no. 437, pp. 84–91, 1997.
- [24] T. T. Cai, A. Guntuboyina, and Y. Wei, "Adaptive estimation of planar convex sets," *The Annals of Statistics*, vol. 46, no. 3, pp. 1018–1049, 2018.
- [25] A. Devonport and M. Arcak, "Estimating reachable sets with scenario optimization," *Proceedings of Machine Learning Research*, vol. 120, pp. 75–84, 2020.
- [26] A. Chakrabarty, C. Danielson, S. Di Cairano, and A. Raghunathan, "Active learning for estimating reachable sets for systems with unknown dynamics," *IEEE Transactions on Cybernetics*, 2020.
- [27] Y. S. Soh and V. Chandrasekaran, "Fitting tractable convex sets to support function evaluations," *Discrete & Computational Geometry*, vol. 66, no. 2, pp. 510–551, 2021.
- [28] E. O'Reilly and V. Chandrasekaran, "Spectrahedral regression," *arXiv preprint arXiv:2110.14779*, 2021.
- [29] S. Patil and R. Alterovitz, "Interactive motion planning for steerable needles in 3d environments with obstacles," in *2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 893–899, IEEE, 2010.
- [30] S. Patil, J. Burgner, R. J. Webster, and R. Alterovitz, "Needle steering in 3-d via rapid replanning," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 853–864, 2014.
- [31] A. Kuntz, L. G. Torres, R. H. Feins, R. J. Webster, and R. Alterovitz, "Motion planning for a three-stage multilumen transoral lung access system," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3255–3261, IEEE, 2015.
- [32] J. Nilsson, J. Fredriksson, and A. C. Ödblom, "Verification of collision avoidance systems using reachability analysis," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 10676–10681, 2014.

- [33] A. M. Bayen, I. M. Mitchell, M. M. Oishi, and C. J. Tomlin, "Aircraft autolander safety analysis through optimal control-based reach set computation," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 68–77, 2007.
- [34] Y. Wang and F. J. Doyle III, "Reachability of particle size distribution in semi-batch emulsion polymerization," *AIChE journal*, vol. 50, no. 12, pp. 3049–3059, 2004.
- [35] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal toolbox (ET)," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 1498–1503, IEEE, 2006.
- [36] "CORA: A tool for continuous reachability analysis." <https://tumcps.github.io/CORA/>. Accessed: 2021-02-14.
- [37] "A toolbox of level set methods." <https://www.cs.ubc.ca/~mitchell/ToolboxLS/>. Accessed: 2021-10-04.
- [38] A. Devonport and M. Arcak, "Estimating reachable sets with scenario optimization," in *Learning for Dynamics and Control*, pp. 75–84, PMLR, 2020.
- [39] C. Fan, B. Qi, S. Mitra, and M. Viswanathan, "Dryvr: data-driven verification and compositional reasoning for automotive systems," in *International Conference on Computer Aided Verification*, pp. 441–461, Springer, 2017.
- [40] A. Devonport and M. Arcak, "Data-driven reachable set computation using adaptive gaussian process classification and monte carlo methods," in *2020 American Control Conference (ACC)*, pp. 2629–2634, IEEE, 2020.
- [41] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *ASME international mechanical engineering congress and exposition*, Citeseer, 1995.
- [42] A. Isidori, *Nonlinear control systems*, vol. 3. Springer, 1989.
- [43] H. K. Khalil, *Nonlinear systems*. Pearson, 2002.
- [44] R. T. Rockafellar, *Convex analysis*, vol. 18. Princeton university press, 1970.
- [45] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of convex analysis*. Springer Science & Business Media, 2004.
- [46] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms I: Fundamentals*, vol. 305. Springer Science & Business media, 2013.
- [47] R. Schneider, *Convex bodies: the Brunn–Minkowski theory*. No. 151, Cambridge University Press, 2014.

