Set Programming: Theory and Computation

Benoît Legat

Thesis submitted in partial fulfillment of the requirements for the Degree of Doctor in Applied Sciences

June 2020

ICTEAM
Louvain School of Engineering
Université catholique de Louvain
Louvain-la-Neuve
Belgium

Thesis Committee:

Pr. Raphaël M. **Jungers** (Advisor)	UCLouvain/ICTEAM, Belgium
Pr. Pablo A. **Parrilo** (Advisor)	MIT/LIDS, USA
Pr. Yurii **Nesterov**	UCLouvain/CORE, Belgium
Pr. Anthony **Papavasiliou**	UCLouvain/CORE, Belgium
Pr. Paulo **Tabuada**	UCLA/CyPhyLab, Belgium
Pr. Roland **Keunings** (Chair)	UCLouvain/ICTEAM, Belgium

Set Programming: Theory and Computation

by Benoît Legat

ICTEAM
Université catholique de Louvain
Place Sainte-Barbe, 2
1348 Louvain-la-Neuve
Belgium

This work was partially supported by the F.R.S-FNRS.

Preamble

The complexity of systems that are relevant to engineering today has grown tremendously. The control techniques based on frequency analysis that were perfectly adequate for simple systems tend to be difficult to use for more complex systems. While the windup issues generated by ignoring constraints on the controller may be worked around in simple cases, it is an indicator of the need to consider alternative techniques when state space and controller constraints play a fundamental role in a control problem. On the other hand, techniques based on optimization such as Lyapunov functions and Model Predictive Control (MPC) may be considered overkill for simple systems but they thrive in the complexity of modern systems as reported for instance in [May14]:

> [The phenomenal success of MPC] in the process industries is well described in [QB03] and was mainly due to its conceptual simplicity and its ability to handle easily and effectively complex systems with hard control constraints and many inputs and outputs.

An important challenge arising for these complex systems is the need to obtain sets satisfying given properties. While the search over all sets is in general intractable, there is usually a well defined objective for the set to be computed that entirely depends on the application. For instance, does the set need to outer or inner approximate a given set or does it need to satisfy a given property ? If several sets are feasible, which one should be chosen ? The one with maximal/minimal volume, the one that contains a point the furthest possible in a given direction ? How should the set be represented ? By inequalities that should be satisfied for the set ? Can it be represented as the projection of a simpler set ?

Once the intended end-use of the set as well as the properties it should satisfy are clarified, a specific family of sets, often called *template*, is chosen to formulate the search of the appropriate set as a problem that is numerically tractable. The algorithms employed vary considerably depending on the choice of template but always largely rely on solving optimization problems.

In classical numerical optimization, the introduction of a modeling layer on top of optimization solvers has been a key enabler, allowing different communities to share emergent technologies more easily. As detailed in [Orc84],

it started in the early history of numerical optimization with a punch-card input format called MPS [IBM76]. Due to the lack of convenience of the MPS format, algebraic modeling languages were developed such as AMPL [FGK90] and GAMS [BKM88]. Later, high-level interpreted languages were developed that allowed to write numerical algorithms in a high-level language, while the core of the methods used were implemented in low-level compiled languages such as C or Fortran, e.g. the optimized LAPACK [And+99] library for numerical computing. For numerical algorithms written in these high-level languages that needed to solve optimization programs, it became essential to be able to access algebraic modeling frameworks embedded in the language. This led to the creation CVX [GB14] and YALMIP [Lof04] in MATLAB™ and CVXPY [DB16] and Pyomo [Har+17] in Python. For large-scale problems and problems requiring substantial transformations, the interpreted nature of the high-level languages led to significant time spent in model transformation compared to the solve time. This issue was resolved by the creation of the high-level compiled language Julia and the `Convex.jl` [Ude+14] and `JuMP.jl` [DHL17b] algebraic modeling languages embedded in Julia that are built on top of the MathOptInterface [Leg+20] layer for interfacing with the solvers.

When the purpose is to compute sets rather than numbers, the numerical complexity is amplified, yet there does not currently exist such a clearly defined interface. The purpose of this thesis is to define this interface between

1. algorithms relying on the computation of the solution of a *set program*, that is, the computation of a set satisfying given properties and maximizing a given objective, these are independent of the way these sets are computed; and

2. algorithms for solving a specific class of set program for a given template, these are used as interchangeable solvers for solving the set programs formulated by algorithms described the previous item.

The set program searching over all sets, not specific to any template, is called the *generic set program*. The generic set program restricted to sets of a given template T is called the *set program for* T. Several templates are studied in this thesis including the classical templates polyhedra, zonotopes and ellipsoids but also more elaborate ones that provide a richer family of sets but are more complicated to implement both theoretically and algorithmically. These includes sublevel sets of polynomial forms, sets with support function that is the root of a polynomial, piecewise semi-ellipsoidal sets, and piecewise polynomial sets.

We study two questions in detail about set programs that both examine different aspects of duality.

- *Conic duality*

  First, we reinterpret the classical duality of conic programs for set programs. That is, given a definition of the convex combination of two sets, a template can be regarded as a convex cone where each point of the cone is a set of this template. As the set program for a given template is a restriction of the generic set program, its dual is a relaxation of the dual of the generic set program. This means for instance that while an infeasibility certificate of the dual of a generic set program is a valid certificate for any set, an infeasible certificate of the dual of a set program for a given template is only valid for this template. In other words, it is a relaxed infeasibility certificate for the generic set program.

  In numerous fields, the solution of relaxations can still be "rounded" to an actual solution. This is for instance the purpose of *heuristic callbacks* in mixed integer programming, the *randomized hyperplane rounding* technique for Max-Cut [GW95] or the *low rank rounding* of moment matrices to moments of Dirac measures in polynomial optimization; see Section 2.3.2 for more details. While the rounding techniques can be shown to work reliably on specific instances, a given guarantee on the quality of the rounded solution is appreciated as it ensures that an algorithm can rely on the performance of the rounding. We provide in Theorem 5.3.2 a rounding of the infeasibility certificate of a certain class of set program for the family of sublevel sets of polynomial forms as well as a guarantee on the quality of the rounded infeasibility certificate for the generic set program.

- *Geometric duality*

  Second, we focus on decision variables of set program that are constrained to be convex sets. In this case, the set can be represented in two different ways that take various forms depending on the template.

  1. The first way is with the *gauge* or *Minkowski* function of the set. In other words, the set is represented by several inequalities that all points in the set should satisfy. For polyhedra, this is called the H-representation. For ellipsoids, this is the positive definite matrix Q such that the ellipsoid is the set of points x such that $x^\top Qx \leq 1$.
  2. The second way is with the *support* function of the set. This represents the *polar* set by several inequalities that all points in

the polar set should satisfy. For polyhedra, this is called the V-representation. It represents the set as the convex hull of points. For ellipsoids, this is the positive definite matrix Q such that the ellipsoid is the set of points x such that $x^\top Q^{-1} x \leq 1$ and the polar of the ellipsoids is the set of points y such that $y^\top Q y \leq 1$.

An algorithm computing a convex set of a given template needs to choose the representation that is used for computation. Indeed, changing the representation is often not an option. For instance, computing the V-representation from the H-representation of a polyhedron is computationally expensive and when formulating a semidefinite program that searches for an ellipsoid $\mathcal{E}_Q$, the decision variable is either the matrix Q or the matrix Q^{-1}. Moreover, the choice of representation for a given set may affect the choice for another set. For instance, given an inclusion constraint between two sets, if both sets are represented with their support function, this can be formulated as an inequality between the support function but if one set is represented with its support function and the other one with its gauge function, it is not clear how the inclusion can be encoded in terms of the support and gauge functions[1].

The key question is therefore: which representation should be used for each set in the set program ? We address this question for specific cases of inclusion constraints in Theorem 4.3.1 and Theorem 4.4.1 and highlight the fact that this answer is essentially independent from the template. The choice is fundamentally based on convex properties of gauge or support functions and on the type of inclusion constraint the sets are involved in. This sheds a light on this choice of representation which often resort to template-specific arguments.

The thesis is organized as follows. The first part introduces background materials while the second part builds on these results to formulate our contributions. The first part is divided into three chapters. Chapter 1 provides the background on convex analysis as well as algebraic geometry needed for the thesis. Then, Chapter 2 surveys optimization results and algorithms that the thesis relies on. Finally, Chapter 3 introduces the classes of systems under consideration as well as classical techniques used to compute invariant sets for these systems.

The second part is split into four chapters. In Chapter 4, we define the set programming interface. We first discuss the different formulations of generic

[1] except in specific cases, for instance checking the inclusion of a polyhedron given its V-representation in a polyhedron given its H-representation simply amounts to checking whether each element of the V-representation of the first set is inside all halfspaces in the H-representation of the second set, see Proposition 1.3.9.

set programs as well as the choice of representation for convex sets. Then we particularize these results to the different templates considered in the thesis. In Chapter 5, we explore the conic duality of set programs and provide a rounding procedure to deduce infeasibility certificates for the generic set program given an infeasibility certificate for the set program for sublevel sets of polynomials. This can be used to certify the unstability of a switched system. We provide as well a guarantee of the quality of the certificates that is based on the entropy of the language generated by the automaton formed by the constraints of the set program. In Chapter 6, we use the geometric duality of set programs to compute controlled invariant sets for control systems. This is applied to model predictive control and stochastic programming. In Chapter 7, we show how to formulate an approximation of the entropic cone as a hierarchy of set programs. We then show how to circumvent the curse of dimensionality in the high-dimensional polyhedral approximation of the entropic cone by approximating it only in a specific direction using dual dynamic programming.

Reproducibility We put a particular emphasis on the reproducibility of our result and the quality and reusability of the code. The codes used to obtain the results of Section 5.2, Section 5.3 and Example 4.0.1 are published on codeocean respectively in [LPJ19a], [LPJ20a] and [LRJ20a].

These codes rely on both new Julia packages we have developed and existing Julia packages. We now briefly mention the packages developed for this thesis in collaboration with several other contributors that we mention in the acknowledgment. Julia is a programming language particularly well suited for numerical computing. Its *multiple dispatch* mechanism as well as its *just-in-time* compilation nature make it particularly well suited for numerical computing [Bez+17].

Related to Section 1.3, the package `Polyhedra.jl` provides a unified interface for polyhedral computation library. Currently, CDD [Fuk03], LRS [Avi00] and QHull [BDH96] are supported through the `CDDLib.jl`, `LRSLib.jl` and `QHull.jl` packages respectively. The `ConvexHull.jl` pure julia library also implements the `Polyhedra.jl` interface. The main features of `Polyhedra.jl` are representation conversion, polyhedral projection, intersections, convex/conic hulls of union, Minkowski sum, linear image and preimage, redundancy removal and 2D visualization with `Plots.jl` and 3D visualization using `Makie.jl` or `MeshCat.jl`.

Related to Section 1.5, the package `MultivariatePolynomials.jl` provides a unified interface for implementation of polynomials. Two implementations are available: `DynamicPolynomials.jl` and `TypedPolynomials.jl`. The package `SemialgebraicSets.jl` implements datastructures for basic semialgebraic sets and varieties and implement the Buchberger's algorithm

as well as the approach presented in [MD95] for computing the elements of a zero-dimensional variety from the multiplication matrices; see Section 1.5.1.

Related to Chapter 2, `MathOptInterface.jl` defines an interface for communicating with optimization solvers implementing the bridges mechanism described in Section 2.1.2 and `JuMP.jl` provides an algebraic modeling language on top of this interface. We developed the MathOptInterface for the following SDP solvers: `CDCS.jl` [Zhe+20], `CSDP.jl` [Bor99], `DSDP.jl` [BYZ00], `Mosek.jl` [ApS19], `SCS.jl` [ODo+16], `SDPA.jl` [YFK03], `SDPT3.jl` [TTT03], `SDPNAL.jl` [YST15], and `SeDuMi.jl` [Stu99]

Related to Section 2.3, we developed `MultivariateBases.jl` that provides an interface for bases of multivariate polynomials as well as the implementation of some of them, `MultivariateMoments.jl` that implements datastructures for moment series and moment matrices as well as the atom extraction procedure described in Section 2.3.2 based on the method developed in `SemialgebraicSets.jl` for finding the elements of zero-dimensional varieties. `PolyJuMP.jl` that extends `JuMP.jl` and `MathOptInterface.jl` to polynomial variables, polynomial equalities and polynomial inequalities. `SumOfSquares.jl` that extends `JuMP.jl` and `MathOptInterface.jl` to sum-of-squares variables, sum-of-squares constraints.

Related to Section 2.4, we developed `StructDualDynProg.jl` that implements dual dynamic programming on arbitrary markov chains with support for both optimality and feasibility cuts. If too many cuts start getting accumulated, the package can eliminate cuts based on different heuristics. These heuristics are implements in `CutPruners.jl`.

Related to Chapter 3, we developed `MathematicalSystems.jl` that defines an interface for dynamical systems and implements datastructures for the commonl used ones. Related to Section 3.1, we developed `HybridSystems.jl` that extends `MathematicalSystems.jl` to hybrid systems.

Related to Chapter 4, we implemented `SetProg.jl` that extends `JuMP.jl` to set variables and constraints and implements the reformulation to `SumOfSquares.jl` in two steps: first determine the representation of the set variables, second reformulate the objective and each constraint independently. The template of each variable is specified at the creation of the variable and the rest of the set program is specified independently of the template.

Related to Chapter 5 and Chapter 6, we implemented `SwitchOnSafety.jl` that compute invariant sets, from a given template, of systems defined with `MathematicalSystems.jl` or hybrid systems defined with `HybridSystems.jl` using `SetProg.jl`. It also implements the alternative techniques for approximating the joint spectral radius described in Section 5.3.

Related to Chapter 7, we developed `EntropicCone.jl` for working with the Entropic Cone. It implements optimization on the entropic cone, as discussed in Section 7.8, using the implementation of Algorithm 3 included in

`StructDualDynProg.jl`.

The codes of Example 1.5.1, Example 2.3.1, Example 2.3.2 and Example 2.3.3 using the above mentioned packages are included inline in the text for ease of reproducibility.

Bibliographic notes

Conference Publications

1. B. Legat, R. M. Jungers, and P. A. Parrilo. "Generating unstable trajectories for Switched Systems via Dual Sum-Of-Squares techniques". In: *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. HSCC '16. Vienna, Austria: ACM, 2016, pp. 51–60. ISBN: 978-1-4503-3955-1. DOI: 10.1145/2883817.2883821. URL: http://doi.acm.org/10.1145/2883817.2883821.

2. C. Gomes, B. Legat, R. M. Jungers, and H. Vangheluwe. "Stable Adaptive Co-simulation: A Switched Systems Approach". In: *IUTAM Symposium on Co-Simulation and Solver Coupling*. Springer. 2019, pp. 81–97.

3. B. Legat, P. Tabuada, and R. M. Jungers. "Computing controlled invariant sets for hybrid systems with applications to model-predictive control". In: vol. 51. 16. 6th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2018. 2018, pp. 193–198. DOI: https://doi.org/10.1016/j.ifacol.2018.08.033. URL: http://www.sciencedirect.com/science/article/pii/S2405896318311480.

4. C. Gomes, R. M. Jungers, B. Legat, and H. Vangheluwe. "Minimally Constrained Stable Switched Systems and Application to Co-simulation". In: *57th IEEE Conference on Decision and Control*. IEEE. 2018.

Journal Publications

1. B. Legat, P. A. Parrilo, and R. M. Jungers. "An entropy-based bound for the computational complexity of a switched system". In: *IEEE Transactions on Automatic Control* (2019). DOI: 10.1109/TAC.2019.2902625.

2. B. Legat, P. Tabuada, and R. M. Jungers. "Sum-of-Squares methods for controlled invariant sets with applications to model-predictive control". In: *Nonlinear Analysis: Hybrid Systems* 36 (2020), p. 100858.

3. B. Legat, P. A. Parrilo, and R. M. Jungers. "Certifying unstability of switched systems using Sum of Squares Programming". In: *SIAM Journal on Control and Optimization* (2020).

4. B. Legat, S. Raković, and R. M. Jungers. "Piecewise semi-ellipsoidal control invariant sets". In: *IEEE Control Systems Letters* (2020).

Acknowledgments

My ability to complete this thesis is first rooted into my education and childhood. For this I need to thank my parents who have thaught me the value of hard work and made me curious and interested about science and technology. The unconditional support of my broader family through my childhood, my higher education and then this Phd were also invaluable for me.

Then, I would like to thank my two advisors, Raphaël and Pablo. Pablo welcomed me during my two first months of research and then invited me several times at MIT to continue to help me through my Phd. Raphaël patiently guided me through my master thesis and my Phd thesis. I am grateful for his dedication, responsiveness and bright advises. During my Phd, I was lucky to meet several researchers that took some of their time to share their knowledge and experience. For this, I want to thank Nikolaos Athanasopoulos, Didier Henrion, Vincent Leclere, Ian Morris, Saša V. Raković, Paulo Tabuada and Juan Pablo Vielma. The content of Section 5.4 is based on a joint collaboration with Cláudio Gomes and Hans Vangheluwe. Our numerous interaction, helped by the small distance between UAntwerpen and UCLouvain, were enjoyable and stimulating.

As mentioned in the introduction, the code developed for this thesis was a collaborative work. For this, I would like to thank Mathieu Besançon, Guilherme Bodin, Paul Breiding, Chris Coey, Carleton Coffrin, Robin Deits, Oscar Dowson, Marcelo Forets, Joaquim Dias Garcia, Céline Gérard, Joey Huchette, Lea Kapelevich, Elias Kuthe, Miles Lubin, Daisuke Oyama, François Pacaud, Gilles Peiffer, Christian Schilling, Sascha Timme, Tillmann Weisser and Ulf Worsøe for their help, advises, reviews and contributions to the packages we co-developed.

During his master thesis, Jean Bouchat helped me explore many ideas and although I was not able to include his results in this thesis, our interactions greatly inspired for many aspect of this thesis.

I am also grateful for the passionate interest of Alexis Libert and Ryan Albert in the study of polytopes and in particular in the definition of proper Chebyshev centers of Section 4.5.1 which originates from our vibrant debates on the matter sitted in chairlifts thousands of meters above sea level.

Last but not least, I want to thank Léa Paulus for her love and support. She patiently reviewed many of my texts and directed the design of most of the figures. Finally, I thank the rest of our small family made of our cat Al-

bus and horse Charly which, although their understanding of complex algebraic geometry remains limited, significantly helped me to remain motivated throughout this thesis.

Contents

Notation

Common notations used across the thesis are collected in this section. Other notations are defined in the text, sometimes just after they are used first as in (5.23) where $\operatorname{supp}(\mu)$ is first used in an formula and then follows its definition "where $\operatorname{supp}(\mu)$ is the support of μ".

Basics:

fields	$\mathbb{R}, \mathbb{C}, \mathbb{P}, \mathbb{Q}, \mathbb{Z}$
nonnegative integers	$\mathbb{N}$
standard basis vectors	e_i
$[m]$	$\{1, \dots, m\}$
$A^{-1}\mathcal{S}$ where A is a matrix and $\mathcal{S}$ is a set	$\{ x \mid Ax \in \mathcal{S} \}$
Dirac measure centered at c	δ_c
projection matrix on the coordinates $I \subseteq [n]$	$\pi_{I,n}$
projection matrix on the subspace $\mathcal{V}$	$\pi_{\mathcal{V}}$

Matrices:

$m \times n$ matrices	$\mathbb{R}^{m \times n}$
identity matrix	$I_n \subseteq \mathbb{R}^{n \times n}$
transpose of A	$A^\top$
ith column of A	$A_{:,i}$
ith row of A	$A_{i,:}$
matrix of ith to nth column of A	$A_{:,i:}$
matrix of ith to mth row of A	$A_{i:,:}$
p-norm	$\lVert \cdot \rVert_p$
$n \times n$ symmetric matrices	$\mathcal{S}^n$
$n \times n$ positive semidefinite matrices	$\mathcal{S}^n_+$
$n \times n$ positive definite matrices	$\mathcal{S}^n_{++}$
positive semidefinite	$\succeq$
positive definite	$\succ$
inverse of A	A^{-1}
pseudo-inverse of A	$A^\dagger$
invertible $n \times n$ matrices	$\mathrm{GL}(\mathbb{R}^n)$
condition number of A	$\kappa(A)$

Graphs:

$G(V, E)$	Graph with nodes V and edges E
$E^\top$	$\{ (v, u, \sigma) : (u, v, \sigma) \in E \}$
$G^\top(V, E^\top)$	Graph with reversed edges
E^k	kth cartesian power of E
E_k	subset of E^k containing all valid paths of length k
$s(i)$ where s is a path	ith node of the path
$s[i]$ where s is a path	ith edge of the path
$s(i:)$ where $s \in E^k$	$(s(i), \dots, s(k))$
$E_k(u, v)$	$\{ s \in E_k^- \mid s(1) = u, s(k+1) = v \}$
$E_k^-(v)$	$\{ s \in E_k^- \mid s(k+1) = v \}$
$E_k^+(v)$	$\{ s \in E_k^+ \mid s(1) = v \}$
$E_k^-[e]$	$\{ s \in E_k^- \mid s[k] = e \}$
$E_k^+[e]$	$\{ s \in E_k^+ \mid s[1] = e \}$
$d^-(v)$	indegree of v
$d^+(v)$	outdegree of v
$\Delta^-(G)$	$\max_{v \in V} d^-(v)$
$\Delta^+(G)$	$\max_{v \in V} d^+(v)$
$d_k^-(v)$	$\lvert E_k^-(v) \rvert$, number of paths of length k ending at v
$d_k^+(v)$	$\lvert E_k^+(v) \rvert$, number of paths of length k starting at v
$\Delta_k^-(G)$	$\max_{v \in V} d_k^-(v)$
$\Delta_k^+(G)$	$\max_{v \in V} d_k^+(v)$

Note that $\Delta_1^-(G) = \Delta^-(G)$, $\Delta_1^+(G) = \Delta^+(G)$ and for any k, $\Delta_k^+(G^\top) = \Delta_k^-(G)$.

The k-tuple $(\sigma_1, \sigma_2, \dots, \sigma_k)$ is said to be *G-admissible* if $\sigma_1, \dots, \sigma_k$ are the respective labels of a path of length k in G. We denote the set of all k-tuples of $[m]^k$ that are G-admissible as G_k. The sequence $\sigma_1, \sigma_2, \dots$ is G-admissible (resp. $G^\top$-admissible) if $(\sigma_1, \dots, \sigma_k)$ (resp. $(\sigma_k, \dots, \sigma_1)$) is G-admissible for any $k \geq 1$. We denote $A_{\sigma_k} \cdots A_{\sigma_1}$ as A_s where $s = (\sigma_1, \dots, \sigma_k)$ or s is a path with these respective labels.

Labelled graphs:

$(u, v, \sigma) \in E$	An edge from u to v with label σ
$(\sigma_1, \dots, \sigma_k)$ is G-admissible	$\sigma_1, \dots, \sigma_k$ are the respective labels of a path of length k
G_k	subset of $[m]^k$ contaning all G-admissible k-uples
$A_{(u,v,\sigma)}$	A_σ
A_s where $s \in E^k$	$A_{s[k]} \cdots A_{s[1]}$

Convex analysis:

$\mathrm{aff}(\mathcal{S})$	affine hull of $\mathcal{S}$
$\mathrm{conv}(\mathcal{S})$	convex hull of $\mathcal{S}$
$\mathrm{ray}(\mathcal{S})$	smallest cone containing $\mathcal{S}$
$\mathrm{cone}(\mathcal{S})$	convex cone generated by $\mathcal{S}$
C°	polar of a convex set C
$\mathcal{K}^*$	dual of a convex cone $\mathcal{K}$
$\overline{\mathcal{H}}_{a,\alpha}$	hyperplane $\{ x \mid \langle a, x \rangle = \alpha \}$
$\mathcal{H}_{a,\alpha}$	halfspace $\{ x \mid \langle a, x \rangle \leq \alpha \}$

Optimization:

linear matrix inequality	LMI
semidefinite program	SDP
sum-of-squares	SOS
$\alpha \in \mathbb{N}^n$ (for exponents of monomials)	$\lvert\alpha\rvert = \sum \alpha_i$
forms of degree $2d$	$\mathbb{R}_{\mathbf{2d}}[x]$
SOS polynomials in n variables	Σ_n
if n is clear	Σ
SOS polynomials in n variables, degree at most $2d$	$\Sigma_{n,2d}$
if n is clear	Σ_{2d}
SOS forms in n variables, degree at most $2d$	$\Sigma_{\mathbf{n,2d}}$
if n is clear	$\Sigma_{\mathbf{2d}}$
the dual of $\Sigma_{\mathbf{2d}}$	$\Sigma^*_{\mathbf{2d}}$

Algebra:

Part I

Background

Convex Algebraic Geometry

1

This chapter gives a brief introduction to the concepts of convex algebraic geometry that are used in this thesis. Convex algebraic geometry is at the intersection of the fields of convex analysis, covered in Section 1.2 and Section 1.3, algebraic geometry, covered in Section 1.4 and Section 1.5, and optimization, covered in Chapter 2.

1.1 Preliminaries

In order for this thesis to be self-contained, we cover in this section preliminaries that are used throughout the text.

Definition 1.1.1 (Semigroup)**.** Given a set $\mathcal{S}$ and a binary operation $\cdot$, if $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for all $a, b, c \in \mathcal{S}$ then $\mathcal{S}$ with $\cdot$ is a *semigroup*.

Definition 1.1.2 (Monoid)**.** Consider a semigroup $\mathcal{S}$ with $\cdot$. If there exists an element $e \in \mathcal{S}$ such that, for every element $a \in \mathcal{S}$, $e \cdot a = a \cdot e = a$ then $\mathcal{S}$ with $\cdot$ is a *monoid*.

Definition 1.1.3. Given elements $a_1, \ldots, a_m$ in a monoid $\mathcal{S}$ with $\cdot$, the *monoid generated by* $a_1, \ldots, a_m$ is the set

$$\{ a_{\sigma_k} \cdots a_{\sigma_2} a_{\sigma_1} \mid \sigma_1, \ldots, \sigma_k \in [m], k \geq 0 \}.$$

Given a set C and a function f, we define the following notation:

$$\begin{aligned} f(C) &= \{ f(x) \mid x \in C \} \\ f^{-1}(C) &= \{ x \mid f(x) \in C \} \end{aligned} \tag{1.1}$$

Note that f does not need to be injective in these definitions. By slight abuse of notation, we also use the notation AC, $A^{-1}C$ and $A^{-\top}C \coloneqq [A^\top]^{-1}C$ where A is a matrix.

Given a subset $\mathcal{S} \subseteq \mathbb{R}^n$, and a subset $I \subseteq [n]$, we denote by $\pi_{I,n}$ the projection matrix on the coordinates I. and by

The L^p norm is defined by

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p}.$$

The p-norm is decreasing in p.

Proposition 1.1.1. For any real numbers $1 \leq p \leq q$, we have

$$\|x\|_\infty \leq \|x\|_q \leq \|x\|_p.$$

with equality when x has at most one nonzero element.

The p-mean of a vector x of nonnegative numbers is defined by $\|x\|_p / \sqrt[p]{n}$. The p-mean is increasing in p.

Proposition 1.1.2 ([HLP52])**.** For any nonnegative integers $a_1, \ldots, a_n$ and positive real numbers $p \leq q$, we have the following inequalities

$$\sqrt[n]{\prod_{i=1}^{n} a_i} \leq \left(\frac{1}{n}\sum_{i=1}^{n} a_i^p\right)^{\frac{1}{p}} \leq \left(\frac{1}{n}\sum_{i=1}^{n} a_i^q\right)^{\frac{1}{q}} \leq \max\{\, a_i \mid i = 1, \ldots, n \,\}$$

with equality if and only if $a_1 = \ldots = a_n$.

Given $\lambda \in \Delta^{n-1}$, the λ-weighted p-mean of a vector x of nonnegative numbers is defined by

$$\sqrt[p]{\frac{1}{n}\sum_{i=1}^{n} \lambda_i x_i^m}$$

For any given $\lambda \in \Delta^{n-1}$, the λ-weighted p-mean is increasing in p.

Proposition 1.1.3 ([HLP52])**.** For any nonnegative integers $a_1, \ldots, a_n$, positive real numbers $p \leq q$ and $\lambda \in \Delta^{n-1}$,

$$\left(\sum_{i=1}^{n} \lambda_i a_i^p\right)^{\frac{1}{p}} \leq \left(\sum_{i=1}^{n} \lambda_i a_i^q\right)^{\frac{1}{q}} \leq \max\{\, a_i \mid i = 1, \ldots, n \,\}$$

with equality if and only if $a_1 = \ldots = a_n$.

Proposition 1.1.4 (Schur complement)**.** Consider the matrices $A \in \mathbb{R}^{n_1 \times n_1}$, $B \in \mathbb{R}^{n_1 \times n_2}$, $C \in \mathbb{R}^{n_2 \times n_1}$, $D \in \mathbb{R}^{n_2 \times n_2}$. Let

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

If D is invertible, the *Schur complement* of the block D of the matrix M is defined by

$$M/D = A - BD^{-1}C.$$

If A is invertible, the *Schur complement* of the block A of the matrix M is defined by

$$M/A = D - BA^{-1}C.$$

We have the following properties of the Schur complement: If D is invertible then the matrix M is positive definite if and only if D and M/D are positive definite. If A is invertible then the matrix M is positive definite if and only if A and M/A are positive definite. Moreover, if $A \in \mathcal{S}^{n_1}, D \in \mathcal{S}^{n_2}$ and $C = B^\top$, we have the following properties. If D is positive definite then the matrix M is positive semidefinite if and only if M/D is positive semidefinite. If A is positive definite then the matrix M is positive semidefinite if and only if M/A is positive semidefinite.

Proposition 1.1.5. Given a subset $\mathcal{S} \subseteq \mathbb{R}^n$ and matrices $A \in \mathbb{R}^{r\times n}, B \in \mathbb{R}^{r\times m}$, the following holds:

$$A\mathcal{S} + B\mathbb{R}^m = {\pi_{\mathrm{Im}(B)^\perp}}^{-1} \pi_{\mathrm{Im}(B)^\perp} A\mathcal{S}$$

where $\pi_{\mathrm{Im}(B)^\perp}$ is a projection on the orthogonal subspace of $\mathrm{Im}(B)$.

Proof. Given $x \in \mathcal{S}$ and $y \in \mathbb{R}^r$, we have $y \in A\{x\} + B\mathbb{R}^m$ if and only if $y - Ax \in \mathrm{Im}(B)$ or equivalently, $\pi_{\mathrm{Im}(B)^\perp} y = \pi_{\mathrm{Im}(B)^\perp} Ax$. □

1.2 Convex Analysis

This section defines convexity for functions and sets and details the convexity properties that will be used throughout the text. These results will be particularized in later sections for the representation of specific families of complex sets but this section highlights the properties that can be ftrmalized in the generic settings. The reader is referred to [Roc15; RW98; Sch13; HL12; Bal97] for more details and proofs of the results. The references to these books of the various results are given for this purpose.

Consider r points $x_1, \ldots, x_r \in \mathbb{R}^n$. A vector $\sum_{i=1}^r \lambda_i x_i$ is

- an *affine combination* if $\sum_{i=1}^r \lambda_i = 1$.
- a *conic combination* if $\lambda_i \geq 0$, $i = 1, \ldots, r$.
- a *convex combination* if it is affine and conic.

The set of λ that are the coefficient of a convex combination is the $r-1$-*dimensional standard simplex* denoted by Δ^{r-1}.

An *affine set* is a set closed to affine combination, a *cone* is a set closed to positive scalar multiplication, a *convex cone* is a set closed to conic combination and a *convex set* is a set closed to convex combination. An arbitrary intersection of affine sets (resp. cone, convex cone, convex set) is an affine set (resp. cone, convex cone, convex set).

The *affine hull* is of a set $\mathcal{S}$ is the smallest affine set containing $\mathcal{S}$,

$$\operatorname{aff}(\mathcal{S}) = \{ \lambda_1 x_1 + \cdots + \lambda_r x_r \mid x_1, \ldots, x_r \in \mathcal{S}, \lambda_1 + \cdots + \lambda_r = 1 \}.$$

The *convex hull* is of a set $\mathcal{S}$ is the smallest convex set containing $\mathcal{S}$,

$$\operatorname{conv}(\mathcal{S}) = \{ \lambda_1 x_1 + \cdots + \lambda_r x_r \mid x_1, \ldots, x_r \in \mathcal{S}, \lambda \in \Delta^{r-1} \}.$$

The smallest cone containing the origin and $\mathcal{S}$ is

$$\operatorname{ray}(\mathcal{S}) = \{ \lambda x \mid x \in \mathcal{S}, \lambda \in \mathbb{R}, \lambda \geq 0 \}$$

The *convex cone generated* by $\mathcal{S}$ is the smallest convex cone containing the origin and $\mathcal{S}$,

$$\operatorname{cone}(\mathcal{S}) = \operatorname{conv}(\operatorname{ray}(\mathcal{S})).$$

Subspaces are affine sets which contain the origin. Each non-empty affine set is the translate of a unique subspace. Its dimension is defined as the dimension of this subspace. The dimension of the empty set $\emptyset$ is -1 by convention.

A set of $m+1$ points $x_0, \ldots, x_m$ is said to be *affinely independent* if $\operatorname{aff}(\{x_0, \ldots, x_m\})$ is m-dimensional. In such case, for each point x in the affine hull, the parameter λ of the affine combination is unique and is called the *barycentric coordinate*.

Affine sets of dimension 0, 1 and 2 are called points, lines, planes, respectively. The $(n-1)$-dimensional affine sets are called hyperplanes. For any hyperplanes, there exists a, α such that the hyperplane is equal to $\overline{\mathcal{H}}_{a,\alpha} \triangleq \{ x \in \mathbb{R}^n \mid \langle x, a \rangle = \alpha \}$. Any affine subsets of $\mathbb{R}^n$ is the intersection of finitely many hyperplanes.

The set $\mathcal{H}_{a,\alpha} \triangleq \{ x \in \mathbb{R}^n \mid \langle x, a \rangle \leq \alpha \}$ is called an *halfspace* and is convex.

If $x_0, \ldots, x_m$ are affinely independent, $\operatorname{conv}(\{x_0, \ldots, x_m\})$ is an *m-dimensional simplex*. The standard $r-1$-dimensional simplex define above is $\operatorname{conv}\{e_1, \ldots, e_r\}$ where the vectors e_i are the vector of the canonical basis. When $m = 0, 1, 2$, or 3, the simplex is called a point, (closed) line segment, triangle or tetrahedron, respectively.

The dimension of a convex set is defined as the dimension of its affine hull. It is equal to the maximum of the dimension of the simplices contained in it.

A set of $d+2$ or more points is always affinely dependent.

Theorem 1.2.1 (Caratheodory [HL12, Theorem 1.3.6])**.** Let $\mathcal{S}$ be a collection of points of $\mathbb{R}^n$. If $x \in \text{conv}(\mathcal{S})$, then x is the convex combination of $n+1$ points of $\mathcal{S}$.

An inequality $\langle x, a\rangle \leq \alpha$ is said to be valid on a convex set C if it holds for any $x \in C$. A set $\mathcal{F} \subseteq C$ of a convex set C is a *face* if and only if there exists a valid linear inequality $\langle x, a\rangle \leq \alpha$ such that $\langle x, a\rangle = \alpha$ for any $x \in \mathcal{F}$. Such a hyperplane is called a *supporting hyperplane* of $\mathcal{F}$. A face of dimension 0, 1 or $\dim(C) - 1$ is called a vertex, edge or facet respectively. The face of dimension $\dim(C)$ is C itself and the face of dimension -1 is the emptyset. Faces of dimension between 0 and $\dim(C)-1$ are called *proper faces*. The faces of a convex set form a finite partially ordered set (poset) by containment.

1.2.1 Functional representation of convex sets

In this section, we discuss the relation between the different functions that represent a convex sets: the *gauge function*, the *indicator function* and the *support function*.

Definition 1.2.1 (Minkowski function)**.** We define the *gauge* or *Minkowski* function of a closed convex set $\mathcal{S}$ containing the origin as:

$$\text{g}(\mathcal{S}, x) = \min_{\gamma}\{\, \gamma : x \in \gamma\mathcal{S}, \gamma \geq 0 \,\}, \text{ for } x \in \mathbb{R}^n.$$

The set $\mathcal{S}$ is the 1-sublevel set of its Minkowski function and the Minkowski function is the only homogeneous function that has this property.

Definition 1.2.2 (Indicator function)**.** We define the *indicator function* of a set $\mathcal{S}$ as:

$$\delta(x|\mathcal{S}) = \begin{cases} 0 & \text{if } x \in \mathcal{S} \\ \infty & \text{otherwise.} \end{cases}$$

The conjugate function of the indicator function is the *support function*.

Definition 1.2.3 (Conjugate function)**.** The *conjugate*, also called the *Legendre-Fenchel transformation*, of a function $f : \mathbb{R}^n \to \mathbb{R}$ is the function

$$f^*(v) \triangleq \sup\{\, \langle v, x\rangle - f(x) \mid x \in \mathbb{R}^n \,\}.$$

Definition 1.2.4 (Support function)**.** We define the *support function* of a nonempty closed convex subset $\mathcal{S} \subseteq \mathbb{R}^n$ as:

$$\delta^*(y|\mathcal{S}) = \sup_{x \in \mathcal{S}} \langle y, x\rangle.$$

Proposition 1.2.1 ([Roc15, Theorem 14.5] [Sch13, Lemma 1.7.13])**.** Given a closed convex subset $\mathcal{S} \subseteq \mathbb{R}^n$ containing the origin, for all $x \in \mathbb{R}^n$, the following holds:

$$\mathrm{g}(\mathcal{S}, x) = \delta^*(x|\mathcal{S}^\circ).$$

While Proposition 1.2.1 allows to go from the gauge function of a set to the support function of its polar, it is sometimes desirable to compute the support function of the set itself or the gauge function of its polar.

Definition 1.2.5 (Polar function [Roc15, p. 136])**.** The *polar* of a nonnegative convex function f such that $f(0) = 0$ is the function

$$f^\circ(v) \triangleq \inf\{\, \mu \geq 0 \mid \langle v, x\rangle \leq 1 + \mu f(x), \forall x \,\}. \tag{1.2}$$

Remark 1.2.1. Note that if $f(x)$ is positively homogeneous, then (1.2) is equivalent to

$$f^\circ(v) \triangleq \inf\{\, \mu \geq 0 \mid \langle v, x\rangle \leq \mu f(x), \forall x \,\}.$$

Proposition 1.2.2 ([Roc15, Corollary 15.3.2])**.** Let f be a positively homogeneous function of degree p, where $1 < p < \infty$. Then $(pf)^{1/p}$ is a closed gauge whose polar is $(qf^*)^{1/q}$, where $1 < q < \infty$ and $1/p + 1/q = 1$.

Proposition 1.2.3. The polar of the unit ball of the p-norm for $1 < p < \infty$ is the unit ball of the q-norm where $1 < q < \infty$ and $1/p + 1/q = 1$.

Proof. Consider the function $f(x) = (x_1^p + \cdots + x_n^p)/p$. The conjugate of $f(x)$ is $f^*(x) = (x_1^q + \cdots + x_n^q)/q$ where $1 < q < \infty$ is such that $1/p + 1/q = 1$. As $(pf)^{1/p}$ is the p-norm and $(qf^*)^{1/q}$ is the q-norm, we can conclude with Proposition 1.2.2. □

Given a closed convex set $\mathcal{S}$ containing the origin, the function $\mathrm{g}^p(\mathcal{S}, x)/p$ is denoted $\mathrm{g}_p(\mathcal{S}, x)$ and referred to as the *gauge-like function of degree p*.

The inclusion of convex sets is equivalent to an inequality between the gauge functions.

Proposition 1.2.4. Consider two closed convex subsets $S_1, S_2 \subseteq \mathbb{R}^n$ containing the origin. The inclusion $S_1 \subseteq S_2$ is equivalent to the inequality $\mathrm{g}(S_1, x) \geq \mathrm{g}(S_2, x)$ for all $x \in \mathbb{R}^n$.

As the function $x \mapsto x^p$ is strictly increasing for $p > 1$, the inclusion can be verified as well using the gauge-like function.

Proposition 1.2.5. Consider two closed convex subsets $S_1, S_2 \subseteq \mathbb{R}^n$ containing the origin and a positive integer p. The inclusion $S_1 \subseteq S_2$ is equivalent to the inequality $\mathrm{g}_p(S_1, x) \geq \mathrm{g}_p(S_2, x)$ for all $x \in \mathbb{R}^n$.

The inclusion can be rewritten as an inequality between the support functions as well.

Proposition 1.2.6 ([Roc15, Corollary 13.1.1])**.** Consider two nonempty closed convex subsets $S_1, S_2 \subseteq \mathbb{R}^n$. The inclusion $S_1 \subseteq S_2$ is equivalent to the inequality $\delta^*(x|S_1) \leq \delta^*(x|S_2)$ for all $x \in \mathbb{R}^n$.

1.2.2 Operations preserving convexity

We consider in this section six operations that preserve the convexity of its operands: the intersection $\cap$, the convex hull of the union $\operatorname{conv} \cup$, the Minkowski sum $+$, the inverse sum $\sharp$, the linear image A and the linear preimage A^{-1}. In particular, we study their impact on the gauge function, support function and polar set. The results are summarized in Table 8.1.

Lemma 1.2.1 ([Roc15, Corollary 16.4.2])**.** Let $\mathcal{K}_i$ be a non-empty convex cone in $\mathbb{R}^n$ for each $i \in I$. Then

$$(\sum_{i \in \mathcal{K}} \mathcal{K}_i)^\circ = \bigcap_{i \in I} \mathcal{K}_i^\circ$$
$$(\bigcap_{i \in I} \operatorname{cl} \mathcal{K}_i)^\circ = \operatorname{cl}(\sum_{i \in I} \mathcal{K}_i^\circ).$$

Lemma 1.2.2 ([Roc15, Corollary 16.5.2])**.** Let C_i be a convex set in $\mathbb{R}^n$ for each $i \in I$. Then

$$\left(\operatorname{conv} \bigcup_{i \in I} C_i\right)^\circ = \bigcap_{i \in I} C_i^\circ$$
$$(\bigcap_{i \in I} \operatorname{cl} C_i)^\circ = \operatorname{cl}\left(\operatorname{conv} \bigcup_{i \in I} C_i^\circ\right).$$

The gauge function of the intersection of sets is obtained as follows in terms of the support functions of the two sets being intersected.

Proposition 1.2.7. Given closed convex sets $\mathcal{S}_1, \mathcal{S}_2$ containing the origin, the following holds:

$$\mathrm{g}(\mathcal{S}_1 \cap \mathcal{S}_2, x) = \max(\mathrm{g}(\mathcal{S}_1, x), \mathrm{g}(\mathcal{S}_2, x)).$$

The support function of the intersection of sets is the *infimal convolution* of the support function of each set.

Definition 1.2.6 (Proper convex function [Roc15, p. 24])**.** We say that a convex function $f(x)$ is *proper* if its epigraph is non-empty and contains no vertical lines.

Proposition 1.2.8 (Infimal convolution [Roc15, Theorem 5.4])**.** Let $f_1, \ldots, f_m$ be proper convex functions, their *infimal convolution*

$$(f_1 \square f_2 \square \cdots \square f_m)(x) = \inf\{ f_1(x_1) + \cdots, f_m(x_m) \mid x_1 + \cdots + x_m = x \}.$$

is convex.

Proposition 1.2.9 ([Roc15, Corollary 16.4.1])**.** Given nonempty closed convex sets $\mathcal{S}_1, \mathcal{S}_2$, the following holds:

$$\delta^*(y|\mathcal{S}_1 \cap \mathcal{S}_2) = (\delta^*(\cdot|\mathcal{S}_1) \square \delta^*(\cdot|\mathcal{S}_2))(y).$$

The support function of the convex hull of the union of sets is obtained as follows in terms of the support functions of the two sets.

Proposition 1.2.10 ([Roc15, Corollary 16.5.1])**.** Given nonempty closed convex sets $\mathcal{S}_1, \mathcal{S}_2$, the following holds:

$$\delta^*(y|\operatorname{conv}(\mathcal{S}_1 \cup \mathcal{S}_2)) = \max(\delta^*(y|\mathcal{S}_1), \delta^*(y|\mathcal{S}_2)).$$

Proof. It follows from Lemma 1.2.2, Proposition 1.2.1 and Proposition 1.2.7. □

The gauge function of the convex hull is obtained in as an infimal convolution.