- [48] A. A. Liapounoff, “Sur les fonctions-vecteurs completement additives,” *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, vol. 4, no. 6, pp. 465–478, 1940.
- [49] P. R. Halmos, “The range of a vector measure,” *Bulletin of the American Mathematical Society*, vol. 54, no. 4, pp. 416–421, 1948.
- [50] J. Diestel and B. Faires, “On vector measures,” *Transactions of the American Mathematical Society*, vol. 198, pp. 253–271, 1974.
- [51] N. Dinculeanu, *Vector measures*. Elsevier, 2014.
- [52] P. McMullen, “On zonotopes,” *Transactions of the American Mathematical Society*, vol. 159, pp. 91–109, 1971.
- [53] G. Shephard, “Combinatorial properties of associated zonotopes,” *Canadian Journal of Mathematics*, vol. 26, no. 2, pp. 302–321, 1974.
- [54] H. S. M. Coxeter, *Regular polytopes*. Courier Corporation, 1973.
- [55] A. Girard, “Reachability of uncertain linear systems using zonotopes,” in *International Workshop on Hybrid Systems: Computation and Control*, pp. 291–305, Springer, 2005.
- [56] A. Girard and C. Le Guernic, “Zonotope/hyperplane intersection for hybrid systems reachability analysis,” in *International Workshop on Hybrid Systems: Computation and Control*, pp. 215–228, Springer, 2008.
- [57] M. Althoff, O. Stursberg, and M. Buss, “Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes,” *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 233–249, 2010.
- [58] M. Althoff and B. H. Krogh, “Zonotope bundles for the efficient computation of reachable sets,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 6814–6821, IEEE, 2011.
- [59] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz, “Constrained zonotopes: A new tool for set-based estimation and fault detection,” *Automatica*, vol. 69, pp. 126–136, 2016.
- [60] A. S. Adimoolam and T. Dang, “Using complex zonotopes for stability verification,” in *2016 American Control Conference (ACC)*, pp. 4269–4274, IEEE, 2016.
- [61] M. Althoff, “Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets,” in *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*, pp. 173–182, 2013.

- [62] N. Kochdumper and M. Althoff, “Sparse polynomial zonotopes: A novel set representation for reachability analysis,” *IEEE Transactions on Automatic Control*, 2020.
- [63] D. Cox, J. Little, and D. OShea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.
- [64] Y. Zheng and K. Yamane, “Generalized distance between compact convex sets: Algorithms and applications,” *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 988–1003, 2015.
- [65] S. Hornus, “Detecting the intersection of two convex shapes by searching on the 2-sphere,” *Computer-Aided Design*, vol. 90, pp. 71–83, 2017.
- [66] A. Grothendieck, *Résumé de la théorie métrique des produits tensoriels topologiques*. : Soc. de Matemática de São Paulo, 1956.
- [67] N. Alon and A. Naor, “Approximating the cut-norm via Grothendieck’s inequality,” in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 72–80, 2004.
- [68] G. Kindler, A. Naor, and G. Schechtman, “The UGC hardness threshold of the L_p Grothendieck problem,” *Mathematics of Operations Research*, vol. 35, no. 2, pp. 267–283, 2010.
- [69] A. Girard and C. Le Guernic, “Efficient reachability analysis for linear systems using support functions,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 8966–8971, 2008.
- [70] M. Althoff, “An introduction to CORA 2015,” in *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015.
- [71] J. Yong and X. Y. Zhou, *Stochastic controls: Hamiltonian systems and HJB equations*, vol. 43. Springer Science & Business Media, 1999.
- [72] S. Haddad and A. Halder, “Certifying the intersection of reach sets of integrator agents with set-valued input uncertainties,” *IEEE Control Systems Letters*, vol. 6, pp. 2852–2857, 2022.
- [73] S. Haddad and A. Halder, “A note on the hausdorff distance between norm balls and their linear maps,” *Set-Valued and Variational Analysis*, vol. 31, no. 3, p. 30, 2023.
- [74] E. D. Bolker, “A class of convex bodies,” *Transactions of the American Mathematical Society*, vol. 145, pp. 323–345, 1969.