Proposition 1.2.11. Given closed convex sets $\mathcal{S}_1, \mathcal{S}_2$ containing the origin, the following holds:

$$\mathrm{g}(\operatorname{conv}(\mathcal{S}_1 \cup \mathcal{S}_2), x) = (\mathrm{g}(\mathcal{S}_1, \cdot) \square \mathrm{g}(\mathcal{S}_2, \cdot)))(y).$$

Proof. It follows from Lemma 1.2.2, Proposition 1.2.1 and Proposition 1.2.9. □

The support function of the Minkowski sum of sets is obtained as follows in terms of the support functions of the two sets being summed.

Proposition 1.2.12 ([Roc15, Corollary 16.4.1])**.** Given nonempty closed convex sets $\mathcal{S}_1, \mathcal{S}_2$, the following holds:

$$\delta^*(y|\mathcal{S}_1 + \mathcal{S}_2) = \delta^*(y|\mathcal{S}_1) + \delta^*(y|\mathcal{S}_2).$$

The gauge function of the Minkowski sum of sets is the *inverse sum* of the gauge function of each set.

Proposition 1.2.13 (Inverse sum [Roc15, Theorem 5.8])**.** Let $f_1, \dots, f_m$ be proper convex functions, their *inverse sum*

$$(f_1 \mathbin{\sharp} f_2 \mathbin{\sharp} \cdots \mathbin{\sharp} f_m)(x) = \inf\{ \max(f_1(x_1), \dots, f_m(x_m)) \mid x_1 + \cdots + x_m = x \}.$$

is convex.

Proposition 1.2.14. Given closed convex sets $\mathcal{S}_1, \mathcal{S}_2$ containing the origin, the following holds:

$$\mathrm{g}(\mathcal{S}_1 + \mathcal{S}_2, x) = (\mathrm{g}(\mathcal{S}_1, \cdot) \mathbin{\sharp} \mathrm{g}(\mathcal{S}_2, \cdot))(x).$$

Proof. We have $x \in \mathcal{S}_1 + \mathcal{S}_2$ if and only if there exists $x_1 \in \mathcal{S}_1, x_2 \in \mathcal{S}_2$ such that $x_1 + x_2 = x$. That is, $\mathrm{g}(\mathcal{S}_1 + \mathcal{S}_2, x) \leq 1$ if and only if there exists x_1, x_2 such that $\max(\mathrm{g}(\mathcal{S}_1, x_1), \mathrm{g}(\mathcal{S}_2, x_2)) \leq 1$ and $x_1 + x_2 = x$. The result follows. □

Proposition 1.2.15 ([Roc15, Theorem 3.7])**.** Consider the convex subsets $C_1, C_2 \subseteq \mathbb{R}^n$, their *inverse sum*

$$C_1 \mathbin{\sharp} C_2 = \bigcup_{\lambda \in \Delta^1} \lambda_1 C_1 \cap \lambda_2 C_2$$

is convex.

Proposition 1.2.16. Consider the closed convex subsets $C_1, C_2 \subseteq \mathbb{R}^n$, The following holds

$$(C_1 + C_2)^\circ = C_1^\circ \mathbin{\sharp} C_2^\circ.$$

Proof. By Proposition 1.2.12, we have $\delta^*(y|\mathcal{S}_1 + \mathcal{S}_2) = \delta^*(y|\mathcal{S}_1) + \delta^*(y|\mathcal{S}_2)$. Therefore, $y \in (\mathcal{S}_1 + \mathcal{S}_2)^\circ$ if and only if there exists $0 < \lambda < 1$ such that $y \in \lambda \mathcal{S}_1^\circ$ and $y \in (1 - \lambda)\mathcal{S}_2^\circ$. □

Proposition 1.2.17. Given closed convex sets $\mathcal{S}_1, \mathcal{S}_2$ containing the origin, the following holds:

$$\mathrm{g}(\mathcal{S}_1 \mathbin{\sharp} \mathcal{S}_2, x) = \mathrm{g}(\mathcal{S}_1, x) + \mathrm{g}(\mathcal{S}_2, x).$$

Proof. It follows from Proposition 1.2.16, Proposition 1.2.1 and Proposition 1.2.12. □

Proposition 1.2.18. Given nonempty closed convex sets $\mathcal{S}_1, \mathcal{S}_2$, the following holds:

$$\delta^*(y|\mathcal{S}_1 \mathbin{\sharp} \mathcal{S}_2) = (\delta^*(\cdot|\mathcal{S}_1) \mathbin{\sharp} \delta^*(\cdot|\mathcal{S}_2))(y).$$

Proof. It follows from Proposition 1.2.16, Proposition 1.2.1 and Proposition 1.2.14. □

We have the following property for the Minkowski functions of linear preimages and support functions of linear images.

Proposition 1.2.19. Given a matrix $A \in \mathbb{R}^{n_1 \times n_2}$ and a closed convex subset $\mathcal{S} \subseteq \mathbb{R}^{n_1}$ containing the origin, for all $x \in \mathbb{R}^{n_2}$, the following holds:

$$\mathrm{g}(A^{-1}\mathcal{S}, x) = \mathrm{g}(\mathcal{S}, Ax). \tag{1.3}$$

Proposition 1.2.20 ([RW98, Corollary 11.24(c)] or [Roc15, Corollary 16.3.1])**.** Given a matrix $A \in \mathbb{R}^{n_1 \times n_2}$ and a nonempty closed convex set $\mathcal{S} \subseteq \mathbb{R}^{n_2}$, for all $y \in \mathbb{R}^{n_1}$, the following holds:

$$\delta^*(y|A\mathcal{S}) = \delta^*(A^\top y|\mathcal{S}). \tag{1.4}$$

Corollary 1.2.1. Given a projection $\pi_{I,n}$ on the coordinates $I \subseteq [n]$ and a nonempty closed convex set $\mathcal{S} \subseteq \mathbb{R}^{n_2}$, the following holds:

$$\delta^*(y|\pi_{I,n}(\mathcal{S})) = \delta^*(\pi_{I,n}^\top(y)|\mathcal{S}).$$

The following properties show the relation between the linear image of a convex set and its polar.

Proposition 1.2.21 ([Roc15, Corollary 16.3.2])**.** For any convex set $\mathcal{S}$, convex cone $\mathcal{K}$ and linear map A, the following equality holds:

$$(A\mathcal{C})^\circ = A^{-\top}\mathcal{C}^\circ, (A^{-1}(\mathrm{cl}\,\mathcal{C}))^\circ \qquad = \mathrm{cl}(A^\top(\mathcal{C}^\circ)), (A\mathcal{K})^* = A^{-\top}\mathcal{K}^*.$$

The Figure 1.1 illustrates the relation between these four propositions, showing that either of these four is a consequence of the three other ones.

$$\begin{array}{ccc} \mathrm{g}(A^{-1}\mathcal{S}, x) & = & \mathrm{g}(\mathcal{S}, Ax) \\ \| & & \| \\ \delta^*(x|A^\top\mathcal{S}^\circ) & = & \delta^*(Ax|\mathcal{S}^\circ) \end{array}$$

Figure 1.1: Relation between Proposition 1.2.1, Proposition 1.2.19, Proposition 1.2.20 and Proposition 1.2.21.

Proposition 1.2.21 plays a prominent role in Chapter 6. The following example provides a geometric intuition for it.

Example 1.2.1. We can rewrite $(A\mathcal{C})^\circ = A^{-\top}(\mathcal{C}^\circ)$ as follows:

$$x \in (A\mathcal{C})^\circ \iff A^\top x \in \mathcal{C}^\circ. \tag{1.5}$$

Let $\mathcal{C}$ be a sphere of radius 1 centered at $(0, 0, 2)$ and A be the linear projection on the plane $z = 0$. With these choices of $\mathcal{C}$ and A, the set $A\mathcal{C}$ is the circle of

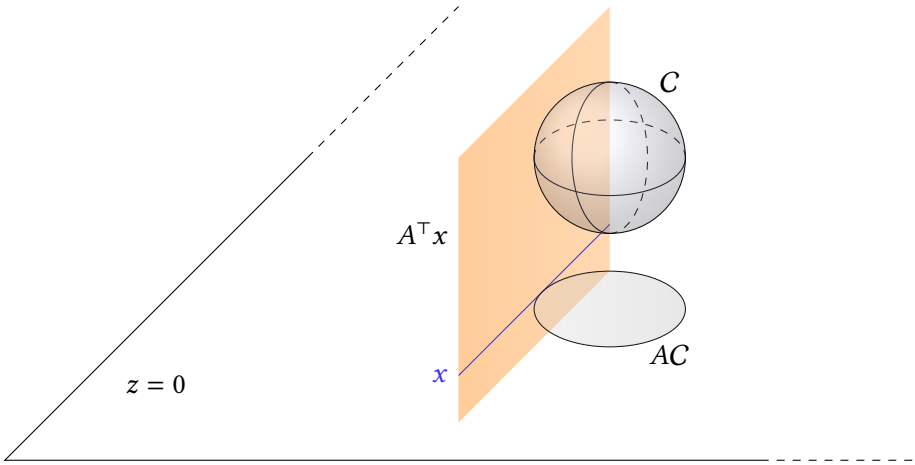


Figure 1.2: Illustration for Example 1.2.1.

radius 1 of the plane $z = 0$ centered at the origin. This is illustrated by the Figure 1.2. Let x be the normal vector of the line of the plane $z = 0$ shown in Figure 1.2. By (1.5), x is in the polar of the circle, if and only if $A^\top x$, which is the plane perpendicular to the plane $z = 0$ containing the line, is in the polar of the sphere.

The *barrier cone* of a convex set C is the set of y such that for some $\beta \in \mathbb{R}$, $\langle x, y\rangle \leq \beta$ for all $x \in C$. The polar of a set $\mathcal{S}$ is

$$\mathcal{S}^\circ = \{\, y \in \mathbb{R}^n \mid \langle x, y\rangle \leq 1, \forall x \in \mathcal{S} \,\}.$$

The polar transformation satisfies the following property: For any convex sets $\mathcal{S}, \mathcal{T}$,

$$\mathcal{S} \subseteq \mathcal{T} \Leftrightarrow \mathcal{T}^\circ \subseteq \mathcal{S}^\circ. \tag{1.6}$$

1.2.3 Polar correspondence of boundary points

Definition 1.2.7 (Tangent cone [BM15, Definition 4.6])**.** Given a closed convex set $\mathcal{S}$ and a distance function $d(\mathcal{S}, x)$, the *tangent cone* to $\mathcal{S}$ at x is defined as follows:

$$T_{\mathcal{S}}(x) = \left\{ y \mid \lim_{\tau \to 0} \frac{d(\mathcal{S}, x + \tau y)}{y} = 0 \right\}.$$

The tangent cone is a convex cone and is independent of the distance chosen.

For a convex set C, the *normal cone* is the polar of the tangent cone $N_C(x) = T_C^\circ(x)$.

Proposition 1.2.22. Given a closed convex set C containing the origin, if $\mathrm{g}_p(C, x)$ is differentiable at $x \in \partial C$ then $N_C(x) = \{\nabla \mathrm{g}_p(C, x)\}$.

The exposed face is also called the support set in [Sch13, Section 1.7.1].

Definition 1.2.8 (Exposed face [HL12, Definition 3.1.3])**.** Consider a nonempty closed convex set C. Given a vector $y \neq 0$, the *exposed face* of C associated to y is

$$F_C(y) = \{ x \in C \mid \langle x, y \rangle = \delta^*(y|C) \}.$$

Proposition 1.2.23 ([HL12, Proposition 3.1.4])**.** Consider a nonempty closed convex set C. For any $x \in C$ and nonzero vector y, $x \in F_C(y)$ if and only if $y \in N_C(x)$.

When the support function is differentiable, F_C is a singleton and may be directly obtained from z using the following result:

Proposition 1.2.24 ([Roc15, Corollary 25.1.2])**.** Given a nonempty closed convex set C, if $\delta^*(y|C)$ is differentiable at y then $F_C(y) = \{\nabla\delta^*(y|C)\}$.

For nonempty compact convex sets, the differentiability at y is even equivalent to the unicity of $F_C(y)$ [Sch13, Corollary 1.7.3].

A similar flavor of Proposition 1.2.23 in terms of conjugacy is given by the following:

Proposition 1.2.25 ([Roc15, Theorem 23.5])**.** For any proper convex function f and vectors x, y, $y \in \partial f(x)$ if and only if $x \in \partial f^*(y)$.

For differentiable functions, this allows to compute the conjugate of a function by inverting its gradient.

Corollary 1.2.2. For any proper convex differentiable function f and vectors x, y, $y = \nabla f(x)$ if and only if $x = \nabla f^*(y)$.

1.3 Polyhedra and Polytopes

Hyperplanes and halfspaces are respectively denoted by

$$\overline{\mathcal{H}}_{a,\alpha} = \{ x \mid a^\top x = \alpha \}$$
$$\mathcal{H}_{a,\alpha} = \{ x \mid a^\top x \leq \alpha \}$$

For homogeneous hyperplanes and halfspaces, the constant is zero so we simply write $\overline{\mathcal{H}}_a$ and $\mathcal{H}_a$.

$$\overline{\mathcal{H}}_a = \{ x \mid a^\top x = 0 \}$$
$$\mathcal{H}_a = \{ x \mid a^\top x \geq 0 \}$$

Note the different sign for the halfspace between $\mathcal{H}_{a,\alpha}$ and $\mathcal{H}_a$.

The intersection of finitely many halfspaces is called a *polyhedral convex set* or a *polyhedron.* The convex hull of finitely many points is called a *polytope.*

Theorem 1.3.1 (Weyl-Minkowski)**.** Every polytope is a polyhedron. Every bounded polyhedron is a polytope.

1.3.1 $\mathcal{H}$-representation and $\mathcal{V}$-representation

A halfspace can be seen as a point in the polar set.

Proposition 1.3.1. Consider a halfspace $\mathcal{H}_{a,\alpha}$ with $\alpha > 0$ and a closed convex set $\mathcal{S}$ containing the origin. The inclusion $\mathcal{S} \subseteq \mathcal{H}_{a,\alpha}$ is equivalent to $a/\alpha \in \mathcal{S}^\circ$.

Theorem 1.3.2 (Weyl-Minkowski (continued))**.** For a subset $\mathcal{P}$ of $\mathbb{R}^n$, the following statements are equivalent

- $\mathcal{P}$ has an *$\mathcal{H}$-representation*, that is, there are s vectors $a_i \in \mathbb{R}^n$ and s scalars $\alpha_i \in \mathbb{R}$ such that

$$\mathcal{P} = \{ x \mid a_i^\top x \leq \alpha_i, \forall i \} = \bigcap_i \mathcal{H}_{a_i,\alpha_i}. \tag{1.7}$$

- $\mathcal{P}$ has a *$\mathcal{V}$-representation*, that is, there is finitely many vectors $v_1, \ldots, v_s \in \mathbb{R}^n$, $r_1, \ldots, r_t \in \mathbb{R}^n$ such that

$$\mathcal{P} = \operatorname{conv}\{v_1, \ldots, v_s\} + \operatorname{cone}\{r_1, \ldots, r_t\}. \tag{1.8}$$

  Or equivalently, there exists matrices $V \in \mathbb{R}^{n\times s}, R \in \mathbb{R}^{n\times t}$ such that

$$\mathcal{P} = \{ x \in \mathbb{R}^n \mid \exists \lambda \in \Delta^{s-1}, \mu \geq 0, x = V\lambda + R\mu \}. \tag{1.9}$$

Definition 1.3.1. The set of *H-representations* of a polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$, denoted $\mathcal{H}_{\text{rep}}(\mathcal{P})$ is the family of sets of pairs $(a_i, \alpha_i)_{i=1}^s \in \mathbb{R}^{n+1}$ such that (1.7) holds.

Definition 1.3.2. The set of *homogeneous H-representations* of a polyhedral cone $\mathcal{P} \subseteq \mathbb{R}^n$, denoted $\mathcal{H}_{\text{rep}}(\mathcal{P})$ is the family of sets $(a_i)_{i=1}^s \in \mathbb{R}^n$ such that

$$\mathcal{P} = \{ x \mid a_i^\top x \geq 0, \forall i \} = \bigcap_i \mathcal{H}_{a_i}.$$

The V-representation of a polytope containing the origin in its interior can directly be obtained from the H-representation.

Proposition 1.3.2. Consider a polytope $\mathcal{P}$ with H-representation given by $(a_i, \alpha_i)_{i=1}^s \in \mathcal{H}_{\text{rep}}(\mathcal{P})$ If $\mathcal{P}$ contains the origin in its interior then $(a_i/\alpha_i)_{i=1}^s \in \mathcal{V}_{\text{rep}}(\mathcal{P}^\circ)$.

As a consequence, we can complete Proposition 1.2.3 which only gave the polar of the p-norm for $1 < p < \infty$.

Corollary 1.3.1. The polar of the unit ball of the 1-norm is the unit ball of the ∞-norm.

The H-representation of polyhedron gives the gauge function.

Proposition 1.3.3. Consider a polyhedron $\mathcal{P}$ with H-representation given by $(a_i, \alpha_i)_{i=1}^s \in \mathcal{H}_{\text{rep}}(\mathcal{P})$. If $\mathcal{P}$ contains the origin in its interior then

$$\mathrm{g}(\mathcal{P}, x) = \max_{i=1}^{s} \langle a_i/\alpha_i, x\rangle.$$

Proof. As the origin is in the interior of $\mathcal{P}$, $\alpha_i > 0$ for $i = 1, \ldots, s$. Therefore, $\mathrm{g}(\mathcal{H}_{a_i,\alpha_i}, x) = \langle a_i/\alpha_i, x\rangle$. By Theorem 1.3.2, $\mathcal{P}$ is the intersection of the hyperplanes, see (1.7). The result follows from Proposition 1.2.7. □

As a consequence, the following propositions follow from Proposition 1.2.7 and Proposition 1.2.19.

Proposition 1.3.4. Consider polyhedra $\mathcal{P}_1, \mathcal{P}_2$, the concatenation of a H-representation for each set is a H-representation for their intersection $\mathcal{P}_1 \cap \mathcal{P}_2$.

Proposition 1.3.5. Consider a polyhedron $\mathcal{P}$ with H-representation given by $(a_i, \alpha_i)_{i=1}^s \in \mathcal{H}_{\text{rep}}(\mathcal{P})$ and a matrix $A \in \mathbb{R}^{n\times m}$. A H-representation of the preimage of $A^{-1}\mathcal{P}$ is given by

$$(A^\top a_i, \alpha_i)_{i=1}^s \in \mathcal{H}_{\text{rep}}(A^{-1}\mathcal{P}).$$

Definition 1.3.3. The set of *V-representations* of a polyhedral cone $\mathcal{P} \subseteq \mathbb{R}^n$, denoted $\mathcal{H}_{\text{rep}}(\mathcal{P})$ is the family of sets of vectors $v_1, \ldots, v_s \in \mathbb{R}^n$, $r_1, \ldots, r_t \in \mathbb{R}^n$ such that (1.8) holds.

By Proposition 1.2.1 and Proposition 1.3.1, the following proposition follows from Proposition 1.3.3.

Proposition 1.3.6. Consider a nonempty polyhedron $\mathcal{P}$ with V-representation given by $v_1, \ldots, v_s \in \mathbb{R}^n$, $r_1, \ldots, r_t \in \mathbb{R}^n$. The support function of $\mathcal{P}$ is given by

$$\delta^*(x|\mathcal{P}) \begin{cases} \infty & \text{if } \exists i \in [t] : \langle r_i, x\rangle > 0, \\ \max_{i=1}^s \langle v_i, x\rangle & \text{otherwise.} \end{cases}$$

As a consequence of Proposition 1.3.6, the following propositions follow from Proposition 1.2.12 and Proposition 1.2.20.

Proposition 1.3.7. Consider polyhedra $\mathcal{P}_1, \mathcal{P}_2$, with V-representations $v_1^k, \ldots, v_{s_k}^k \in \mathbb{R}^n$, $r_1^k, \ldots, r_{t_k}^k \in \mathbb{R}^n$ for $\mathcal{P}_k$. The pairwise sum of the vertices $v_i^1 + v_j^2$ for $i \in s_1, j \in s_2$ and concatenation of the rays form a V-representation is a V-representation for the Minkowski sum $\mathcal{P}_1 + \mathcal{P}_2$.

Proof. It follows from Proposition 1.2.12, noticing that

$$\max_{i=1}^{s_1}\langle v_i^1, x\rangle + \max_{i=1}^{s_2}\langle v_i^2, x\rangle = \max_{i=1}^{s_1}\max_{j=1}^{s_2}\langle v_i^1 + v_j^2, x\rangle.$$

□

The quadratic increase of the number of vertices of the V-representation is resolved by the Z-representation used by Zonotopes, see Section 1.3.4.

Proposition 1.3.8. Consider a polyhedron $\mathcal{P}$ with V-representation given by $((v_i)_{i=1}^s, (r_i)_{i=1}^t) \in \mathcal{V}_{\text{rep}}(\mathcal{P})$ and a matrix $A \in \mathbb{R}^{m\times n}$. A V-representation of the image $A\mathcal{P}$ is given by

$$((Av_i)_{i=1}^s, (Ar_i)_{i=1}^t) \in \mathcal{V}_{\text{rep}}(A\mathcal{P}).$$

Similarly to Corollary 1.2.1, we have the following corollary.

Corollary 1.3.2. Consider a polyhedron $\mathcal{P}$ with V-representation given by $((v_i)_{i=1}^s, (r_i)_{i=1}^t) \in \mathcal{V}_{\text{rep}}(\mathcal{P})$ and a subset $I \subseteq [n]$. A V-representation of the projection of $\mathcal{P}$ on I is given by

$$((\pi_{I,n}v_i)_{i=1}^s, (\pi_{I,n}r_i)_{i=1}^t) \in \mathcal{V}_{\text{rep}}(\pi_{I,n}(\mathcal{P})).$$

For a polyhedral cone, the representations can be simplified:

- the $\mathcal{H}$-representation is

$$\mathcal{P} = \{\, x \in \mathbb{R}^n \mid Ax \geq 0 \,\}. \tag{1.10}$$

- and the *$\mathcal{R}$-representation* is

$$\mathcal{P} = \{\, x \in \mathbb{R}^n \mid \exists \lambda \geq 0, x = R\lambda \,\}. \tag{1.11}$$

Every convex set $C \in \mathbb{R}^n$ can be regarded as the cross-section of a convex cone $\mathcal{K} \in \mathbb{R}^{n+1}$. Indeed, let $\mathcal{K} = \text{cone}\{\, (1, x) \mid x \in C \,\}$, the set C is the intersection of $\mathcal{K}$ with the hyperplane $\{\, (\lambda, x) \mid \lambda = 1 \,\}$.