- [75] R. Schneider and W. Weil, “Zonoids and related topics,” in *Convexity and its Applications*, pp. 296–317, Springer, 1983.
- [76] P. Goodey and W. Wolfgang, “Zonoids and generalisations,” in *Handbook of Convex Geometry*, pp. 1297–1326, Elsevier, 1993.
- [77] J. Bourgain, J. Lindenstrauss, and V. Milman, “Approximation of zonoids by zonotopes,” *Acta Mathematica*, vol. 162, no. 1, pp. 73–141, 1989.
- [78] S. Haddad and A. Halder, “The convex geometry of integrator reach sets,” in *2020 American Control Conference (ACC)*, pp. 4466–4471, IEEE, 2020.
- [79] A. B. Kurzhanski and P. Varaiya, *Dynamics and Control of Trajectory Tubes: Theory and Computation*, vol. 85. Springer, 2014.
- [80] C. Vinzant, “The geometry of spectrahedra,” in *Sum of Squares: Theory and Applications*, pp. 11–36, American Mathematical Society, 2020.
- [81] A. Tarski, “A decision method for elementary algebra and geometry,” in *Quantifier elimination and cylindrical algebraic decomposition*, pp. 24–84, Springer, 1998.
- [82] A. Seidenberg, “A new decision method for elementary algebra,” *Annals of Mathematics*, pp. 365–374, 1954.
- [83] J. Bochnak, M. Coste, and M.-F. Roy, *Real algebraic geometry*, vol. 36. Springer Science & Business Media, 2013.
- [84] G. Blekherman, P. A. Parrilo, and R. R. Thomas, *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.
- [85] G. Zaimi, “A polynomial implicitization.” MathOverflow.
- [86] H. S. Wilf, *generatingfunctionology*. CRC Press, 2005.
- [87] L. Kronecker, *Zur Theorie der Elimination einer Variablen aus zwei algebraischen Gleichungen*. Buchdruckerei der Königl. Akademie der Wissenschaften (G. Vogt), 1881.
- [88] R. Salem, *Algebraic numbers and Fourier analysis*. Wadsworth Publishing Company, 1983.
- [89] R. Schneider, “Zonoids whose polars are zonoids,” *Proceedings of the American Mathematical Society*, vol. 50, no. 1, pp. 365–368, 1975.
- [90] Y. Lonke, “On zonoids whose polars are zonoids,” *Israel Journal of Mathematics*, vol. 102, no. 1, pp. 1–12, 1997.

- [91] J. W. Helton and V. Vinnikov, "Linear matrix inequality representation of sets," *Communications on Pure and Applied Mathematics*, vol. 60, no. 5, pp. 654–674, 2007.
- [92] J. W. Helton and J. Nie, "Semidefinite representation of convex sets," *Mathematical Programming*, vol. 122, no. 1, pp. 21–64, 2010.
- [93] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.
- [94] OEIS Foundation Inc., "The On-Line Encyclopedia of Integer Sequences." <http://oeis.org/A107254>, 2019.
- [95] S. R. Finch, *Mathematical constants*. Cambridge University Press, 2003.
- [96] N. G. De Bruijn, *Asymptotic methods in analysis*, vol. 4. Courier Corporation, 1981.
- [97] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, vol. 55. US Government Printing Office, 1970.
- [98] F. C. Schweppe, *Uncertain dynamic systems*. Prentice Hall, 1973.
- [99] F. Chernousko, "Optimal guaranteed estimates of indeterminacies with the aid of ellipsoids, I," *Engineering Cybernetics*, vol. 18, no. 3, pp. 1–9, 1980.
- [100] F. Chernousko, "Guaranteed ellipsoidal estimates of uncertainties in control problems," *IFAC Proceedings Volumes*, vol. 14, no. 2, pp. 869–874, 1981.
- [101] D. Maksarov and J. Norton, "State bounding with ellipsoidal set description of the uncertainty," *International Journal of Control*, vol. 65, no. 5, pp. 847–866, 1996.
- [102] 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.
- [103] T. Alamo, J. M. Bravo, and E. F. Camacho, "Guaranteed state estimation by zonotopes," *Automatica*, vol. 41, no. 6, pp. 1035–1043, 2005.
- [104] 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)*, pp. 4040–4045, IEEE, 2018.
- [105] A. Halder, "Smallest ellipsoid containing p-sum of ellipsoids with application to reachability analysis," *IEEE Transactions on Automatic Control*, 2020.

- [106] A. Kurzhanskiĭ and I. Vályi, *Ellipsoidal calculus for estimation and control*. Nelson Thornes, 1997.
- [107] F. John, “Extremum problems with inequalities as subsidiary conditions,” *Studies and Essays: Courant Anniversary Volume, presented to R. Courant on his 60th Birthday*, pp. 187–204, 1948.
- [108] M. Henk, “Löwner-John ellipsoids,” *Documenta Math*, vol. 95, p. 106, 2012.
- [109] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.
- [110] M. Grant and S. Boyd, “CVX: Matlab Software for Disciplined Convex Programming, version 2.1.” <http://cvxr.com/cvx>, Mar 2014.
- [111] A. Kurzhanski and I. Vályi, *Ellipsoidal calculus for estimation and control*. Springer, 1997.
- [112] S. Haddad and A. Halder, “Anytime ellipsoidal over-approximation of forward reach sets of uncertain linear systems,” in *Proceedings of the Workshop on Computation-Aware Algorithmic Design for Cyber-Physical Systems*, pp. 20–25, 2021.
- [113] S. Kousik, A. Dai, and G. X. Gao, “Ellipsotopes: Uniting ellipsoids and zonotopes for reachability analysis and fault detection,” *IEEE Transactions on Automatic Control*, 2022.
- [114] N. Kochdumper, C. Schilling, M. Althoff, and S. Bak, “Open-and closed-loop neural network verification using polynomial zonotopes,” in *NASA Formal Methods Symposium*, pp. 16–36, Springer, 2023.
- [115] H. Witsenhausen, “A remark on reachable sets of linear systems,” *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 547–547, 1972.
- [116] A. Girard, C. Le Guernic, and O. Maler, “Efficient computation of reachable sets of linear time-invariant systems with inputs,” in *Hybrid Systems: Computation and Control: 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006. Proceedings 9*, pp. 257–271, Springer, 2006.
- [117] A. A. Kurzhanskiy and P. Varaiya, “Ellipsoidal techniques for reachability analysis of discrete-time linear systems,” *IEEE Transactions on Automatic Control*, vol. 52, no. 1, pp. 26–38, 2007.
- [118] S. Kaynama and M. Oishi, “Schur-based decomposition for reachability analysis of linear time-invariant systems,” in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 69–74, IEEE, 2009.

- [119] S. Haddad and A. Halder, “Boundary and taxonomy of integrator reach sets,” in *2022 American Control Conference (ACC)*, pp. 4133–4138, IEEE, 2022.
- [120] S. Haddad and A. Halder, “The curious case of integrator reach sets, part i: Basic theory,” *Transactions on Automatic Control*, 2023.
- [121] Z. Artstein, “Yet another proof of the Lyapunov convexity theorem,” *Proceedings of the American Mathematical Society*, vol. 108, no. 1, pp. 89–91, 1990.
- [122] I. Ekeland, P. Marcellini, A. Marino, M. Tosques, C. Olech, G. Pianigiani, T. Rockafeller, M. Valadier, and C. Olech, “The Lyapunov theorem: its extensions and applications,” in *Methods of Nonconvex Analysis: Lectures given at the 1st Session of the Centro Internazionale Matematico Estivo (CIME) held at Varenna, Italy, June 15–23, 1989*, pp. 84–103, Springer, 1990.
- [123] P. Antsaklis and A. Michel, *A Linear Systems Primer*. Massachusetts: Birkhauser, 2007.
- [124] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2012.
- [125] J. Lee, *Introduction to topological manifolds*, vol. 202. Springer Science & Business Media, 2010.
- [126] L. Montejano, “Some results about Minkowski addition and difference,” *Mathematika*, vol. 43, no. 2, pp. 265–273, 1996.
- [127] A. Lin and S.-P. Han, “On the distance between two ellipsoids,” *SIAM Journal on Optimization*, vol. 13, no. 1, pp. 298–308, 2002.
- [128] S. Iwata, Y. Nakatsukasa, and A. Takeda, “Computing the signed distance between overlapping ellipsoids,” *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 2359–2384, 2015.
- [129] S. Haddad and A. Halder, “The curious case of integrator reach sets, Part I: Basic theory,” *arXiv preprint arXiv:2102.11423*, 2021.
- [130] M. Grant and S. Boyd, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control* (V. Blondel, S. Boyd, and H. Kimura, eds.), Lecture Notes in Control and Information Sciences, pp. 95–110, Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.
- [131] C. Berge, *Topological Spaces: including a treatment of multi-valued functions, vector spaces, and convexity*. Translated into English by E.M. Patterson, Macmillan, New York, 1963.