Let A^c, R^c be the matrices A and R of the $\mathcal{H}$-representation and $\mathcal{R}$-representation of $\mathcal{K}$. We have

$$\begin{aligned} A^c &= \begin{bmatrix} -b & A \end{bmatrix} \\ R^c &= \begin{bmatrix} \mathbf{1}^\top & 0 \\ V & R \end{bmatrix} \\ b &= -A^c_{:,1} \\ A &= A^c_{:,2:}. \end{aligned}$$

To extract V and R from R^c we need to scale the columns of R_c by nonnegative scalars so that the first row contains only zeros and ones (we drop the columns containing negative numbers in the first row). Then the columns with one in the first row are put in V and the columns with zeros in the first row are put in R.

The pair (A, R) is called a Double Description (DD)-pair. The DD-pair of the dual cone is $(R^\top, A^\top)$.

$$\mathcal{P}^* = \{ x \in \mathbb{R}^n \mid R^\top x \geq 0 \} \tag{1.12}$$

$$\mathcal{P}^* = \{ x \in \mathbb{R}^n \mid \exists \lambda \geq 0, x = A^\top \lambda \}. \tag{1.13}$$

This shows that going from A to R, which is called the *vertex enumeration* problem, or going from R to A, which is called the *convex hull* problem, are essentially the same problems: the *representation conversion* problem.

One way to compute A from R is to eliminate λ from the linear system of (1.11). To compute R from A we can do the same to (1.13). Eliminating variables from a polyhedron is called *polyhedral projection*. That is, we have just provided a polynomial reduction of representation conversion to polyhedral projection of $\mathcal{H}$-representation.

The family of polyhedra has the important property of being closed under projection:

Theorem 1.3.3 (Fourier-Motzkin)**.** The projection of a polyhedral set on a linear subspace is a polyhedral set.

This is proved constructively by exhibiting the Fourier-Motzkin elimination which, given the H-representation of a polyhedron, gives a procedure to compute the H-representation of the polyhedron obtained by eliminating one coordinate. Even if there exists heuristics to limit this, the Fourier-Motzkin elimination procedure generally generates a redundant H-representation. Removing these redundant halfspaces is crucial when eliminating several coordinates to avoid a prohibitive number of redundant halfspaces.

Polyhedral projection of a polyhedron in its $\mathcal{V}$-representation is simple. As shown by Corollary 1.3.2, it boils down to dropping the eliminated coordinates of vertices and rays. We can see that polyhedral projection of an $\mathcal{H}$-representation can be achieved by two representation conversion and the polyhedral projection of $\mathcal{V}$-representation. This provides a polynomial reduction of $\mathcal{H}$-representation of polyhedral projection to representation conversion.

This shows that representation conversion is essentially "as hard" as polyhedral projection.

1.3.2 Containment and inclusion

Given a point x, its containment in a polyhedron can be checked using either of the following two properties.

Proposition 1.3.9. A point x belongs to a polyhedron $\mathcal{P}$ defined by (1.7) if and only if $\langle a_i, x\rangle \leq \alpha_i$ for $i = 1, \ldots, s$.

Proposition 1.3.10. A point x belongs to a polyhedron $\mathcal{P}$ defined by (1.9) if and only if there exists $(\lambda_1 \ldots, \lambda_s) \in \Delta^{s-1}$ and $\gamma_1, \ldots, \gamma_t \in \mathbb{R}_+$ such that

$$x = \sum_{i=1}^{s} \lambda_i v_i + \sum_{i=1}^{t} \gamma_i r_i.$$

Given a point x, its inclusion in a polyhedron can be checked using either of the following two properties.

Proposition 1.3.11. A ray $\mathrm{ray}(r)$ is included in a polyhedron $\mathcal{P}$ defined by (1.7) if and only if $\langle a_i, x\rangle \leq 0$ for $i = 1, \ldots, s$.

Proposition 1.3.12. A ray $\mathrm{ray}(r)$ is included in a polyhedron $\mathcal{P}$ defined by (1.9) if and only if there exists $\gamma_1, \ldots, \gamma_t \in \mathbb{R}_+$ such that

$$r = \sum_{i=1}^{t} \gamma_i r_i.$$

Given a halfspace $\mathcal{H}_{a,\alpha}$, the inclusion of a polyhedron in the halfspace can be checked using either of the following two properties.

Proposition 1.3.13. A polyhedron $\mathcal{P}$ defined by (1.9) is included in a halfspace $\mathcal{H}_{a,\alpha}$ if and only if $\langle a, v_i\rangle \leq \alpha$ for $i = 1, \ldots, s$ and $\langle a, r_i\rangle \leq 0$ for $i = 1, \ldots, t$.

Proposition 1.3.14. A polyhedron $\mathcal{P}$ defined by (1.7) is included in a halfspace $\mathcal{H}_{a,\alpha}$ if and only if there exists $(\lambda_1 \ldots, \lambda_s) \in \mathbb{R}_+$ such that

$$\alpha \geq \sum_{i=1}^{s} \lambda_i \alpha_i$$

and

$$a = \sum_{i=1}^{s} \lambda_i a_i.$$

The Proposition 1.3.14 can be seen as a quadratic module (see Definition 1.5.10) certificate of nonnegativity of the affine polynomial $\alpha - \langle a, x\rangle$ over the polyhedron $\mathcal{P}$.

Affine hull

In this section, we analyse the following questions: 1) Given the H-representation, how to determine the affine hull ? 2) Let the *line hull* of a polyhedron be the set of lines contained in the polyhedron. The corresponding question for the V-representation is how to determine the line hull given the V-representation.

A first important fact to highlight is that the coordinates of the hyperplanes defining the affine hull (resp. lines defining the line hull) are among the halfspaces (resp. rays) of any H-representation (resp. V-representation) of the polyhedron. It remains to determine this subset I or halfspaces or rays. One simple approach, which is not the most efficient computationally, is to check for each $i \in [s]$ (resp $i \in [t]$) whether $\mathcal{P} \subseteq \mathcal{H}_{-a_i,-\alpha_i}$ (resp $\text{ray}(-r_i) \subseteq \mathcal{P}$) with Proposition 1.3.14 (resp. Proposition 1.3.12). This approach requires the solution of one linear program per halfspace (resp. ray). In fact, it turns out that a more sophisticated approach allows to only solve a number of linear programs that is bounded by the dimension of the space.

Program 1.3.1 (Primal-dual pair of programs for detecting lines)**.**

$$\begin{array}{rr} \underset{x,z}{\text{maximize}}\ z & \underset{\lambda}{\text{minimize}}\ 0 \\ \langle r_i, x\rangle \geq z & \lambda \geq 0 \\ x \text{ free} & \sum_{i=1}^{t} r_i\lambda_i = 0 \\ z \text{ free} & \sum_{i=1}^{t} \lambda_i = 1. \end{array}$$

Proposition 1.3.15. Consider a polyhedron $\mathcal{P}$ with V-representation given by $((v_i)_{i=1}^s, (r_i)_{i=1}^t) \in \mathcal{V}_{\text{rep}}(\mathcal{P})$. Let $x^\star, z^\star, \lambda^\star$ be the optimal solution of Program 1.3.1.

The polyhedron contains a line if and only if $z^\star = 0$. Moreover, there exists a subset $I \subseteq [t]$ such that the line hull of $\mathcal{P}$ is generated by r_i for $i \in I$. Furthermore, we have the following properties:

1. If $\lambda_i^\star > 0$ then $i \in I$.
2. If $\langle r_i, x^\star\rangle > z$ then $i \notin I$.

Proof. By Proposition 1.3.12, if there is a line generated by l included in the polyhedron then there exists $\mu_i \geq 0$ and $\nu_i \geq 0$ such that $\sum_i \mu_i r_i = l$ and $\sum_i \nu_i r_i = -l$. We deduce from this that $\sum \lambda_i r_i = 0$ where $\lambda = \mu + \nu$.

Conversely, if there are $\lambda \geq 0$ such that $\sum \lambda_i r_i = 0$ then let j be such that $\lambda_j > 0$. We have $\sum_{i\neq j} \lambda_i/\lambda_j r_i = -r_j$. As both r_j and $-r_j$ are in the cone, r_j generates a line in the cone. However, this means that we now have

$\sum_{i\neq j} \lambda_i/\lambda_j r_i \equiv 0 \pmod{r_j}$ so if there is another λ_i with nonzero value, we can transform it to a line as well. Therefore, we have a line generated by r_i for each i such that $\lambda_i > 0$.

If $z^\star = 0$, as $\langle r_i, x^\star\rangle \geq 0$ for each i, any ray r in the cone is such that $\langle r, x^\star\rangle \geq 0$. Therefore, if $\langle r_i, x\rangle > z$ for some i, we know that ray$(-r_i)$ does not belong to the cone. That is, $i \notin I$. □

Program 1.3.2 (Primal-dual pair of programs for detecting hyperplanes)**.**

$$\begin{aligned} &\underset{x,z}{\text{maximize}}\ z && \underset{\lambda}{\text{minimize}} \sum \lambda_i\alpha_i \\ &\quad \langle a_i, x\rangle \geq \alpha_i + z && \lambda \geq 0 \\ &\quad x \text{ free} && \sum_{i=1}^{s} a_i\lambda_i = 0 \\ &\quad z \text{ free} && \sum_{i=1}^{s} \lambda_i = 1. \end{aligned}$$

Proposition 1.3.16. Consider a polyhedron $\mathcal{P}$ with H-representation given by $(a_i, \alpha_i)_{i=1}^s \in \mathcal{H}_{\text{rep}}(\mathcal{P})$. Let $x^\star, z^\star, \lambda^\star$ be the optimal solution of Program 1.3.1.

The polyhedron is included in a hyperplane if and only if $z^\star \leq 0$. Moreover, there exists a subset $I \subseteq [s]$ such that

$$\text{aff}(\mathcal{P}) = \cap_{i\in I} \overline{\mathcal{H}}_{a_i,\alpha_i}.$$

Furthermore, we have the following properties:

1. If $\lambda_i^\star > 0$ then $i \in I$.
2. If $\langle a_i, x^\star\rangle > \alpha_i + z$ then $i \notin I$.
3. If $z^\star < 0$, then the polyhedron is empty.

Proof. By Proposition 1.3.14, if the polyhedron is included in a hyperplane $\overline{\mathcal{H}}_{b,\beta}$ then there exists $\mu_i \geq 0$ and $\nu_i \geq 0$ such that $\sum_i \mu_i a_i = b$, $\sum_i \mu_i\alpha_i \leq \beta$, $\sum_i \nu_i a_i = -b$ and $\sum_i \mu_i\alpha_i \leq -\beta$. We deduce from this that $\sum \lambda_i a_i = 0$ and $\sum_i \lambda_i\alpha_i \leq 0$ where $\lambda = \mu + \nu$.

Conversely, if there are $\lambda \geq 0$ such that $\sum \lambda_i a_i = 0$ and $\sum \lambda\alpha_i \leq 0$ then let j be such that $\lambda_j > 0$. We have $\sum_{i\neq j} \lambda_i/\lambda_j a_i = -a_j$ and $\sum_{i\neq j} \lambda_i/\lambda_j\alpha_i \leq -\alpha_j$. As the polyhedron is included in both $\mathcal{H}_{a_j,\alpha_j}$ and $\mathcal{H}_{-a_j,-\alpha_j}$, it is included in $\overline{\mathcal{H}}_{a_j,\alpha_j}$. However, this means that we now have $\sum_{i\neq j} \lambda_i/\lambda_j(\langle a_i, x\rangle + \alpha_i) \equiv \langle 0, x\rangle + \beta \pmod{\langle a_j, x\rangle + \alpha_j}$ where $\beta \leq 0$ so if there is another λ_i with nonzero value, we can transform it to a hyperplane as well. Therefore, we have a hyperplane $\overline{\mathcal{H}}_{a_i,\alpha_i}$ for each i such that $\lambda_i > 0$.

If $z^\star = 0$, as $\langle a_i, x^\star \rangle \geq \alpha_i$ for each i, any halfspace $\mathcal{H}_{b,\beta}$ that the polyhedron is included in is such that $\langle b, x^\star \rangle \geq \beta$. Therefore, if $\langle a_i, x \rangle > \alpha_i$ for some i, we know that $\mathcal{H}_{-a_i,-\alpha_i}$ does not belong to the cone. That is, $i \notin I$. □

Remark 1.3.1. In the latest version of cddlib [Fuk03] at the time of writing (v0.94j), `dd_MatrixCanonicalizeLinearity` calls `dd_ImplicitLinearityRows` and `dd_ImplicitLinearityRows` solves the Program 1.3.1 for V-representation and Program 1.3.2 for H-representation. Then, using Proposition 1.3.15.2 (resp. Proposition 1.3.16.2), some rays (resp. halfspaces) are determined not to correspond to a linearity. For all other ray r_i (resp. halfspace $\mathcal{H}_{a_i,\alpha_i}$), it calls `dd_ImplicitLinearity` which solves an LP to determine whether $-r_i$ is included in the cone with Proposition 1.3.12 (resp. the polyhedron is included in $-\mathcal{H}_{a_i,-\alpha_i}$ with Proposition 1.3.14). The number of linear programs to solve in the worstcase is equal to the number of rays (resp. halfspaces).

By comparison, Program 1.3.1 (resp. Program 1.3.2) allows to find all implicit linearities with a number of linear program equal to the dimension of the space in the worst case. This is significantly more efficient as the number of rays (resp. halfspaces) can be exponential in terms of the dimension of the space. Indeed, everytime Program 1.3.1 (resp. Program 1.3.2) is solved, either no now linearity is to be detected or, Proposition 1.3.15.1/Proposition 1.3.16.1 allows to increase the dimension of the line hull (resp. affine hull) by at least one. As the dimension is upper bounded by the dimension of the space, the number of LPs to solve is also upper bounded by the dimension of the space. This is the procedure implemented in `Polyhedra.jl`.

1.3.3 Output-sensitive computational complexity

The complexity of algorithms is commonly described using the only input size (in bits) and not the output size. This is not appropriate for representation conversion and polyhedral projection algorithms that need to list the facets (resp. vertices, rays) of a polyhedron. Indeed, the output size itself might not be polynomial in terms of the input size. We say that an algorithm is *output-sensitive* if it is in P-TIME in both the input and output size. It was shown that none of the known algorithms for the representation conversion are output-sensitive [ABS95] but it is still unknown whether there exists an output-sensitive algorithm for this problem.

An algorithm with a large output might produce its output incrementally without needing to store its output to continue. Therefore it might run in P-SPACE in its input size even if its output size is exponential in its input size. Such algorithm is said to be *compact* [Fuk96]. This is the case of the reversed search vertex enumeration algorithm implemented in LRS [Avi00].

1.3.4 Zonotopes

We saw in Proposition 1.3.7 that taking the minkowski sum of two polytopes increase the size of the V-representation quadratically (even if some vertices may be redundant). One way to go around this issue is to define the following representation.

Definition 1.3.4. A set of vectors $c, (v_i)_{i=1}^z$ is a *Z-representation* of a polyhedron $\mathcal{P}$, denoted $(c, (v_i)_{i=1}^z) \in \mathcal{Z}_{\text{rep}}(\mathcal{P})$, if

$$\mathcal{P} = \{c\} + \sum_{i=1}^{z} \text{conv}(-v_i, v_i).$$

Note that not all polytopes have a Z-representation. The polytopes with a Z-representation are called Zonotopes.

Proposition 1.3.17. Consider a nonempty zonotope $\mathcal{P}$ with Z-representation given by $(c, (v_i)_{i=1}^z) \in \mathcal{Z}_{\text{rep}}(\mathcal{P})$. The support function of $\mathcal{P}$ is given by

$$\delta^*(x|\mathcal{P}) \langle c, x \rangle + \sum_{i=1}^{z} |\langle z_i, x \rangle|.$$

As a consequence, the following propositions follow from Proposition 1.2.12 and Proposition 1.2.20.

Proposition 1.3.18. Consider zonotopes $\mathcal{P}_1, \mathcal{P}_2$, with Z-representations $(c^k, (v_i^k)_{i=1}^z) \in \mathcal{Z}_{\text{rep}}(\mathcal{P}_k)$. The tuple made of $c^1 + c^2$ and the concatenation of v^1 and v^2 is a Z-representation for the Minkowski sum $\mathcal{P}_1 + \mathcal{P}_2$.

Proposition 1.3.19. Consider a zonotope $\mathcal{P}$ with Z-representation given by $(c, (v_i)_{i=1}^z) \in \mathcal{Z}_{\text{rep}}(\mathcal{P})$ and a matrix $A \in \mathbb{R}^{m \times n}$. A Z-representation of the image $A\mathcal{P}$ is given by

$$(Ac, (Av_i)_{i=1}^z) \in \mathcal{Z}_{\text{rep}}(A\mathcal{P}).$$

We see with these proposition that the zonotope family is closed under Minkowski sum and image (hence also projection). However, it is not closed under intersection or convex hull (otherwise any polytope would be a zonotope).

1.4 Ellipsoids

An *ellipsoid* is defined by

$$\mathcal{E}_P = \{ x \in \mathbb{R}^n \mid x^\top P x \leq 1 \}. \tag{1.14}$$

where P is a positive definite matrix. When P is only positive semidefinite, the set is a *semi-ellipsoid*.

The gauge function and gauge-like function of degree 2 of the ellipsoid are

$$\mathrm{g}(\mathcal{E}_P, x) = \sqrt{x^\top P x} \qquad \mathrm{g}_2(\mathcal{E}_P, x) = x^\top P x/2. \tag{1.15}$$

To obtain the polar of $\mathcal{E}_P$, we first develop the following property of the conjugate of quadratic forms.

Proposition 1.4.1. Given a positive definite matrix $P \in \mathcal{S}^n_{++}$, the conjugate of the convex function $f(x) = x^\top P x/2$ is the convex function $f(y) = y^\top P^{-1} y/2$.

Proof. By Corollary 1.2.2, $y = Px$ if and only if $x = \nabla f^*(y)$. Therefore, $\nabla f^*(y) = P^{-1}y$ and $f^*(y) = y^\top P^{-1} y/2$. □

We can deduce the following.

Proposition 1.4.2. Given a positive definite matrix $P \in \mathcal{S}^n_{++}$, the polar of the ellipsoid $\mathcal{E}_P$ is $\mathcal{E}_P^\circ = \mathcal{E}_{P^{-1}}$.

Proof. Using (1.15), Proposition 1.2.2 and Proposition 1.4.1, we see that the polar of the gauge function is $\mathrm{g}(\mathcal{E}_P^\circ, x) = \mathrm{g}^\circ(\mathcal{E}_P, x) = \sqrt{x^\top P^{-1} x}$. □

We deduce the following from Proposition 1.4.2 and Proposition 1.2.1.

Proposition 1.4.3. Given a positive definite matrix $P \in \mathcal{S}^n_{++}$, the support function of the ellipsoid $\mathcal{E}_P$ is $\delta^*(y|\mathcal{E}_P) = \sqrt{x^\top P^{-1} x}$.

1.4.1 Operations

As the ellipsoid is invariant under polarity operation, the six operations can be considered by pairs $(\cap, \operatorname{conv}\cup)$, $(+, \sharp)$ and (A, A^{-1}). The family of ellipsoids is either invariant under both operations of the pair or none.

The intersection and convex hull of the union of ellipsoid is not necessarily an ellipsoid, but it is always a *piecewise ellipsoid*, that we study in Section 1.4.5.

The sum and inverse sum of ellipsoids is not ellipsoidal in general. By Proposition 1.2.12 and Proposition 1.4.3, we have that

$$\delta^*(y|\mathcal{E}_{P_1} + \mathcal{E}_{P_2}) = \sqrt{y^\top P_1 y} + \sqrt{y^\top P_2 y}$$

which is not a quadratic form in general. The following example provides a simple counterexample of the invariant of the family of ellipsoids under sum and inverse sum.

Example 1.4.1. Consider the semi-ellipsoids $\mathcal{E}_{e_1e_1^\top}$ and $\mathcal{E}_{e_2e_2^\top}$. By Proposition 1.2.17 and (1.15), we have

$$\mathrm{g}(\mathcal{E}_{e_1e_1^\top} \sharp \mathcal{E}_{e_2e_2^\top}, x) = \sqrt{x^\top e_1e_1^\top x} + \sqrt{x^\top e_2e_2^\top x} = |x_1| + |x_2|.$$

Hence $\mathcal{E}_{e_1e_1^\top} \sharp \mathcal{E}_{e_2e_2^\top}$ is the unit ball of the 1-norm and, by Corollary 1.3.1, its polar is the ∞-norm.

We have the following property for the linear preimage of an ellipsoid.

Proposition 1.4.4. Given an ellipsoid $\mathcal{E}_P \subseteq \mathbb{R}^n$ and a matrix $A \in \mathbb{R}^{n\times m}$, the following holds:

$$A^{-1}\mathcal{E}_P = \mathcal{E}_{A^\top PA}.$$

Proof. This is a consequence of (1.15) and Proposition 1.2.19. □

In view of Proposition 1.4.3 and Proposition 1.2.20, the linear image of an ellipsoid is more naturally obtained from P^{-1} and its polar.

Proposition 1.4.5. Given an ellipsoid $\mathcal{E}_P \subseteq \mathbb{R}^n$ and a matrix $A \in \mathbb{R}^{m\times n}$, the following holds:

$$\delta^*(y|A\mathcal{E}_P) = y^\top AP^{-1}A^\top y \tag{1.16}$$

$$A\mathcal{E}_P^\circ = \mathcal{E}_{AP^{-1}A^\top}. \tag{1.17}$$

Moreover, if $AP^{-1}A^\top$ is invertible, we have

$$A\mathcal{E}_P = \mathcal{E}_{(AP^{-1}A^\top)^{-1}}. \tag{1.18}$$

Proof. The equation (1.16) follows from Proposition 1.4.3 and Proposition 1.2.20. By Proposition 1.2.1 and Proposition 1.4.3, equation (1.17) is obtained from (1.16). The set $\mathcal{E}_{AP^{-1}A^\top}$ is a semi-ellipsoid but it is an ellipsoid only if $AP^{-1}A^\top$ is invertible. If it is an ellipsoid, (1.18) follows from Proposition 1.4.2 and (1.17). □

1.4.2 Boundary points

The following follows from (1.15) and Proposition 1.2.22.

Proposition 1.4.6. Given a positive definite matrix $P \in \mathcal{S}^n_{++}$, the normal cone of a vector $x \in \partial\mathcal{E}_P$ is given by $N_{\mathcal{E}_P}(x) = \mathrm{ray}(\{Px\})$.

The following follows from Proposition 1.4.3 and Proposition 1.2.24.