- [132] M. C. Campi and S. Garatti, “The exact feasibility of randomized solutions of uncertain convex programs,” *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.
- [133] M. Campi, G. Calafiore, and S. Garatti, “Interval predictor models: Identification and reliability,” *Automatica*, vol. 45, no. 2, pp. 382–392, 2009.
- [134] S. Garatti and M. C. Campi, “Risk and complexity in scenario optimization,” *Mathematical Programming*, vol. 191, no. 1, pp. 243–279, 2022.
- [135] R. Tempo, G. Calafiore, F. Dabbene, *et al.*, *Randomized algorithms for analysis and control of uncertain systems: with applications*, vol. 7. Springer, 2013.
- [136] J. E. Littlewood and G. H. Hardy, *Inequalities*. Cambridge University Press, 1988.
- [137] S. Haddad and A. Halder, “Convex and nonconvex sublinear regression with application to data-driven learning of reach sets,” *arXiv preprint arXiv:2210.01919*, 2022.
- [138] P. Awasthi, B. Kalantari, and Y. Zhang, “Robust vertex enumeration for convex hulls in high dimensions,” in *International Conference on Artificial Intelligence and Statistics*, pp. 1387–1396, PMLR, 2018.
- [139] F. Hausdorff, *Grundzüge der mengenlehre*, vol. 7. : von Veit, 1914.
- [140] W. Hildenbrand, “Core and equilibria of a large economy.(psme-5),” in *Core and Equilibria of a Large Economy.(PSME-5)*, : Princeton university press, 2015.
- [141] D. Stoyan, W. S. Kendall, S. N. Chiu, and J. Mecke, *Stochastic geometry and its applications*. : John Wiley & Sons, 2013.
- [142] J. Serra, “Hausdorff distances and interpolations,” *Computational Imaging and Vision*, vol. 12, pp. 107–114, 1998.
- [143] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, “Comparing images using the Hausdorff distance,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 9, pp. 850–863, 1993.
- [144] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, “Robust face detection using the Hausdorff distance,” in *International conference on audio-and video-based biometric person authentication*, pp. 90–95, Springer, 2001.
- [145] F. De Blasi, “On the differentiability of multifunctions,” *Pacific Journal of Mathematics*, vol. 66, no. 1, pp. 67–81, 1976.
- [146] M. Serry and G. Reissig, “Overapproximating reachable tubes of linear time-varying systems,” *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 443–450, 2021.