Proposition 1.4.7. Given a positive definite matrix $P \in \mathcal{S}^n_{++}$, the exposed face of a vector y is given by $F_{\mathcal{E}_P}(y) = \{P^{-1}y\}$.

1.4.3 Containment and inclusion

As a consequence of Proposition 1.2.5 and (1.15), the inclusion between ellipsoids is verified using the following property.

Proposition 1.4.8. Consider two ellipsoids $\mathcal{E}_{P_1}, \mathcal{E}_{P_2}$. The inclusion $\mathcal{E}_{P_1} \subseteq \mathcal{E}_{P_2}$ is equivalent to $P_1 \succeq P_2$.

By definition of the gauge function and Proposition 1.1.4, we have the following two conditions for the containment of a point in an ellipsoid.

Proposition 1.4.9. Given a point $x \in \mathbb{R}^n$ and an ellipsoid $\mathcal{E}_P$, the membership $x \in \mathcal{E}_P$ is equivalent to the following three equivalent conditions

$$x^\top P x \leq 1, \tag{1.19}$$

$$\begin{bmatrix} 1 & x^\top \\ x & P^{-1} \end{bmatrix} \succeq 0, \tag{1.20}$$

$$P^{-1} \succeq x x^\top. \tag{1.21}$$

Proof. The condition (1.19) is by definition of the ellipsoid, see (1.14). The condition (1.20) follows from (1.19) and Proposition 1.1.4. The condition (1.21) follows from (1.20) and Proposition 1.1.4 but we provide here an alternative proof that gives a more geometric interpretation. As $\mathcal{E}_P$ is a symmetric convex set, $x \in \mathcal{E}_P$ if and only if $\operatorname{conv}(\{-x, x\}) \subseteq \mathcal{E}_P$. By Proposition 1.3.6, we have $\delta^*(y|\operatorname{conv}(\{-x, x\})) = |\langle x, y\rangle| = \sqrt{y^\top x x^\top y}$. By Proposition 1.4.3, $\delta^*(y|\mathcal{E}_P) = \sqrt{x^\top P^{-1} x}$, Therefore, by Proposition 1.2.6, we have (1.21). □

By Proposition 1.4.2 and Proposition 1.1.4, we have the following two conditions for the inclusion of an ellipsoid in a halfspace.

Proposition 1.4.10. Given a halfspace $\mathcal{H}_{a,\alpha} \subseteq \mathbb{R}^n$ with $\alpha > 0$ and an ellipsoid $\mathcal{E}_P$, the inclusion $\mathcal{E}_P \subseteq \mathcal{H}_{a,\alpha}$ is equivalent to the following three equivalent conditions

$$a^\top P^{-1} a \leq \alpha^2, \tag{1.22}$$

$$\begin{bmatrix} \alpha^2 & a^\top \\ a & P \end{bmatrix} \succeq 0, \tag{1.23}$$

$$P \succeq \alpha^{-2} a a^\top. \tag{1.24}$$

Proof. The condition (1.22) follows Proposition 1.3.1 and Proposition 1.4.2. The condition (1.23) follows from (1.22) and Proposition 1.1.4. The condition (1.24) follows from (1.23) and Proposition 1.1.4 but we provide here an alternative proof that gives a more geometric interpretation. As $\mathcal{E}_P$ is a symmetric convex set, $\mathcal{E}_P \subseteq \mathcal{H}_{a,\alpha}$ if and only if $\mathcal{E}_P \subseteq \mathcal{H}_{a,\alpha} \cap \mathcal{H}_{-a,\alpha}$. By Proposition 1.3.3, we have $\mathrm{g}(\mathcal{H}_{a,\alpha} \cap \mathcal{H}_{-a,\alpha}, x) = |\langle x, a/\alpha\rangle| = \sqrt{x^\top a a^\top x}/\alpha$. By Proposition 1.4.3, $\delta^*(y|\mathcal{E}_P) = \sqrt{x^\top P^{-1} x}$, Therefore, by (1.15) and Proposition 1.2.4, we have (1.24). □

1.4.4 Volume and semi-axis

An ellipsoid $\mathcal{E}_P$ can be represented as the rotation of an ellipsoid of the form $\mathcal{E}_{e_1 e_1^\top/a_1^2+\ldots+e_n e_n^\top/a_n^2}$. Geometrically, the numbers a_i represent the length of the semi-axis of the ellipsoid. The length of the semi-axes is an invariant of the ellipsoids under rotation.

The volume of an ellipsoid does not depend on the rotation and can be expressed in terms of this invariant.

Proposition 1.4.11. The volume of an ellipsoid with semi-axis of length $a_1, \ldots, a_n$ is given by

$$\frac{\pi^{n/2}}{\Gamma(n/2+1)} \prod_{i=1}^{n} a_i.$$

The numbers a_i are the inverse of square root of the eigenvalues of the matrix P, in other words, they are the square root of the eigenvalues of the inverse of P. The following two properties follow from this observation.

Proposition 1.4.12. The volume of an ellipsoid $\mathcal{E}_P$ is given by

$$\frac{\pi^{n/2}}{\Gamma(n/2+1)\sqrt{\det(P)}}.$$

Proposition 1.4.13. The sum of the squares of the length semi-axes of an ellipsoid $\mathcal{E}_P$ is given by $\operatorname{Tr} P^{-1}$.

1.4.5 Piecewise semi-ellipsoidal sets

The conservatism of ellipsoidal controlled invariant sets and the complexity of the representation of polyhedral controlled invariant sets has motivated the search for alternative templates of sets. Moreover, for some sets such as the entropic cone studied in Chapter 7, the set has a polyhedral boundary in some directions and a smooth boundary in other directions, using different representations for these different directions would allow to significantly improve out approximation ability for such sets. The template we study in this section is the family of piecewise semi-ellipsoids.

Piecewise semi-ellipsoids may either be defined as the 1-sublevel sets of piecewise quadratic forms or as sets with a piecewise semi-ellipsoidal Minkowski function.

Definition 1.4.1 (Conic partition)**.** A *conic partition* of $\mathbb{R}^n$ is a set of m polyhedral cones $(\mathcal{P}_i)_{i=1}^m$ with nonempty interior such that for all $i \neq j$, $\dim(\mathcal{P}_i \cap \mathcal{P}_j) < n$ and $\cup_{i=1}^m \mathcal{P}_i = \mathbb{R}^n$.

Given a conic partition, we use the notation

$$\mathcal{N} = \{ (i,j) \mid \dim(\mathcal{P}_i \cap \mathcal{P}_j) = n-1 \}.$$

For each $(i,j) \in \mathcal{N}$, the affine hull, of $\mathcal{P}_i \cap \mathcal{P}_j$ is a hyperplane. We denote by n_{ij} the normal of this hyperplane directed towards $\mathcal{P}_i$, i.e., such that $\mathcal{P}_i \subseteq \mathcal{H}_{n_{ij}}$.

Definition 1.4.2 (Piecewise semi-ellipsoidal sets)**.** A closed convex subset $\mathcal{S} \subseteq \mathbb{R}^n$ containing the origin is said to be *piecewise semi-ellipsoidal* if there exists a conic partition $(\mathcal{P}_i)_{i=1}^m$ and symmetric positive semidefinite matrices $Q_i \in \mathcal{S}_+^n$ for $i = 1, \dots, m$, such that

$$\begin{aligned} \mathrm{g}(\mathcal{S}, x) &= \sqrt{x^\top Q_i x} && \text{if } x \in \mathcal{P}_i, \\ x^\top Q_i x &= x^\top Q_j x, \quad \forall i,j \in \mathcal{N}, x \in \mathcal{P}_i \cap \mathcal{P}_j && (1.25) \\ n_{ij}^\top Q_i x &\geq n_{ij}^\top Q_j x, \quad \forall i,j \in \mathcal{N}, x \in \mathcal{P}_i \cap \mathcal{P}_j. && (1.26) \end{aligned}$$

Note that without (1.25) and (1.26), the 1-sublevel set of the piecewise semi-ellipsoidal function may be non-convex. The condition (1.25) ensures the continuity of the function and (1.26) ensures its convexity.

This family generalizes both ellipsoids and polyhedra. Indeed, if $m = 1$ and $\mathcal{P}_1 = \mathbb{R}^n$, we recover the family of ellipsoids and if the matrices Q_i are rank-1, we recover the family of polyhedra.

Remark 1.4.1. The intersection of ellipsoids is not necessarily a piecewise semi-ellipsoids. Indeed, consider a nonempty intersection of different ellipsoids $\mathcal{E}_{Q_1}$ and $\mathcal{E}_{Q_2}$. The partition should be defined according to the quadratic form $p(x) = x^\top P x$ where $P = Q_1 - Q_2$. In order for the partition to be polyhedral, we need the variety defined by the set of zeros of $p(x)$ to be the union of two (possibly identical) hyperplanes $\overline{\mathcal{H}}_{a_1}, \overline{\mathcal{H}}_{a_2}$. That is, we need $p(x) = \langle a_1, x\rangle\langle a_2, x\rangle$ or in other words, we need $P = (a_1 a_2^\top + a_2 a_1^\top)/2$. This holds if and only if P has rank 1 or 2. This shows that while the intersection of ellipsoids is always a piecewise semi-ellipsoid in the planar case, this does not necessarily hold in higher dimension.

Given a piecewise semi-ellipsoidal sets, its polar is also piecewise semi-ellipsoidal but the conic partition of the polar depends on the matrices Q_i. This is a consequence of Proposition 1.2.2 and the fact that the conjugate of a piecewise quadratic function is a piecewise quadratic function; see [RW98, Theorem 11.14]. The Minkowski function of the polar set has the closed form expression given by Proposition 1.4.14.

Proposition 1.4.14. Given a piecewise semi-ellipsoidal set $\mathcal{S}$, as defined in Definition 1.4.2, the polar set $\mathcal{S}^\circ$ is the piecewise semi-ellipsoidal representation with the conic partition made of the polyhedra $Q_i\mathcal{P}_i$ with matrices $Q_i^\dagger$

for $i = 1, \ldots, m$ and the polyhedral cones

$$\operatorname{conv}_{i \in I} Q_i \bigcap_{i \in I} \mathcal{P}_i \tag{1.27}$$

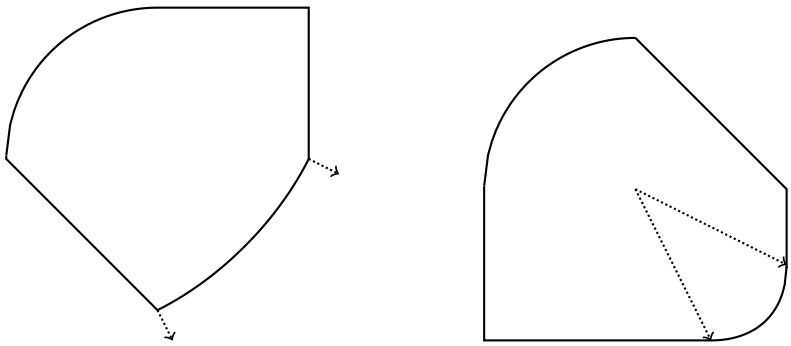
with matrices

$$E^\top (E Q_i E^\top)^\dagger E$$

where[1] $i \in I$ and $E = P_{\operatorname{aff} \cap_{i \in I} \mathcal{P}_i}$ for any subset I of $\{1, \ldots, m\}$.

Remark 1.4.2. The conic partition for the polar set created by Proposition 1.4.14 seems to contain many polyhedral cones. This seems surprising as the polar operation is an involution for closed convex sets containing the origin. In fact, many polyhedral cones of the conic partition created can be dropped without changing the set as they are not full-dimensional. For instance, the pieces of the partition created in (1.27) are only full-dimensional in case the subdifferential of $\mathrm{g}(\mathcal{S}, x)$ is not a singleton for all $x \in \cap_{i \in I} \mathcal{P}_\rangle$. That is if the inequality (1.26) is not satisfied with equality for each pair of $i, j \in I$.

Example 1.4.2 illustrates the computation of the polar of a piecewise semi-ellipsoidal set with Proposition 1.4.14.



(a) Set $\mathcal{S}$ defined by (1.28). **(b) Set $\mathcal{S}^\circ$ defined by** (1.29).

Figure 1.3: Illustration for sets $\mathcal{S}$ and $\mathcal{S}^\circ$ defined in Example 1.4.2.

Example 1.4.2. Consider the piecewise semi-ellipsoidal set defined by

$$\mathrm{g}(\mathcal{S}, x) = \begin{cases} |x| & \text{if } 0 \le y \le x, \\ |y| & \text{if } 0 \le x \le y, \\ \sqrt{x^2 + y^2} & \text{if } x \le 0 \le y, \\ |x + y| & \text{if } x, y \le 0, \\ \sqrt{x^\top Q_5 x} & \text{if } y \le 0 \le x. \end{cases} \tag{1.28}$$

[1] We have $E Q_i E^\top = E Q_j E^\top$ for any $i, j \in I$ by (1.25) so the matrix is independent on the $i \in I$ chosen.

where $x^\top Q_5 x = x^2 - xy + y^2$. The polar set is also piecewise semi-ellipsoidal:

$$g(\mathcal{S}^\circ, x) = \begin{cases} |x+y| & \text{if } 0 \le x, y, \\ \sqrt{x^2+y^2} & \text{if } x \le 0 \le y, \\ |x| & \text{if } x \le y \le 0, \\ |y| & \text{if } y \le x, 2x+y \le 0, \\ \sqrt{x^\top Q_5^{-1} x} & \text{if } 2x+y \ge 0, x+2y \le 0, \\ |x| & \text{if } x+2y \ge 0, y \ge 0. \end{cases} \tag{1.29}$$

where $x^\top Q_5^{-1} x = (4/3) \cdot (x^2 + xy + y^2)$.

Note that the partition $\mathcal{P}_i$ of $\mathcal{S}^\circ$ does not only depend on the conic partition of $\mathcal{S}$, it also depends on the value of the matrices Q_i.

For instance, the cone defined by $2x + y \ge 0, x + 2y \ge 0$ is the conic hull of the gradient of $\sqrt{x^2 - xy + y^2}$ evaluated at $(0, -1)$ and $(1, 0)$ as shown in dotted arrows in Figure 1.3. The gradients are obtained by multiplying $(0, -1)$ and $(1, 0)$ by the matrix Q_5. This illustrates why the partition of the polar is the image of the original partition under Q_5.

1.5 Semialgebraic sets

Definition 1.5.1 ([BCR13, Definition 2.1.4])**.** A *semialgebraic subset* of $\mathbb{R}^n$ is a set of the form

$$\bigcup_{i=1}^{s} \{ x \in \mathbb{R}^n \mid p_i(x) < 0, i = 1, \dots, m_j, f_i(x) = 0, i = 1, \dots, r_j \}$$

where p_i, f_i are polynomials.

Definition 1.5.2 ([BCR13, Definition 2.7.1])**.** A *basic closed semialgebraic subset* of $\mathbb{R}^n$ is a set of the form

$$\{ x \in \mathbb{R}^n \mid p_i(x) \ge 0, i = 1, \dots, m \} \tag{1.30}$$

where $p_1, \dots, p_m$ are polynomials.

Theorem 1.5.1 (Finiteness Theorem [BCR13, Theorem 2.7.2])**.** A closed semialgebraic set is a finite union of basic closed semialgebraic sets.

One sees immediately that the family of semialgebraic sets is closed under union and intersection. Similarly to Theorem 1.3.3, it is also has the remarkable property of being closed under projections.

Theorem 1.5.2 (Tarski-Seidenberg [BCR13, Theorem 2.2.1])**.** The projection of a semialgebraic set on the space of a subset of its coordinates is a semialgebraic set.

Corollary 1.5.1. The polar of a convex semialgebraic set is semialgebraic.

Proof. Let $\mathcal{S} \subseteq \mathbb{R}^n$ be a convex semialgebraic set, the set

$$\mathcal{S} \times \mathcal{S}^\circ = \{ (x, y) \in \mathbb{R}^{2n} \mid \langle x, y \rangle \leq 1, x \in \mathcal{S} \}$$

is semialgebraic as $\mathcal{S}$ is semialgebraic and $\langle x, y \rangle$ is a polynomial in x, y. As $\mathcal{S}^\circ$ is a projection of this set, it is semialgebraic by Theorem 1.5.2. □

When a basic closed semialgebraic set can be described by equalities only, it is called an *algebraic set* or *variety*.

Definition 1.5.3. A *variety* of $\mathbb{R}^n$ is a set of the form

$$\{ x \in \mathbb{R}^n \mid p_i(x) = 0, i = 1, \ldots, m \} \tag{1.31}$$

where $p_1, \ldots, p_m$ are polynomials.

1.5.1 Varieties and the Nullstellensatz

Given a variety $\mathcal{V}$ as defined in (1.31), the product of any of the polynomials p_i with any polynomial is nonnegative on I. The set of polynomials that can be certified to be zero by summing such products is called the *ideal* generated by the polynomials p_i.

Definition 1.5.4 (Ideal [CLO15, Definition 1.4.2, Lemma 1.4.3])**.** Given polynomials $p_1(x), \ldots, p_m(x)$, the *ideal* generated by the polynomials p_i is set

$$\langle p_1, \ldots, p_m \rangle \triangleq \{ q \mid q = \sum_{i=1}^{m} c_i p_i, c_i \in C[x] \}. \tag{1.32}$$

Given an ideal $I = \langle p_1, \ldots, p_m \rangle$, let $V(I)$ be the variety defined in (1.31). Given a variety $\mathcal{V}$, the set of polynomials vanishing for every point of the variety is denoted $I(\mathcal{V})$ and is an ideal.

Theorem 1.5.3 (Hilbert's Nullstellensatz [CLO15, Theorem 2])**.** Consider the ideal $I = \langle p_1, \ldots, p_m \rangle$. A polynomial $q \in I(V(I))$ if and only if $q^m \in I$ for some $m \geq 1$.

The power m can be dropped if the ideal is *radical*.

Definition 1.5.5 (Radical ideal [CLO15, Definition 2])**.** An ideal I is *radical* if $q^m \in I$ implies that $q \in I$.

Corollary 1.5.2. Consider the ideal $I = \langle p_1, \ldots, p_m \rangle$. If I is radical then a polynomial $q \in I(V(I))$ if and only if $q \in I$.

To check that a polynomial $q(x)$ belongs to an ideal I, the degree needed for the coefficients c_i in (1.32) is unknown a priori. Moreover, checking whether $q(x)$ can be reduced to zero via polynomial remainder with the polynomials $p_i(x)$ is only sufficient. However, given an ideal I, one can compute the *Gröbner basis* from the polynomials $p_1, \ldots, p_m$ with the *Buchberger's algorithm*, see [CLO15, Section 2.7].

Given a monomial ordering, let $\mathrm{LT}(p)$ denote the leading term of p. Given two polynomials in the generating basis, a polynomial with a new leading term can potentially be created by taking the polynomial obtained by canceling their leading terms with each other. This polynomial is called their *S-polynomial*.

Definition 1.5.6 (S-polynomial [CLO15, Definition 2.6.4])**.** Given nonzero polynomials p, q, let x^α be the least common multiplier of their leading terms. The *S-polynomial* of p and q is

$$\frac{x^\alpha}{\mathrm{LT}(p)} p - \frac{x^\alpha}{\mathrm{LT}(q)} q.$$

By repeatedly adding the remainder of the S-polynomial modulo the polynomials of a generating basis for each pair of polynomials in the basis if the remainder is nonzero, we converge, after finitely many additions, to a *Gröbner basis*. This is the procedure underlying the *Buchberger's algorithm*.

Definition 1.5.7 (Gröbner basis [CLO15, Definition 2.5.5])**.** The set of polynomials $\{p_1, \ldots, p_m\}$ is a *Gröbner basis* if

$$\langle \mathrm{LT}(p_1), \ldots, \mathrm{LT}(p_m) \rangle = \langle \mathrm{LT}(p) \mid p \in \langle p_1, \ldots, p_m \rangle. \tag{1.33}$$

The ideal generated by the Gröbner basis computed by the Buchberger's algorithm is I as well but a polynomial belongs to the ideal if and only if its remainder modulo the polynomials of the Gröbner basis is zero. Therefore, if I is radical then $q \in I(V(I))$ if and only if its remainder modulo the Gröbner basis is zero. This gives an algorithmic way to decide the whether $q \in I(V(I))$ for a radical ideal I. The downside of this approach is that computing Gröbner basis through the Buchberger's algorithm can be computationally intensive as the degree of the polynomials considered by the Buchberger's algorithm is not known a priori in general.

Computing the elements of zero-dimensional varieties

A second application of Gröbner basis that is relevant to this thesis is determining whether $V(I)$ is zero-dimensional and, if it is the case, determining the (finitely many) elements of $V(I)$. As shown by the following result, the variety $V(I)$ has finitely many elements if and only if $\mathbb{R}[x]/I$ is finite-dimensional.

Theorem 1.5.4 ([CLO15, Theorem 5.3.6])**.** The set $V(I)$ is a zero-dimensional if and only if the quotient ring $\mathbb{R}[x]/I$ is finite-dimensional.

By (1.33), the monomial basis b of $\mathbb{R}[x]/I$ is directly computable from the Gröbner basis. The remainder of a polynomial modulo the Gröbner basis is a polynomial of $\mathbb{R}[x]/I$ hence if $V(I)$ is zero-dimensional, it can be represented as a finite vector of the coordinates in b. Given the monomial basis of $\mathbb{R}[x]/I$, we define the multiplication matrices of I as the n matrices M_{x_i} for each variable x_i such that the jth column of M_{x_i} is the vector of coordinates of the remainder of $x_i b_j$ in b. It turns out that the multiplication matrices can be simultaneously diagonalized. That is, there exists some matrix Z such that $Z^* M_{x_i} Z$ is upper triangular, this is called the Schur decomposition of M_{x_i}. The diagonal elements of $Z^* M_{x_i} Z$ are the eigenvalues of M_{x_i} and the eigenvalues corresponding to the same column of Z form the elements of $V(I)$. There are several ways to obtain these elements of numerically using the multiplication matrices. An approach presented in [MD95] is to sample a random convex combination M of the multiplication matrices M_{x_i}. As it is a convex combination of the matrices M_{x_i}, it also admits the same eigenvectors. Let $M = ZTZ^*$ be the Schur decomposition of the matrix M. Each column q of Z provides the element $a \in V(I)$ with $a_i = q^\top M_{x_i} q$. Numerically, there is an important subtlety: A solution of multiplicity m will is given m times in the eigenvalues of M_{x_i}. However, the numerical eigenvalues in T will not be equal due to inaccuracy in the floating point representation of M and/or the computation of the Schur decomposition. Interestingly, as shown in [MD95, Section 4.1], the m eigenvalues will be around the correct eigenvalue, at a same distance which is function of the perturbations induced by numerics. Moreover, the arithmetic mean of the m eigenvalues will cancel these inaccuracies and give an accurate estimation of the correct eigenvalues. For this reason, the eigenvalues close to each other are clustered by taking their mean values and the element $a \in V(I)$ is computed as the mean of $q_j^\top M_{x_i} q_j$ for the eigenvectors q_j corresponding to each eigenvalue of the cluster.

Example 1.5.1. Consider $\mathcal{S} = \mathcal{P}_{x_1^4+x_2^4}$, the unit ball of the 4-norm. In this example, we show how to use Gröbner basis to find the face exposed by a point $y \in \mathcal{S}^\circ$ such that $2y_1 = y_2$. By Proposition 1.2.22, the face $\{x\}$ is exposed by the point $\nabla(x_1^4 + x_2^4) = (4x_2^3, 4x_2^3)$ of the $\mathcal{S}^\circ$. So the point x we are looking for is such that $x_1^4 + x_2^4 = 1$ and $4x_1^3 = 8x_2^3$. We define this variety with `SemialgebraicSets.jl` as follows:

```
using DynamicPolynomials
@polyvar x[1:2]
p = 1//1 * x[1]^4 + 1//1 * x[2]^4
∇p = differentiate(p, x)
using SemialgebraicSets
V = @set p == 1 && 2∇p[1] == ∇p[2]
```

The Gröbner basis of this variety

$$V = \langle x_1^3 - 2x_2^3, x_1x_2^3 + \frac{1}{2}x_2^4 - \frac{1}{2}, x_2^6 - \frac{4}{17}x_1^2 + \frac{2}{17}x_1x_2 - \frac{1}{17}x_2^2\rangle$$

is obtained with the Buchberger's algorithm as follows:

```
gröbnerbasis(V.I.p)
```

As the leading term of the first polynomial of the basis is x_1^3, and the leading term of the third one is x_2^6, we see that for all monomial of the monomial basis b of $\mathbb{R}[x]/I$, exponent of x_1 is at most 2 and the exponent of x_2 is at most 5. Removing all monomials that are multiple of $x_1x_2^3$, we obtain

$$b = (x_2^5, x_1^2x_2^2, x_2^4, x_1^2x_2, x_1x_2^2, x_2^3, x_1^2, x_1x_2, x_2^2, x_1, x_2, 1)$$

as follows:

```
b = monomialbasis(V.I)[2]
```

The first column of the moment matrix M_{x_1} is obtained as follows: First, we get $q(x) = -8/17x_1^2 + 16/17x_1x_2 + 2/17x_2^2$, the remainder of x_1b_2 modulo the Gröbner basis, then we get the coordinates of q in the basis b:

$$(0, 0, 0, 0, 0, 0, -8/17, 16/17, 2/17, 0, 0, 0)$$

as follows:

```
q = rem(x[1] * b[1], equalities(V))
coefficients(q, b)
```

The elements $((-0.73008, -0.91984), (0.73008, 0.91984))$ are computed as:

```
els = collect(V)
```

The polar point exposing the element in the nonnegative orthant is $(0.38914, 0.77828)$, as obtained with:

```
normal = [∇pi(x => els[2]) for ∇pi in ∇p]
polar = normal / els[2]'normal
```

This is illustrated in Figure 1.4.

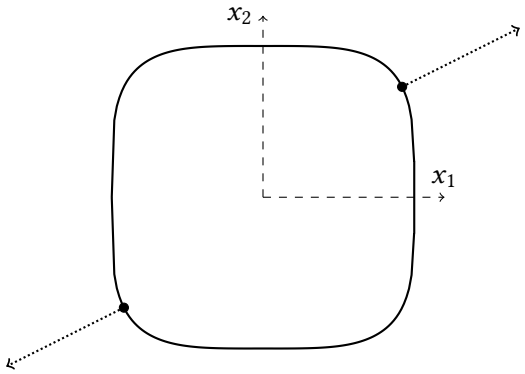


Figure 1.4: Illustration for Example 1.5.1.

We just showed how to compute the elements of a variety by computing first the Gröbner basis and then the multiplication matrices. The drawback of this method is first that the computation of the Gröbner basis can be computationally demanding and second that it is not numerically stable. The Gröbner basis can be computed in exact arithmetic but this makes the computation even more demanding and this is not an option when the coefficients of the polynomials are not known accurately as it is the case in Section 2.3.2. For linear systems, the Gröbner basis computation reduces to the Gauss elimination which is not the method used to compute the solution of linear systems because of its poor numerical performances. For linear systems, the case is simpler as we do not need to consider polynomials of higher degree because the remainder of S-polynomials is always linear. However, for algebraic systems of higher degree, the numerical computation of the remainder of the S-polynomials lead to rounding errors in floating point arithmetic.

One solution suggested in [DD09] is to somehow keep the S-polynomials obtained without taking the remainder. By stacking all the polynomials generated in a large matrix, the singular value decomposition (SVD) is then used instead of polynomial division, as the SVD is numerically stable.

A more orthogonal approach is to find the elements of the variety $V(I)$ via homotopy continuation [Li03; BT18]. This consists in defining an algebraic variety, parametrized by a parameter t, such that $V(I)$ is recovered by setting $t = 0$ and setting $t = 1$ gives a variety for which the elements are known. The approach then starts from the elements at $t = 1$ and follows the path taken

by the elements when t is varied continuously from $t = 1$ to $t = 0$. This path-following method varies t slightly and then adjust the value of the elements of the varieties for this new value of t using the Newton method, similarly to the interior-point path-following scheme mentioned in Section 2.2.

1.5.2 Basic semialgebraic sets and the Positivstellensatz

Given a basic closed semialgebraic set $\mathcal{K}$ as defined in (1.30), the product of any subset of the polynomials p_i is nonnegative on K as well as their product with a *sum-of-squares* polynomial.

Definition 1.5.8. We say that a polynomial is a *sum-of-squares* (SOS) if there exist polynomials $q_1, \ldots, q_M$ such that

$$p(x) = \sum_{k=1}^{M} q_k^2(x).$$

The set of polynomials that can be certified to be nonnegative by summing such products is called the *preorder*.

Definition 1.5.9 (Preorder)**.** Given polynomials $p_1(x), \ldots, p_m(x)$, the *preorder* generated by the polynomials p_i is set

$$\text{preorder}(p_1, \ldots, p_m) \triangleq \{\, q \mid q = \sum_{I \subseteq [m]} s_I \prod_{i \in I} p_i, s_I \in \Sigma \,\} \tag{1.34}$$

where Σ denotes the set of SOS polynomials.

Proposition 1.5.1. If a polynomial $q \in \text{preorder}(p_1, \ldots, p_m)$ then $q(x) \geq 0$ for all $x \in K$ where K is the basic closed semialgebraic set defined in (1.30).

Proposition 1.5.2. A polynomial $q \in \text{preorder}(p_1, \ldots, p_m)$ if only if there exists $q_1, q_2 \in \text{preorder}(p_1, \ldots, p_{m-1})$ such that $q = q_1 + p_m q_2$.

Theorem 1.5.5 ([BCR13, Theorem 4.4.2])**.** Consider the set

$$K = \{\, x \in \mathbb{R}^n \mid f_i(x) \geq 0, \forall i \in [m_f], g_j(x) \neq 0, \forall j \in [m_g], h_k(x) = 0, \forall k \in [m_h] \,\}.$$

The set K is empty if and only if there exists $f \in \text{preorder}(f_1, \ldots, f_{m_f}), g$ in the monoid generated by $g_1, \ldots, g_{m_g}$ (see Definition 1.1.3) and $h \in \langle h_1, \ldots, h_{m_h} \rangle$ such that $f + g^2 + h = 0$.

If $m_g = 0$ in Theorem 1.5.5, the monoid generated by $g_1, \ldots, g_{m_g}$ is the singleton $\{1\}$ hence we have the following corollary.

Corollary 1.5.3. Consider the set

$$K = \{ x \in \mathbb{R}^n \mid f_i(x) \geq 0, \forall i \in [m_f], h_k(x) = 0, \forall k \in [m_h] \}.$$

The set K is empty if and only if there exists $f \in \text{preorder}(f_1, \ldots, f_{m_f})$, and $h \in \langle h_1, \ldots, h_{m_h} \rangle$ such that $f + h = -1$.

We have already seen the particular case of Corollary 1.5.3 for affine f_i and h_k in Proposition 1.3.16.3.

If $m_g = 1$ in Theorem 1.5.5, the monoid generated by $g_1, \ldots, g_{m_g}$ is the set of powers of g_1 hence we have the following corollary.

Corollary 1.5.4. Consider the set

$$K = \{ x \in \mathbb{R}^n \mid f_i(x) \geq 0, \forall i \in [m_f], g(x) \neq 0, h_k(x) = 0, \forall k \in [m_h] \}.$$

The set K is empty if and only if there exists $f \in \text{preorder}(f_1, \ldots, f_{m_f})$, $k \in \mathcal{N}$ and $h \in \langle h_1, \ldots, h_{m_h} \rangle$ such that $f + g^{2k} + h = 0$.

The Positivstellensatz follows from Corollary 1.5.3, Corollary 1.5.4 and Proposition 1.5.2.

Corollary 1.5.5 (Positivstellensatz [BCR13, Corollary 4.4.3])**.** Consider the set

$$K = \{ x \in \mathbb{R}^n \mid p_i(x) \geq 0, \forall i \in [m_p], \forall j \in [m_g], h_k(x) = 0, \forall k \in [m_h] \}$$

and a polynomial $q(x)$.

1. We have $q(x) \geq 0$ for all $x \in K$ if and only if there exists $k \in \mathcal{N}$, $q_1(x), q_2(x) \in \text{preorder}(p_1, \ldots, p_{m_p})$ and $h \in \langle h_1, \ldots, h_{m_h} \rangle$ such that $qq_1 = q^{2k} + q_2 + h$.
2. We have $q(x) > 0$ for all $x \in K$ if and only if there exists $q_1(x), q_2(x) \in \text{preorder}(p_1, \ldots, p_{m_p})$ and $h \in \langle h_1, \ldots, h_{m_h} \rangle$ such that $qq_1 = 1 + q_2 + h$.
3. We have $q(x) = 0$ for all $x \in K$ if and only if there exists $k \in \mathcal{N}$, $p(x) \in \text{preorder}(p_1, \ldots, p_{m_p})$ and $h \in \langle h_1, \ldots, h_{m_h} \rangle$ such that $0 = q^{2k} + p + h$.

Schmüdgen's showed in [Sch91] that if the set K is compact, then we can choose $q_1 = 1$ in the second certificate of Corollary 1.5.5 and we do not need the term 1 in the right-hand side.

Theorem 1.5.6 (Schmüdgen's certificate [Sch91, Corollary 3])**.** Consider a polynomial $q(x)$ and basic closed semialgebraic set K of the form (1.30) that is compact. We have $q(x) > 0$ for all $x \in K$ if and only if $q(x) \in \text{preorder}(p_1, \ldots, p_m)$.

A subset of the preorder is given by the quadratic module.

Definition 1.5.10 (Quadratic module)**.** Given polynomials $p_1(x), \ldots, p_m(x)$, the *qmodule* generated by the polynomials p_i is set

$$\text{qmodule}(p_1, \ldots, p_m) \triangleq \{\, q \mid q = \sum_{i \in [m]} s_i p_i, s_i \in \Sigma \,\}. \tag{1.35}$$

As the quadratic module is a subset of the preorder, the following proposition follows directly from Proposition 1.5.1.

Proposition 1.5.3. If a polynomial $q \in \text{qmodule}(p_1, \ldots, p_m)$ then $p(x) \geq 0$ for all $x \in K$ where K is the basic closed semialgebraic set defined in (1.30).

Putinar showed in [Put93] that if the compactness of the set can be shown with the qmodule certificate then the Schmüdgen's certificate presented in Theorem 1.5.6 than be simplified further.

Definition 1.5.11 (Arichmedean)**.** A quadratic module is *Arichmedean* if there exists $\rho \geq 0$ such that $\rho - x_1^2 - \ldots - x_n^2$ belongs to the quadratic module.

Theorem 1.5.7 (Putinar's certificate [Put93])**.** Consider a polynomial $q(x)$ and a basic closed semialgebraic set K of the form (1.30) such that the quadratic module $\text{qmodule}(p_1, \ldots, p_m)$ is Arichmedean. We have $q(x) > 0$ for all $x \in K$ if and only if $q(x) \in \text{qmodule}(p_1, \ldots, p_m)$.

When the polynomials $q(x)$ and $p_i(x)$ are affine, the quadratic module certificate is necessary and sufficient for the nonnegativity of $q(x)$ over the polyhedron without any assumption on boundedness. Furthermore, the sum-of-squares polynomials $s_i(x)$ of (1.35) can be chosen to be nonnegative numbers. This results in Proposition 1.3.14.

Copositivity

A typical application of the Positivstellensatz is the verification of the positivity of a quadratic polynomial over a polyhedron. The canonical instance of this problem is checking the copositivity of a symmetric matrix.

Definition 1.5.12 (Copositivity)**.** A matrix $Q \in \mathcal{S}^n$ is *copositive* if $x^\top Q x \geq 0$ for all $x \in \mathbb{R}^n_+$.

While checking copositivity of a matrix is co-NP-complete [MK87], we see in this section that a sufficient condition such as Proposition 1.5.4 can be encoded as a LMI and a necessary and sufficient condition is achieved through a hierarchy of LMIs of increasingly larger size.

A commonly used sufficient condition checks the membership of the polynomial $q(x) = x^\top Qx$ in a subset of the preorder generated by $x_1, \ldots, x_n$ when only subsets $I \subseteq [n]$ of two elements are considered in (1.34). In the matrix form, the product $x_i x_j$ corresponds to the off-diagonal entry at row i and column j of a matrix. Therefore, the following property follows directly from Proposition 1.5.1.

Proposition 1.5.4 ([BPT12, Section 3.6.1])**.** Consider a symmetric matrix $Q \in \mathcal{S}^n$. If there exists a matrix $P \in \mathcal{S}^n$ with zero diagonal entries and nonnegative off-diagonal entries such that $Q \geq P$ then Q is copositive.

This is generalized to the following proposition for an arbitrary polyhedral cone.

Proposition 1.5.5. Consider a polyhedral cone $\mathcal{P}$ with homogeneous H-representation given by $(a_i)_{i=1}^s$ and a symmetric matrix $Q \in \mathcal{S}^n$. If there exists nonnegative $\lambda_{ij} \in \mathbb{R}_+$ such that

$$Q \geq \sum_{i \neq j} \lambda_{ij}(a_i a_j^\top + a_j a_i^\top)$$

then for all $x \in \mathcal{P}$, we have $x^\top Qx \geq 0$.

Remark 1.5.1. While Proposition 1.5.4 and Proposition 1.5.5 only provide a sufficient condition, a necessary condition can be obtained using a hierarchy of semidefinite programs of increasingly larger size by increasing the degree of q_1 and q_2 in Corollary 1.5.5; see [Par00, Chapter 5] for more details.

S-procedure

In this section, we consider the case of quadratic $q(x)$ and $p_i(x)$. The *lossy S-procedure* is simply the special case of Proposition 1.5.3 for quadratic forms and scalar numbers for the sum-of-squares polynomials $s_i(x)$ of (1.35).

Proposition 1.5.6 (Lossy S-procedure)**.** Consider symmetric matrices $Q, P_1, \ldots, P_m \in \mathcal{S}^n$, the polynomials $p_i(x) = x^\top P_i x$ and the basic closed semialgebraic set defined in (1.30). If

$$Q \geq \lambda_1 P_1 + \cdots + \lambda_m P_m$$

then $x^\top Qx \geq 0$ for all $x \in K$.

When $m = 1$, under a *constraint qualification* condition, the S-procedure certificate is necessary.

Theorem 1.5.8 (Lossless S-procedure [PT07])**.** Given two symmetric matrices $Q_1, Q_2 \in \mathcal{S}^n$, the existence of a $\lambda \geq 0$ such that the matrix $\lambda Q_1 - Q_2$ is positive semidefinite is sufficient for the following proposition to hold:

for all $x \in \mathbb{R}^n$, $x^\top Q_1 x \leq 0 \Rightarrow x^\top Q_2 x \leq 0$.

Moreover, if there exists $x \in \mathbb{R}^n$ such that $x^\top Q_1 x > 0$ then this condition is also necessary.

In case Q_1, Q_2 are of a specific form, the λ of Proposition 1.5.8 can be chosen to be 1. This is the case for instance for Proposition 1.4.8.

1.5.3 Polysets

We consider in this section the sets that are the sublevel set of a nonnegative *homogeneous*[2] polynomial of degree $2d$ for some positive integer d. We refer to these sets as *polysets of degree* $2d$ and denote it by

$$\mathcal{P}_p = \{ x \in \mathbb{R}^n \mid p(x) \leq 1 \}. \tag{1.36}$$

where p is a homogeneous polynomial of degree $2d$. The family of ellipsoids presented in Section 1.4 is exactly the family of polysets of degree 2. We detail in this section how to generalize the results on ellipsoids to polysets of higher degree.

The gauge function and gauge-like function of degree $2d$ of the polyset are

$$\mathrm{g}(\mathcal{P}_p, x) = \sqrt[2d]{p(x)} \qquad \mathrm{g}_{2d}(\mathcal{P}_p, x) = \frac{p(x)}{2d}. \tag{1.37}$$

As a consequence of Proposition 1.2.5 and (1.37), the inclusion between polysets is verified using the following property.

Proposition 1.5.7. Consider two polysets $\mathcal{P}_{p_1}, \mathcal{P}_{p_2}$ of degree $2d$. The inclusion $\mathcal{P}_{p_1} \subseteq \mathcal{P}_{p_2}$ is equivalent to $p_1(x) \geq p_2(x)$ for all $x \in \mathbb{R}^n$.

Well-known examples of polysets are the $(2d)$-unit balls. The gauge function of the $(2d)$-unit ball is the L^{2d} norm. The L^p norm is convex for any $p \geq 1$ and by Proposition 1.2.3 its polar is the q-unit ball for $q \geq 1$ such that $1/p + 1/q = 1$. Hence the $(2d)$-unit ball is convex for any positive integer d and its polar is the $(2d)/(2d-1)$-unit ball. This shows that, except for $d = 1$, the polar of a polyset is not necessarily a polyset, even if it is always a semialgebraic set by Corollary 1.5.1.

1.5.4 Piecewise polysets

Similarly to the piecewise semi-ellipsoidal sets introduced in Section 1.4.5, we define in this section the family of *piecewise polysets*.

[2] A polynomial is homogeneous if all its monomials have the same total degree.

Definition 1.5.13 (Piecewise semi-ellipsoidal sets)**.** A closed convex set $\mathcal{S} \subseteq \mathbb{R}^n$ containing the origin is said to be a *piecewise polyset* of degree $2d$ if there exists a conic partition $(\mathcal{P}_i)_{i=1}^m$ and convex nonnegative homogeneous polynomials $p_i(x)$ of degree $2d$ for $i = 1, \dots, m$, such that

$$\begin{aligned} \mathrm{g}(\mathcal{S}, x) &= \sqrt[2d]{p_i(x)} && \text{if } x \in \mathcal{P}_i, \\ p_i(x) &= p_j(x), \quad \forall i, j \in \mathcal{N}, x \in \mathcal{P}_i \cap \mathcal{P}_j && (1.38) \\ n_{ij}^\top \nabla p_i(x) &\geq n_{ij}^\top \nabla p_j(x), \quad \forall i, j \in \mathcal{N}, x \in \mathcal{P}_i \cap \mathcal{P}_j. && (1.39) \end{aligned}$$

Note that without (1.38) and (1.39), the 1-sublevel set of the piecewise semi-ellipsoidal function may be non-convex. The condition (1.38) ensures the continuity of the function and (1.39) ensures its convexity.

This family generalizes both polysets and piecewise semi-ellipsoids. Indeed, if $m = 1$ and $\mathcal{P}_1 = \mathbb{R}^n$, we recover the family of polysets and if $d = 1$, we recover the family of piecewise semi-ellipsoids.

The piece $\mathcal{S} \cap \mathcal{P}_i$ is basic closed semialgebraic as it is the intersection of the polyset with $p_i(x)$ as gauge-like function of degree $2d$ and the set $\mathcal{S}_1$. As $\mathcal{S}$ is the union of $\mathcal{S} \cap \mathcal{P}_i$ for each i, it is a closed semialgebraic set. Therefore, its polar is semialgebraic as well. However, it is not a piecewise polyset for $d \neq 1$ as the $2d$-unit ball is a piecewise polyset but its polar, the $(2d)/(2d-1)$-unit ball, is not a piecewise polyset.

Optimization

2

2.1 Conic optimization

Set computation heavily relies on optimization algorithms for certifying membership of point in a set or the inclusion of a set in another one. We survey in this chapter the optimization results that are used in the thesis.

In conic optimization, we term the *standard conic* form:

$$\begin{aligned} \min_{x \in \mathbb{R}^N} \quad & c^\top x \\ \text{subject to:} \quad & Ax = b \\ & x \in \mathcal{K}, \end{aligned} \tag{2.1}$$

with as dual the *geometric conic* form:

$$\begin{aligned} \max_{y \in \mathbb{R}^M} \quad & b^\top y \\ \text{subject to:} \quad & c - A^\top y \in \mathcal{K}^* \\ & y \text{ free.} \end{aligned} \tag{2.2}$$

Here, c is a N-dimensional vector, A is an $M \times N$ matrix, b is an M-dimensional vector, and $\mathcal{K} \subseteq \mathbb{R}^N$ is a convex cone. The cone $\mathcal{K}$ is typically the cartesian product of severel cones.

When $\mathcal{K}$ is a cartesian product of the nonnegative orthant $\mathbb{R}^n_+$ and the nonpositive orthant $\mathbb{R}^n_-$, the conic program is called a *linear program*. The feasible set of a linear program is a polyhedron. Considering that $\mathcal{K} = \mathbb{R}^N$, for (2.1), this polyhedron is the intersection of the nonnegative orthant and the affine subspace defined by the equalities $Ax = b$. For (2.2), the H-representation of the polyhedron is $(A_{:,i}, b_i)_{i=1}^N$.