- [147] O. Shisha and B. Mond, “Differences of means,” *Bulletin of the American Mathematical Society*, vol. 73, no. 3, pp. 328–333, 1967.
- [148] M. J. Atallah, “A linear time algorithm for the Hausdorff distance between convex polygons,” *Information Processing Letters*, vol. 17, no. 4, pp. 207–209, 1983.
- [149] E. Belogay, C. Cabrelli, U. Molter, and R. Shonkwiler, “Calculating the Hausdorff distance between curves,” *Information Processing Letters*, vol. 64, no. 1, 1997.
- [150] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, “Mesh: Measuring errors between surfaces using the Hausdorff distance,” in *Proceedings. IEEE international conference on multimedia and expo*, vol. 1, pp. 705–708, IEEE, 2002.
- [151] A. A. Taha and A. Hanbury, “An efficient algorithm for calculating the exact Hausdorff distance,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 11, pp. 2153–2163, 2015.
- [152] J.-L. Goffin and A. J. Hoffman, “On the relationship between the Hausdorff distance and matrix distances of ellipsoids,” *Linear algebra and its applications*, vol. 52, pp. 301–313, 1983.
- [153] H. Alt, B. Behrends, and J. Blömer, “Approximate matching of polygonal shapes,” *Annals of Mathematics and Artificial Intelligence*, vol. 13, no. 3, pp. 251–265, 1995.
- [154] H. Alt, P. Braß, M. Godau, C. Knauer, and C. Wenk, “Computing the hausdorff distance of geometric patterns and shapes,” *Discrete and Computational Geometry: The Goodman-Pollack Festschrift*, pp. 65–76, 2003.
- [155] S. König, “Computational aspects of the Hausdorff distance in unbounded dimension,” *Journal of Computational Geometry*, vol. 5, no. 1, pp. 250–274, 2014.
- [156] P. Jungeblut, L. Kleist, and T. Miltzow, “The complexity of the Hausdorff distance,” *arXiv preprint arXiv:2112.04343*, 2021.
- [157] T. Marošević, “The Hausdorff distance between some sets of points,” *Mathematical Communications*, vol. 23, no. 2, pp. 247–257, 2018.
- [158] S. Rahal and Z. Li, “Norm induced polyhedral uncertainty sets for robust linear optimization,” *Optimization and Engineering*, pp. 1–37, 2021.
- [159] D. Bertsimas, D. Pachamanova, and M. Sim, “Robust linear optimization under general norms,” *Operations Research Letters*, vol. 32, no. 6, pp. 510–516, 2004.
- [160] “DCCP Python package, GitHub repository.” URL: <https://github.com/cvxgrp/dccp>.

- [161] X. Shen, S. Diamond, Y. Gu, and S. Boyd, “Disciplined convex-concave programming,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 1009–1014, IEEE, 2016.
- [162] J. M. Hendrickx and A. Olshevsky, “Matrix p -norms are NP-hard to approximate if $p \neq 1, 2, \infty$,” *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 5, pp. 2802–2812, 2010.
- [163] D. Steinberg, “Computation of matrix norms with applications to robust optimization,” *Research thesis, Technion-Israel University of Technology*, vol. 2, 2005.
- [164] A. Bhaskara and A. Vijayaraghavan, “Approximating matrix p -norms,” in *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pp. 497–511, SIAM, 2011.
- [165] B. Barak, F. G. Brandao, A. W. Harrow, J. Kelner, D. Steurer, and Y. Zhou, “Hypercontractivity, sum-of-squares proofs, and their applications,” in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 307–326, 2012.
- [166] V. Bhattiprolu, M. Ghosh, V. Guruswami, E. Lee, and M. Tulsiani, “Approximability of $p \rightarrow q$ matrix norms: generalized Krivine rounding and hypercontractive hardness,” in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1358–1368, SIAM, 2019.
- [167] L. Gross, “Logarithmic Sobolev inequalities,” *American Journal of Mathematics*, vol. 97, no. 4, pp. 1061–1083, 1975.
- [168] L. Saloff-Coste, “Lectures on finite Markov chains. Lectures on probability theory and statistics (Saint-Flour, 1996), 301–413,” *Lecture Notes in Math*, vol. 1665, 1997.
- [169] P. Biswal, “Hypercontractivity and its applications,” *arXiv preprint arXiv:1101.2913*, 2011.
- [170] C. Wang, D.-S. Zheng, G.-L. Chen, and S.-Q. Zhao, “Structures of p -isometric matrices and rectangular matrices with minimum p -norm condition number,” *Linear algebra and its applications*, vol. 184, pp. 261–278, 1993.
- [171] C.-K. Li and W. So, “Isometries of ℓ_p norm,” *The American mathematical monthly*, vol. 101, no. 5, pp. 452–453, 1994.
- [172] J. Glueck, “What are the matrices preserving the ℓ^1 -norm?” MathOverflow. URL: <https://mathoverflow.net/q/288084> (version: 2017-12-09).
- [173] R. J. Aumann, “Integrals of set-valued functions,” *Journal of Mathematical Analysis and Applications*, vol. 12, no. 1, pp. 1–12, 1965.