Definition 2.1.1. The *second-order cone* (or Lorenz cone or ℓ_2-norm cone) is the cone

$$\{ (t, x) \in \mathbb{R}_+ \times \mathbb{R}^n \mid t \geq \|x\|_2 \}.$$

Definition 2.1.2. The *rotated second-order cone* is the cone

$$\{ (t, u, x) \in \mathbb{R}^2_+ \times \mathbb{R}^n \mid 2tu \geq \|x\|_2^2 \}. \tag{2.3}$$

When $\mathcal{K}$ additionally contains second-order cones or rotated second-order cones, the conic program is called a *second-order cone program*. The feasible set of a second-order cone program is the intersection between semi-ellipsoids and a polyhedron.

The problems that can be solved via second-order cone programming are called *second-order cone representable*. For instance, Proposition 2.1.1 shows that the membership of a vector of decision variables in an ellipsoid is second-order cone representable.

Proposition 2.1.1 ([BN01, Section 3.3.1])**.** Given a positive semidefinite matrix $Q \in \mathcal{S}^n$ and a vector $c \in \mathbb{R}^n$, the constraint $x \in \mathcal{E}_{Q,c}$ is second-order cone representable.

Proof. Consider a Cholesky factorization $Q = L^\top L$, the inequality $(x-c)^\top Q(x-c) \leq 1$ can be rewritten as $\|L(x-c)\|_2 \leq 1$ where $\|\cdot\|_2$ is the Euclidean norm. □

2.1.1 Duality

The Lagrangian function of this pair of primal-dual programs is given by

$$L(x, y) = y^\top b + c^\top x - y^\top Ax. \tag{2.4}$$

The standard conic program (2.1) is equivalent to

$$\inf_{x \in \mathcal{K}} \sup_{y \in \mathbb{R}^M} L(x, y)$$

and the geometric conic program (2.2) is equivalent to

$$\sup_{y \in \mathbb{R}^M} \inf_{x \in \mathcal{K}} L(x, y).$$

Proposition 2.1.2 (Max-min inequality)**.** For any function $f(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, for any $y \in Y$ and any $x \in X$, we have

$$\inf_{x' \in \mathcal{X}} f(x', y) \leq \sup_{y' \in \mathcal{Y}} f(x, y').$$

and therefore

$$\sup_{y \in \mathcal{Y}} \inf_{x \in \mathcal{X}} f(x, y) \leq \inf_{x \in \mathcal{X}} \sup_{y \in \mathcal{Y}} f(x, y).$$

The *weak duality* follows from this property.

Corollary 2.1.1 (Weak duality)**.** For any feasible solution y of (2.2), its objective value is a lower bound to the optimal value of (2.1). For any feasible solution x of (2.1), its objective value is an upper bound to the optimal value of (2.2). Therefore, the optimal value of (2.2) is smaller than the optimal value of (2.1).

An ray of (2.1) is a vector $x \in \mathcal{K}$ such that $Ax = 0$. An infeasibility ray of (2.1) is a ray x of (2.1) such that $c^\top x < 0$. A ray of (2.2) is a vector y such that $-A^\top y \in \mathcal{K}^*$. An infeasibility ray of (2.2) is a ray y of (2.2) such that $b^\top y > 0$.

The name infeasibility ray comes from the following property which generalize the classical Farkas' lemma in linear programming to conic programs.

Proposition 2.1.3 (Generalized Farkas' lemma)**.** The program (2.1) has a infeasibility ray if and only if (2.2) is infeasible. The program (2.2) has a infeasibility ray if and only if (2.1) is infeasible.

If (2.1) (resp. (2.2)) has both a feasible solution and an infeasibility ray then the program is unbounded. If a program is unbounded then its dual is infeasible but a program may be infeasible even if its dual is not unbounded.

Indeed both (2.1) and (2.2) may both be infeasible. In this case, both programs have an infeasibility ray but none of them have a feasible solution.

The difference between (2.1) and (2.2) is called the *duality* gap. If both programs are infeasible, the duality gap is infinite.

A solution x is said to be strictly feasible for (2.1) if x is feasible and $x \in \text{int}(K)$. A solution y is said to be strictly feasible for (2.2) if $c - A^\top y \in \text{int}(K^*)$.

The existence of a strict feasible solution is often called the *Slater condition*. The strong duality under this *constraint qualification* is given by the following result.

Theorem 2.1.1 (Strong duality)**.** If (2.1) and (2.2) have a strictly feasible solution then the duality gap is zero.

Strong duality may also be established by alternative methods as in Lemma 5.2.10 for instance.

2.1.2 Model transformation with bridges

In this section, we introduce the concept of model transformation with *bridges*. While the concept of model transformation is not novel, the uniqueness of the bridging concept, newly introduced in [Leg+20], relies in its constraint-wise nature. The bridge for each constraint is defined independently which allows, as detailed in Section 2.1.2, to automatically tailor the model transformation to the capability of a given solver even during model construction, i.e. before the full model is known. This was implemented in the `MathOptInterface.jl` Julia package which is the solver interface used by JuMP since its version 0.19 that was released in the 15nd of March 2019. This section develops the bridge concept at the abstract mathematical level and is not specific to its Julia implementation in `MathOptInterface.jl` except Remark 2.1.1.

In general, a conic model can be formulated in the form

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^N} & c^\top x \\ \text{subject to:} & b - Ax \in C \\ & x \in \mathcal{K}, \end{array} \tag{2.5}$$

where $C, \mathcal{K}$ are convex cones. When C is $\{0\}^M$, this is the standard conic form and when $\mathcal{K} = \mathbb{R}^N$, this is the geometric conic form.

A conic solver expects the conic program to be provided in a specific class of conic program. A conic optimization interface should allow the user to formulate the program in (2.5) and perform the transformation to the form required by the solver automatically and transparently when possible. This allows the algorithm relying on the conic program solution to be independent on the actual solver used.

For instance, if a solver only supports the nonnegative orthant and one constraint in (2.5) is $b_i - A_{i,:}x \in \mathbb{R}_-$, it can be replaced by $-b_i - (-A_{i,:})x \in \mathbb{R}_+$. We refer to this transformation as a constraint bridge from the cone $\mathbb{R}_-$ to the cone $\mathbb{R}_+$. If a variable $x_i \in \mathbb{R}_-$, it can be substituted for $-y_i$ with $y_i \in \mathbb{R}_+$. We refer to this transformation as a variable bridge.

In general, given a relation

$$AS_1 = S_2 \tag{2.6}$$

between two cones S_1, S_2, we have

- a variable bridge from S_2 to S_1 that substitutes $x \in S_2$ by Ay where $y \in S_1$.
- a constraint bridge from S_1 to S_2 that replace a constraint $f(x) \in S_1$ into a constraint $Af(x) \in S_2$.

To provide a transparent transformation, the dual of the original constraint should be obtained from the dual of the bridged constraint that was transmitted to the solver. For the constraint bridge, in the original problem, there is a constraint $f(x) \in S_1$ in the primal program, a variable $u \in S_1^*$ in the dual program and the term $\langle f(x), u\rangle_1$ in the Lagrangian function (2.4). In the bridged model, there is the constraint $Af(x) \in S_2$ in the primal program, a variable $v \in S_2^*$ in the dual program and the term $\langle Af(x), v\rangle$ in the Lagrangian function (2.4).

By (2.6) and Proposition 1.2.21, we have

$$S_1^* = A^\top S_2^*. \tag{2.7}$$

hence $u = A^\top v$ is feasible for the original dual program if and only if v is feasible for the bridged dual program. Moreover, we have

$$\langle f(x), A^\top v\rangle_1 = \langle Af(x), y\rangle_2$$

so the term in the Lagrangian function (2.4) is equal as well. We conclude that the dual value u should be computed as $A^\top v$.

For the variable bridge, a similar reasoning shows that the dual v of $x \in S_2$ should be such that $A^\top v$ is the dual $u \in S_1^*$ of $y \in S_1$ computed by the solver. If A is invertible, this means that v can be computed with $A^{-\top} u$. Otherwise, there can either be no solution for the system $A^\top v = u$ or an affine subspace of solutions. Note that by (2.7), the feasibility of u in the bridged dual program implies the existence of a solution for the system $A^\top v = u$. In case the set of solutions is an affine subspace of nonzero dimension, adding the additional condition $v \in S_2^*$ might leave a unique solution.

The following examples exhibit the linear relationship between the rotated second-order cone and the second-order cone.

Example 2.1.1. Variable and constraint bridges from rotated second-order cone to second-order cone and vice versa may be obtained by noticing that $2tu = (t/\sqrt{2} + u/\sqrt{2})^2 - (t/\sqrt{2} - u/\sqrt{2})^2$ hence

$$2tu \geq \|x\|_2^2$$

is equivalent to

$$(t/\sqrt{2} + u/\sqrt{2})^2 \geq \|x\|_2^2 + (t/\sqrt{2} - u/\sqrt{2})^2.$$

See [BN01, p. 104] for more details. Therefore use the linear transformation $A \triangleq (t, u, x) \mapsto (t/\sqrt{2} + u/\sqrt{2}, t/\sqrt{2} - u/\sqrt{2}, x)$. Note that the linear transformation is a symmetric involution (i.e. it is its own transpose and its own inverse). That means that the dual solution of the bridged constraint can simply be multiplied by A to give the dual solution of the original constraint. As A is a symmetric involution, the norm of the dual solutions are preserved by the transformation.

The following lemma follow geometrically from the right angle formed at the origin by the rotated second-order cone. For a similar property for the second-order cone, see Figure 2.1 or [Faw18, Fact 1].

Lemma 2.1.1. If the two vectors $(1, s, x)$, (v, u, y) belong to the rotated second-order cone and $\langle (1, s, x), (v, u, y) \rangle = 0$, then $y = -ux$ and $v = u\|x\|_2^2/2$.

Proof. We have

$$\begin{aligned}
(1, s, x) \cdot (v, u, y) &= v + su + \langle x, y \rangle \\
&\geq v + su - \|x\|_2 \|y\|_2 \\
&\geq v + su - 2\sqrt{suv} \\
&\geq v + su - 2\frac{v + su}{2} \\
&= 0.
\end{aligned}$$

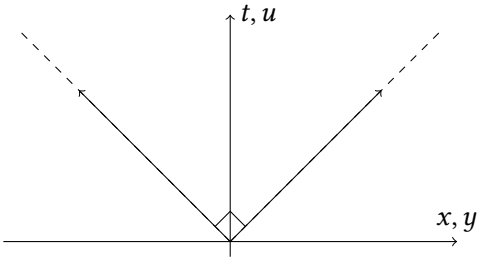


Figure 2.1: Visualization of second order cone and complementarity slackness. If $(t, x), (u, y)$ belong to the second-order cone and $(t, x) \cdot (u, y) = 0$, then (u, y) is parallel to $(t, -x)$ or (t, x) is zero or (u, y) is zero.

The first inequality uses the Cauchy-Schwarz inequality, the second one used (2.3) and the third one uses Proposition 1.1.2 with $p1$. By assumption, the left-hand side is zero, hence all inequalities are equalities. By Cauchy-Schwarz, this means that $\exists \sigma \geq 0$ such that $y = -\sigma x$. By AM-GM, we have $v = su$. By (2.3), we have either:

1. $\|y\|_2^2 < 2uv$ and $\|x\|_2^2 = 2s = 0$: That implies that $v = 0 \cdot u = 0$ hence $\|y\|_2^2 < 0$ which is impossible.
2. $\|x\|_2^2 < 2s$ and $\|y\|_2^2 = 2 \cdot u \cdot v = 0$: we have either:
   (a) $u = 0$: hence $v = s \cdot u = 0$ or
   (b) $v = 0$: since $s > 0$, $u = v/s = 0$.

   In any case, $y = 0$ and $u = v = 0$ hence the statement holds.
3. $\|x\|_2^2 = 2s$ and $\|y\|_2^2 = 2uv$: we have $\sigma^2 \|x\|_2^2 = \|y\|_2^2 = 2uv = 2u^2 s$ hence $u = \sigma$. It follows that at $v = su = \sigma \|x\|_2^2 / 2$.

□

Example 2.1.2 (Quadratic to rotated second-order cone)**.** The dual transformation is trickier to obtain for the bridge from convex quadratic constraints into rotated second-order cone constraints. Given a convex quadratic constraint:

$$\frac{1}{2} x^T Q x + a^T x + \beta \leq 0,$$

the symmetric matrix Q is positive semidefinite as the constraint is convex. Consider the Cholesky decomposition $Q = U^\top U$, the constraint is equivalent to

$$\|Ux\|_2^2 \leq 2(-a^T x - \beta)$$

which is equivalent to the membership of the vector $(1, -a^T x - \beta, Ux)$ to the rotated second-order cone.

Let $z = Ux$ and $s = -a^T x - \beta$, we can see below that the term in the Lagrangian of the bridge model is:

$$\begin{aligned}(1, s, z) \cdot (v, u, y) &= u(1, s, z) \cdot (\|z\|_2^2/2, 1, -z) \\ &= u(\|z\|_2^2/2 + s - \|z\|_2^2) \\ &= u(-\|z\|_2^2/2 + s) \\ &= u(-\|Ux\|_2^2/2 - a^\top x - \beta) \\ &= -u(x^T U^\top Ux/2 + a^\top x + \beta)\end{aligned}$$

hence with the dual $-u$ of the quadratic constraint, the value of term in the lagrangian is the same in both the original and bridged model.

Some constraint bridges create auxiliary variables u. That is, a constraint $f(x) \in \mathcal{S}_1$ is transformed into a constraint $Af(x)+Bu \in \mathcal{S}_2$ for some auxiliary variables u where $A \in \mathbb{R}^{r\times n}, B \in \mathbb{R}^{n\times m}$. Such bridges originate from the lifted representation of the set $\mathcal{S}_1$ in a lifted space with variables x and u: $x \in \mathcal{S}_1 \Leftrightarrow \exists u : Ax + Bu \in \mathcal{S}_2$. This is equivalent to

$$\mathcal{S}_1 = \pi_{[n],n+m} \begin{pmatrix} A & B \end{pmatrix}^{-1} \mathcal{S}_2$$

Using Proposition 1.2.21, this is equivalent to

$$\mathcal{S}_1^* = \pi_{[n],n+m}^{-\top} \begin{pmatrix} A^\top \\ B^\top \end{pmatrix} \mathcal{S}_2^*$$

or equivalently

$$\mathcal{S}_1^* = A^\top \mathcal{S}_2^* \qquad\qquad 0 = B^\top \mathcal{S}_2^*. \tag{2.8}$$

Example 2.1.3. An essential bridge needed for the standard conic form (2.1) is the slack bridge. This bridge transforms a constraint $f(x) \in \mathcal{S}$ into a constraint $f(x) = y$ with $y \in \mathcal{S}$ for any set $\mathcal{S} \subseteq \mathbb{R}^n$. It relies on the relation

$$x \in \mathcal{S} \Leftrightarrow \exists y \in \mathcal{S}, x = y$$

which can be rewritten into

$$\mathcal{S} = \pi_{[n],2n} \begin{pmatrix} 0 & I_n \\ I_n & -I_n \end{pmatrix}^{-1} (\mathcal{S} \times \{0\}^n).$$

Given a dual vector u of $x \in \mathcal{S}$ in the original model and dual vectors v of the constraint $y \in \mathcal{S}$ and w of the constraint $x = y$ in the bridged model, equation (2.8) gives

$$u = w \qquad\qquad v = w.$$

So u can be set to either w or v.

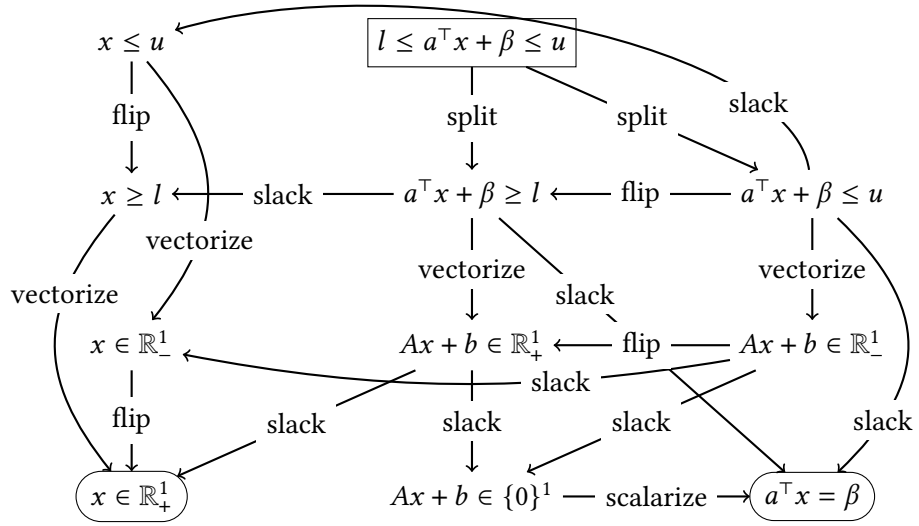


Figure 2.2: Bridging interval constraints to the linear programming standard form. The nodes in the form $x \in S$ represent constrained variables in S. The other nodes represent constraints. The interval constraint is surrounded by a sharp rectangle and the supported constrained variables and constraints are surrounded by a rectangle with rounded corners. The hyperedges are represented by multiple edges with the same label equal to the bridge name. The variable-to-constraint edges are omitted for clarity.

Automatic selection of bridges

We already introduced the variable bridges and constraint bridges. There is a third type of bridges called *objective bridges* that can be used to transform objective functions. Figure 2.3 illustrates the possible choices to bridge a quadratic objective into a form supported by the semi-definite programming standard form (2.9).

The bridges can be nested to allow multiple transformations. It is easy to see that as the number of constraint types and bridges increases, the number of different equivalent reformulations also increases, and choosing an appropriate reformulation becomes difficult. We overcome the proliferation challenge by posing the question of how to transform a constraint into supported equivalents as a shortest path problem through a hyper-graph.

We overcome the proliferation challenge by posing the question of how to transform one constrained variable, constraint or objective into supported equivalents as a shortest path problem through a graph. The nodes in the graph represent either constrained variables in set, function-in-set pairs, or objective functions. The transformation of a node into other nodes is repre-

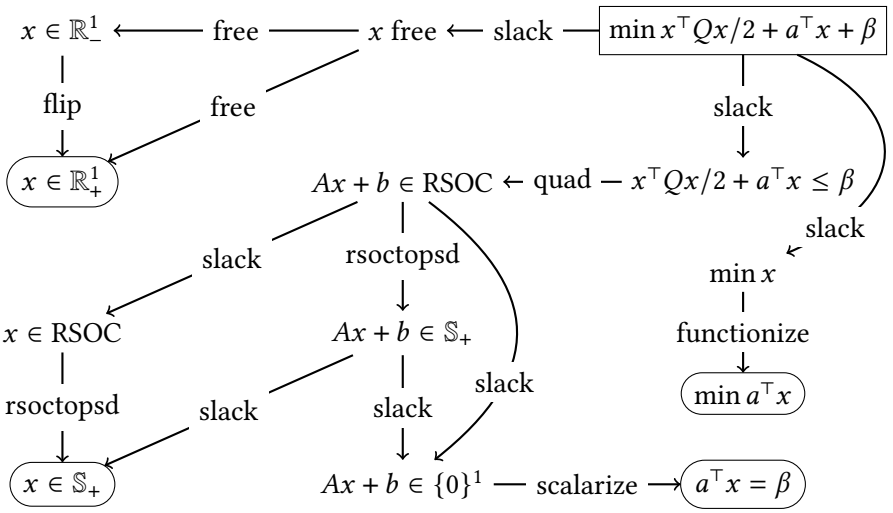


Figure 2.3: Bridging quadratic objective to the semidefinite programming standard form. The nodes in the form $x \in S$ represent constrained variables in S. The nodes in the form min $f(x)$ represent objective function of the form $f(x)$. The other nodes represent constraints. The quadratic objective function is surrounded by a sharp rectangle and the supported constrained variables, constraints and objective functions are surrounded by a rectangle with rounded corners. The hyperedges are represented by multiple edges with the same label equal to the bridge name. The variable-to-constraint edges are omitted for clarity.

sented by an edge with one source and multiple targets. The number of edges representing the transformation of a given bridge is unlimited. For instance, the slack bridge transforms any scalar $f(x) \in \mathcal{S}$ constraint into a constrained variable $s \in \mathcal{S}$ and a constraint $f(x) - s = 0$. For any scalar function type F and scalar set type S, the slack bridge creates an edge from F-in-S to the two nodes S and F-in-$\{0\}$ (assuming that the type of the function $f(x) - s$ is also F).

Remark 2.1.1. Consider that implementation of bridges in Julia contained in MathOptInterface. As the functions and sets considered is unlimited, the user can define new types in Julia even at run-time, representing the complete graph is out of the question. For this reason, the computation of the shortest path is done in a *just-in-time* fashion, so that we only compute the shortest path from a given node immediately prior to the first time a given constrained variable, function-in-set pair or objective function is added to a model.

Note the similarity with the just-in-time compilation of Julia methods. When a method is defined with abstract type, as the set of subtypes for a given abstract is not limited and new types can be added at run-time, Julia does not compile a method for any combination of concrete type but instead only compile a method for given concrete types once it is called with these types. Of course, even if the set of subtypes was limited, compiling all the possible methods would not be very practical, and the same reasoning can be applied to bridges. There is, however, an important distinction between Julia methods and bridges. When a function is called with given arguments types, the most specific method is selected through what is called *multiple dispatch*. However, when a given constraint is added with given function and set types, the bridges selected does not only depend on the function and set types but also on the solver being used through the shortest path algorithm that we present now.

Given a set of nodes N, a *single source hyperedge* (also known as *forward hyperarc*, e.g., in [Gal+93]) $e = (s, T)$ has a source node $s \in N$ and target nodes $T \subseteq N$. A *directed hypergraph* $G(N, E)$ with single source hyperedges (also known as forward hypergraph, e.g., in [Gal+93]) is represented by a set of nodes N and a set of single source hyperedges $E \subseteq N \times \mathcal{P}(N)$. The bridge graph can be represented as a directed hypergraph with single source hyperedges as follows.

There are three types of nodes in N: one variable node for each set type S representing constrained variables in S, one constraint node for each pair of function type F and set type S representing F-in-S constraints and one objective node for each type F representing objective function of type F.

For each variable node S and variable bridge that supports bridging constrained variables in S, there is a hyperedge with the variable node as source

and all the nodes corresponding to all constrained variables and constraints added by the bridge as targets. Similarly, for each constraint node F-in-S and constraint bridge that supports bridging constraints in S, there is a hyperedge with the constraint node as source and all the nodes corresponding to all constrained variables and constraints added by the bridge as targets. At last, for each objective node F and objective bridge that supports bridging objective functions of type F, there is a hyperedge with the objective node as source and all the nodes corresponding to all constrained variables, constraints added and the objective function set by the bridge as targets.