-
- [174] K. G. Guseinov, O. Ozer, E. Akyar, and V. Ushakov, "The approximation of reachable sets of control systems with integral constraint on controls," *Nonlinear Differential Equations and Applications NoDEA*, vol. 14, no. 1, pp. 57–73, 2007.
- [175] D. Dueri, S. V. Raković, and B. Açıkmese, "Consistently improving approximations for constrained controllability and reachability," in *2016 European Control Conference (ECC)*, pp. 1623–1629, IEEE, 2016.
- [176] R. W. Brockett, *Finite dimensional linear systems*. : John Wiley & Sons, Inc., 1970.
- [177] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317. Springer Science & Business Media, 2009.
- [178] C. A. Holloway, "On the estimation of convex functions," *Operations Research*, vol. 27, no. 2, pp. 401–407, 1979.
- [179] E. Lim and P. W. Glynn, "Consistency of multidimensional convex regression," *Operations Research*, vol. 60, no. 1, pp. 196–208, 2012.
- [180] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *International Conference on Machine Learning*, pp. 146–155, PMLR, 2017.
- [181] Y. Chen, Y. Shi, and B. Zhang, "Optimal control via neural networks: A convex approach," *arXiv preprint arXiv:1805.11835*, 2018.
- [182] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [183] M. Fazlyab, M. Morari, and G. J. Pappas, "Probabilistic verification and reachability analysis of neural networks via semidefinite programming," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 2726–2731, IEEE, 2019.
- [184] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, vol. 2. Springer Science & Business Media, 1993.
- [185] L. Doeser, P. Nilsson, A. D. Ames, and R. M. Murray, "Invariant sets for integrators and quadrotor obstacle avoidance," in *2020 American Control Conference (ACC)*, pp. 3814–3821, IEEE, 2020.
- [186] A. Serrani, A. Isidori, C. I. Byrnes, and L. Marconi, "Recent advances in output regulation of nonlinear systems," *Nonlinear control in the year 2000 volume 2*, pp. 409–419, 2001.
- [187] G. M. Ziegler, *Lectures on polytopes*, vol. 152. Springer Science & Business Media, 2012.

Bibliography

- [188] T. M. Apostol, “An elementary view of Euler’s summation formula,” *The American Mathematical Monthly*, vol. 106, no. 5, pp. 409–418, 1999.
- [189] E. Hairer and G. Wanner, *Analysis by its History*, vol. 2. Springer Science & Business Media, 2006.
- [190] N. De Bruijn, “On some multiple integrals involving determinants,” *J. Indian Math. Soc.*, vol. 19, pp. 133–151, 1955.
- [191] J. Milnor and D. W. Weaver, *Topology from the differentiable viewpoint*, vol. 21. Princeton university press, 1997.
- [192] D. Kalman, “Generalized Fibonacci numbers by matrix methods,” *Fibonacci Quart.*, vol. 20, no. 1, pp. 73–76, 1982.
- [193] W. Y. Chen and J. D. Louck, “The combinatorial power of the companion matrix,” *Linear Algebra and its Applications*, vol. 232, pp. 261–278, 1996.
- [194] R. B. Taher and M. Rachidi, “On the matrix powers and exponential by the r -generalized Fibonacci sequences methods: the companion matrix case,” *Linear Algebra and Its Applications*, vol. 370, pp. 341–353, 2003.
- [195] L. P. Eisenhart, *Riemannian geometry*, vol. 51. Princeton university press, 1997.
- [196] R. Shaw, “Vector cross products in n dimensions,” *International Journal of Mathematical Education in Science and Technology*, pp. 803–816, 2006.
- [197] L. N. Trefethen and D. Bau, *Numerical linear algebra*, vol. 181. Siam, 2022.
- [198] G. Bennett, V. Goodman, and C. Newman, “Norms of random matrices,” *Pacific Journal of Mathematics*, vol. 59, no. 2, pp. 359–365, 1975.
- [199] O. Guédon, A. Hinrichs, A. E. Litvak, and J. Prochno, “On the expectation of operator norms of random matrices,” in *Geometric aspects of functional analysis*, pp. 151–162, : Springer, 2017.