In addition to these edges, corresponding to bridges, for each variable node corresponding to a scalar (resp. vector) set S, there is an hyperedge with the variable node as the source and as the target the constraint node F-in-S where F is the type of a scalar variable (resp. the type of a vector of variables) if free variables are supported by the solvers and F is the type of a scalar affine function (resp. the type of a vector affine function) otherwise. This corresponds to constrained variables in S that are added as free variables that are then constrained with a F-in-S constraint. If free variables are not supported by the solver, they are bridged hence the constraint will be force-bridged by the functionize bridge which is the reason the function type of the target is an affine function in this case. We call these edges the variable-to-constraint edges.

Given a directed hypergraph $G(N, E)$ with single source hyperedges, a *cycle* is a sequence of nodes and arcs $(v_1, e_1, \ldots, v_n, e_n, v_1)$ such that $e_i = (v_i, T_i)$ with $v_{i+1} \in T_i$. A *hyperpath* is a directed hypergraph $G'(N', E')$ such that

1. $N' \subseteq N$, $E' \subseteq E$,
2. or each $(s, T) \in E'$, $s \in N'$ and $T \subseteq E'$ and
3. there is no cycle in G'.

Given an unsupported variable, constraint or objective, a possible transformation into supported variables, constraints and objective can be represented by a minimal hyperpath starting at the corresponding nod and ending at supported nodes. Therefore, the search for a suitable model transformation can be formulated as a search for a hyperpath. Moreover, this search can be guided by a chosen weighting function assigning a weight to hyperpaths.

The choice of weighting function has a significant impact both on the optimal hyperpath and on the computational tractability of the shortest hyperpath problem. Indeed, if the weighting function is chosen to be the number of different bridges used, the shortest path problem is NP-complete [IN89]. In the present case, if a bridge is used twice, it makes sense to include its weight twice as well so a more appropriate weight function is the number of bridges used with multiplicity. This weighting function is part of the more

general family of *additive weighting functions* for which the shortest hyperpath problem can be solved efficiently with a generalization of Bellman-Ford or Dijkstra algorithms; see [Gal+93, Section 6] for more details.

In order to accommodate with the just-in-time nature of the approach, we start with an empty hypergraph $G^*(N^*, E^*)$. When the shortest hyperpath is required from a given nodes, the subpart of the hypergraph G reachable from the node that is not yet in the hypergraph G^*. Then we compute the shortest hyperpath from all new nodes.

2.2 Semidefinite programming

For semidefinite programming, the standard conic form becomes:

$$\begin{array}{rl} \min\limits_{X \in \mathcal{S}^n} & \langle C, X \rangle \\ \text{subject to:} & \langle A_i, X \rangle = b_i, \quad i = 1, 2, \ldots, m \\ & X \succeq 0, \end{array} \tag{2.9}$$

and the geometric conic form becomes

$$\begin{array}{rl} \max\limits_{y \in \mathbb{R}^m} & \langle b, y \rangle \\ \text{subject to:} & C \succeq \sum_{i=1}^m A_i y_i \\ & y \text{ free}, \end{array} \tag{2.10}$$

where C and A_i are $n \times n$ symmetric matrices, b_i is a constant scalar, $\langle \cdot, \cdot \rangle$ denotes the inner product, and $X \succeq 0$ enforces the matrix X to be positive semidefinite.

The Lagrangian function of this pair of primal-dual program is given by

$$L(X, y) = \langle b, y \rangle + \langle C, X \rangle - \sum_{i=1}^{m} \langle y_i A_i, X \rangle. \tag{2.11}$$

Adding a slack variable Z, (2.10) is rewritten into

$$\begin{array}{rl} \max\limits_{y \in \mathbb{R}^m} & \langle b, y \rangle \\ \text{subject to:} & Z + \sum_{i=1}^M A_i y_i = C \\ & y \text{ free}, \\ & Z \succeq 0. \end{array} \tag{2.12}$$

The corresponding Lagrangian function becomes

$$L(X, y, Z) = \langle b, y \rangle + \langle Z, X \rangle. \tag{2.13}$$

The logarithmic barrier function associated to (2.11) (resp. (2.13)) is given by $L(X, y) - \mu \ln \det(X)$ (resp. $L(X, y, Z) + \mu \ln \det(Z)$). In either cases, the

KKT conditions are given by the primal and dual feasibility constraints in addition to

$$XZ = \mu I. \tag{2.14}$$

The optimal solutions $X^\star(\mu), y^\star(\mu), Z^\star(\mu)$ for different values of μ form the *central path*. In the limit $\mu \to \infty$, the optimal solution, called the *analytic center*, does not take the objective function of (2.9) or (2.10) into account. Starting from the analytic center, a path-following scheme can be used to follow the central path for decreasing t. The parameter μ is increased step by step and at each step, the value of $X^\star(\mu^+)$ for the next parameter μ^+ is approximated from the approximated value of $X^\star(\mu)$ for the previous parameter μ using the Newton method, similarly to the homotopy continuation path-following scheme described in Section 1.5.1.

More precisely, at each iteration, the current value of μ is estimated with $\langle X, Y\rangle/n$, which comes from (2.14). Then, the target value of μ for the iteration is obtained as $\sigma\langle X, Y\rangle/n$ where σ is the *centering parameter*. Directions $(\Delta X, \Delta y, \Delta Z)$ are then chosen so that $(X + \Delta X, y + \Delta y, Z + \Delta Z)$ satisfies the primal and dual feasibility constraints and satisfy a linear constraint on $\Delta X, \Delta y, \Delta Z$ based on the nonlinear equation (2.14). Many different linearizations of this equation have been explored in the literature with guarantees on the number of iterations ranging from $\sqrt{n}\ln\varepsilon$ to $n^{3/2}\ln\varepsilon$ to reduce the duality gap $\langle X, Y\rangle$ by a factor of ε^{-1}; see [MT00] for details. The complexity of each iteration is however the same for each linearization approach and is given by $O(mn^3 + m^2n^2 + m^3)$; see [MZ99]. Assuming that $m \leq n(n+1)/2$ (otherwise, the system $Ax = b$ simply has redundant rows that can be removed as a presolve step), the complexity of each iteration is simplified to $O(mn^3 + m^2n^2)$. Therefore, the overall complexity of decreasing the duality gap of a semidefinite program by a factor of ε^{-1} is given by

$$O\big((mn + m^2)n^{5/2}\log(\varepsilon)\big). \tag{2.15}$$

This primal-dual path-following method is implemented in various solvers such as CSDP [Bor99], DSDP [BYZ00], MOSEK [ApS19], SDPA [YFK03], SDPT3 [TTT03] or SeDuMi [Stu99]. Alternatively, the semidefinite program can be solved using a first-order operator-splitting method such as

- the *Newton conjugate gradient* (Newton-CG) like SDPNAL [YST15],
- the *alternating direction method of multipliers* (ADMM) like CDCS [Zhe+20], COSMO [GCG19] or SCS [ODo+16] or
- the *primal-dual hybrid gradient* (PDHG) [CP11] like ProxSDP [SGV18];

The first-order methods typically have a smaller memory footprint, as the Hessian is not computed, and faster iteration. On the other hand, second-order methods usually require fewer iterations and can provide more accurate solutions.

Remark 2.2.1. Note that when minimizing the volume of an ellipsoid $\mathcal{E}_Q$ with $Q \in \mathcal{S}^n$, the value $-\ln(\det(Q))$ should be minimized by Proposition 1.4.12. The interior-point method can be adapted by using the term $\ln\det Q$ instead of $\mu \ln\det Q$ in the objective. To the best of our knowledge, despite the apparently simple change, in practice, only SDPT3 [TTT03] have the feature of adding $\ln(\det(Q))$ in the objective function for a positive semidefinite matrix Q. For other solvers than SDPT3, the $\ln(\det(Q))$ objective can be reformulated as a $(2n) \times (2n)$ LMI and a power cone constraint or exponental cone constraints; see [BN01, p. 149]. For a solver not supporting these cones, the objective $\sqrt[n]{\det(Q)}$ should be used instead as it can be reformulated as a $(2n)\times(2n)$ LMI and rotated second-order cone constraints; see [BN01, p. 149].

Remark 2.2.2. As the number of entries in a symmetric matrix grows quadratically with its size, positive semidefinite programs performs better with many PSD variables or LMI constraints of small size than with few of large size.

Note that in the standard conic form, the solver does not support free variables. Solvers such as CDCS [Zhe+20], COSMO [GCG19], MOSEK [ApS19], SCS [ODo+16], SDPT3 [TTT03] or SeDuMi [Stu99] also support free variables but not CSDP [Bor99], DSDP [BYZ00], SDPA [YFK03], or SDPNAL [YST15]. The following bridge is used in MathOptInterface [Leg+20] for solvers not supporting free variables.

Example 2.2.1. As $\mathbb{R} = \begin{bmatrix}1 & -1\end{bmatrix} \mathbb{R}^2_+$, we can define a variable bridge from $\mathbb{R}$ to $\mathbb{R}^2_+$ by substituting the free variable x for $y - z$ where $y, z \in \mathbb{R}_+$.

Example 2.2.2. We can define variable and constraint bridges from the rotated second-order cone to the positive semidefinite cone using the linear transformation mapping the vector (t, u, x) to the matrix

$$\begin{pmatrix} t & x^\top \\ x & 2uI \end{pmatrix}.$$

Indeed by Proposition 1.1.4, the matrix is positive definite if and only if

$$\begin{aligned} uI &\succ 0 \\ t - x^\top (2uI)^{-1} x &> 0 \end{aligned}$$

which is equivalent to

$$\begin{aligned} u &> 0 \\ 2tu &> x^\top x. \end{aligned}$$

Similarly, variable and constraint bridges from the second-order cone to the positive semidefinite cone can be defined using the linear transformation mapping the vector (t, x) to the matrix

$$\begin{pmatrix} t & x^\top \\ x & tI \end{pmatrix}.$$

Note, however, that if $x \in \mathbb{R}^n$, that gives a positive semidefinite constraint of dimension $n + 1$ while combining the bridge from second-order cone to the ro second-order cone defined in Example 2.1.1 with the bridge from rotated second-order cone gives a positive semidefinite matrix of dimension n. Therefore, in view of Remark 2.2.2, the bridge from the second-order cone to the positive semidefinite cone is not recommended, it should be replaced by the combination just mentioned. For instance, it is defined in MathOptInterface [Leg+20] but is not enabled by default.

2.3 Sum-of-Squares programming

Deciding whether a multivariate polynomial of degree $2d \geq 4$ is nonnegative is known to be co-NP-hard. However a sufficient condition for a polynomial to be nonnegative is easy to check. If a polynomial is SOS, see Definition 1.5.8, then it is obviously nonnegative.

It is well known that if $p(x)$ is a homogeneous polynomial of degree $2d$ then each $q_k(x)$ must be an homogeneous polynomial of degree d; this can be shown easily using the Newton polytope of $p(x)$ and [Rez78, Theorem 1]. We can check whether a polynomial is SOS using semidefinite programming thanks to the following theorem.

Theorem 2.3.1 ([CLR95; Nes00; Par00; PL03; Sho87])**.** A multivariate polynomial $p(x)$ is a sum-of-squares if and only if

$$p(x) = b^\top Q b$$

where Q is a symmetric positive semidefinite matrix and b is a vector of polynomials.

Given a polynomial basis b, let C_b be the linear map from the coefficients of Q to the coefficient of $p(x) = b^\top Q b$. Given a vector of polynomials b of length N, the cone $C_b \mathcal{S}^n_+$ is a convex cone of sum-of-squares polynomials. It is not necessarily the cone of all sum-of-squares polynomials with these monomials though as shown in the following example:

Example 2.3.1. Any polynomial $p(x)$ such that the only terms of nonzero coefficient have monomials x^4, x^2y^2 or y^4 can be expressed as $p(x) = b^\top Q b$

where $Q \in \mathcal{S}^2$ and $b = (x^2, y^2)$. However, there does not exist any $Q \in \mathcal{S}^n_+$ such that the sum-of-squares polynomial $2x^4 + 2x^2y^2 = b^\top Q b$. This can be verified numerically as follows using `SumOfSquares.jl` and CSDP [Bor99].

```
model = Model(CSDP.Optimizer)
@variable(model, p, SOSPoly([x^2, y^2]))
@constraint(model, p == 2x^4 + 2x^2 * y^2)
optimize!(model)
dual_status(model) # gives INFEASIBILITY_CERTIFICATE
```

Definition 2.3.1 (Newton polytope)**.** Given a polynomial $p(x)$, the convex hull of the set of exponents of the monomials of terms with nonzero coefficients of $p(x)$ is called the *Newton polytope* of $p(x)$ and is denoted $\mathcal{N}(p)$.

Given the Newton polytope of a polynomial $p(x)$, the following result allows to determine a basis b that contains a sufficient set of monomials to make sure that $p(x)$ is a sum-of-squares if and only if there exists a positive semidefinite matrix Q.

Theorem 2.3.2 ([Rez78, Theorem 1])**.** If $p(x) = \sum_{k=1}^M q_k^2(x)$ then $2\mathcal{N}(q_k) \subseteq \mathcal{N}(p)$ for $k = 1, \ldots, M$.

By Theorem 2.3.2, if the set of entries of the vector b of length N is the set of integer points of a polytope $\mathcal{P}$ then $C_b\mathcal{S}^n_+$ is the set of all sum-of-squares polynomials $p(x)$ such that $\mathcal{N}(p) \subseteq C_b\mathcal{S}^n_+$. This relation between the cone of sum-of-squares polynomials and positive semidefinite matrices is of the form (2.6) hence it yields

- a variable bridge from the cone of sum-of-squares polynomials to the cone of positive semidefinite matrices.
- a constraint bridge from the cone of positive semidefinite matrices to the cone of sum-of-squares polynomials.

The variable bridge is of tremendous practical importance as it allows to solve SOS program with SDP solvers. The constraint bridge could be useful as the dimension of the PSD cone is larger than the dimension of the SOS cone when several entries of the matrix $bb^\top$ are equal. However, it is not currently used in practice as there are no SOS solvers that do not implement support for SDP constraints.

The constraint bridge from the cone of SOS polynomials to the cone of positive semidefinite matrices is obtained by combining the slack bridge of Example 2.1.3 with the variable bridge from SOS polynomials to positive semidefinite matrices.

We denote the set of homogeneous polynomials of degree $2d$ as $\mathbb{R}_{2d}[x]$.

Corollary 2.3.1. A homogeneous multivariate polynomial $p(x)$ of degree $2d$ is a sum of squares if and only if

$$p(x) = b^\top Q b$$

where Q is a symmetric positive semidefinite matrix and b is a polynomial basis of $\mathbb{R}_{\mathbf{d}}[x]$.

We denote the cone of homogeneous SOS polynomials of degree $2d$ as $\Sigma_{\mathbf{2d}}$ and the dual of $\Sigma_{\mathbf{2d}}$ as $\Sigma^*_{\mathbf{2d}}$. Note that the the dimension of $\mathbb{R}_{\mathbf{d}}$ is given by $\binom{n+d-1}{d}$. Therefore, verifying whether a homogeneous multivariate polynomial of n variables and degree $2d$ can be formulated as a semidefinite program (2.9) with a positive semidefinite variable X of size $\binom{n+d-1}{d} \times \binom{n+d-1}{d}$. Does the polynomial complexity of semidefinite programs given in (2.15) imply that Sum-of-Squares program can be solved in polynomial time ? As we have

$$\binom{n+d-1}{d} = \frac{(n+d-1)!}{d!(n-1)!} = \frac{1}{d!}\prod_{i=n}^{n+d-1} i = \frac{1}{(n-1)!}\prod_{i=d+1}^{n+d-1} i,$$

the binomial coefficient is both polynomial in n for fixed d and polynomial in d for fixed n. However, it is not a multivariate polynomial in both n and d. So the answer is affirmative if either n or d is fixed but is negative in terms of both n and d.

From the exact arithmetic viewpoint, the basis b chosen in Theorem 2.3.1 does not affect whether $p(x)$ is SOS or not. A specific choice of basis may, however, improve the numerical behavior of the corresponding semidefinite program. Moreover considering a basis satisfying some properties may facilitate the theoretical development. For instance, in Lemma 5.3.5, we would need the basis $b = x^{[d]}$ such that $\|x^{[d]}\|_2 = \|x\|_2^d$. This leads us to the choice of the *scaled monomial* basis. The elements of this basis are

$$\frac{d!}{\alpha_1!\alpha_2!\cdots\alpha_n!} x_1^{\alpha_1}\cdots x_n^{\alpha_n}.$$

for each n-tuples of nonnegative integers α such that $\alpha_1 + \cdots + \alpha_n = d$. For any matrix $A \in \mathbb{R}^{n\times n}$, the d-lift induces an associated map $A^{[d]} \in \mathbb{R}^{N_d \times N_d}$ which is the unique matrix that satisfies $(Ax)^{[d]} = A^{[d]}x^{[d]}$. We also have the following property for the standard inner product between vectors

$$\langle x^{[d]}, y^{[d]} \rangle = (\langle x, y \rangle)^d. \tag{2.16}$$

2.3.1 Sum-of-Squares Convexity

When considering polar polysets in Chapter 4, we will need to constrain a polynomial to be convex. This can be achieved by constraining its Hessian to

be pointwise positive semidefinite. The Hessian is a polynomial matrices, that is, a matrix with polynomial entries. Thus the problem of checking convexity of a polynomial reduces to the problem of checking the pointwise positive semidefiniteness of a polynomial matrix. Since polynomial nonnegativity is a special case of polynomial matrix positive semidefiniteness, the latter is also NP-hard. Hence this reduction does not provide an efficient algorithm for polynomial convexity.

It turns out that checking the convexity or quasi-convexity of a multivariate polynomial is co-NP-hard even for quartic polynomials [Ahm+13]. Thus checking whether the Hessian is positive semidefinite may actually be the best we can do. We can define the generalization of SOS polynomial for polynomial matrices as follows.

Definition 2.3.2 (SOS matrices and SOS convexity)**.** A symmetric polynomial matrix $P(x) \in \mathbb{R}[x]^{m\times m}$, $x \in \mathbb{R}^n$, is an *SOS matrix* if there exists a polynomial matrix $M(x) \in \mathbb{R}[x]^{s\times m}$ for some $s \in \mathbb{N}$, such that $P(x) = M^\top(x)M(x)$.

A polynomial $p(x)$ is *SOS-convex* if its Hessian is an SOS matrix.

Polynomial matrix semidefiniteness is actually equivalent to polynomial nonnegativity.

Lemma 2.3.1. Let $P(x) \in \mathbb{R}[s]^{m\times m}$ be a symmetric polynomial matrix, with $x \in \mathbb{R}^n$. Let $p(x, y) = y^T P(x)y$ be the associated scalar polynomial in $m + n$ variables where $y = [y_1, \ldots, y_m]$.

1. The matrix $P(x)$ is positive semidefinite if and only if $p(x, y)$ is nonnegative.

2. The matrix $P(x)$ is an SOS matrix if and only if $p(x, y)$ is SOS (in $\mathbb{R}[x, y]$).

More information on SOS matrices can be found in [BPT12, Section 3.3.2] and see [BPT12, Section 3.3.3] or [AP12a; AP13] for sos convexity. Sos convexity also plays a prominent role in certifying the stability of nonlinear switched systems [AJ13; AJ+13].

2.3.2 Moments

A common interpretation of the dual space $\mathbb{R}^*_{\mathbf{2d}}$ of linear functionals on homogeneous polynomials of degree $2d$ is the space of moments of momonials of degree $2d$; see [BPT12, Section 3.5] and [Las09]. If $p(x) = a^\top x^{[d]}$ and m is the vector of moments of $x^{[d]}$ of a measure μ then

$$\langle m, a\rangle = \int p(x)\,\mathrm{d}\mu = \langle \mu, p\rangle.$$

As an SOS polynomial is nonnegative, this integral is nonnegative for any measure μ. Therefore, given a moment vector m, a necessary condition for a measure to exist with these moments is that $\langle m, a \rangle \geq 0$ for any vector of coefficients a of an SOS polynomial. That is, $\Sigma^*_{2\mathbf{d}}$ is a superset of the set of moments of measures. The members of $\Sigma^*_{2\mathbf{d}}$ are often called *pseudo-measures* and denoted $\widetilde{\mu}$; see [Bar+12].

Given a program on measures such as Program 5.3.1, the *moment relaxation* consists in truncating the infinite moment series to the finite set of moments of the monomials in the matrix $M = bb^\top$ where b is a finite polynomial basis. The constraint that a measure exists with these moments is relaxed to a semidefinite constraint on the moment matrix, defined in Definition 2.3.3, which is in fact equivalent to requiring that the measure belongs to the cone $\Sigma^*_{2\mathbf{d}}$ introduced above.

Definition 2.3.3 ([Las09, Section 3.2.1])**.** Given a positive integer d and a vector y of moments of all the monomials up to degree $2d$, the *moment matrix* is defined as the matrix $M_d(y)$, indexed by the exponents of monomials of degree d such that $[M_d(y)]_{\alpha,\beta} = y_{\alpha+\beta}$ where $y_{\alpha+\beta}$ is the moment of the monomial $x^\alpha x^\beta$.

The following propositions shows that the membership of a point in an SOS-convex polyset is SOS-representable.

Proposition 2.3.1 ([HN10, Theorem 9])**.** Given an SOS-convex polynomial $s(x) = \sum_\alpha s_\alpha x^\alpha$ of degree $2d$, the membership of the point x to the set $\{ x \mid s(x) \leq 0 \}$ is equivalent to the existence of a vector y of moments of all the monomials up to degree $2d$ such that $y_0 = 1$, such that the moment matrix $M_d(y)$, defined in Definition 2.3.3, is positive semidefinite and $\sum_\alpha s_\alpha y_\alpha \leq 0$.

Atom extraction

Given a positive semidefinite moment matrix M, let $\mathcal{V}$ be the union of the support of any measure with the moment matrix M. Given a measure μ and a nonnegative polynomial, $\langle \mu, p \rangle = 0$ implies that p is zero over the support of μ. Therefore, if $\langle M, Q \rangle = 0$ for a positive semidefinite matrix Q, the SOS polynomial $b^\top Q b$ is zero over $\mathcal{V}$. In particular, this is true for any rank one matrix $Q = qq^\top$ hence we have $b^\top q = 0$ is zero over $\mathcal{V}$ of any for any vector q such that $q^\top M q = 0$. Let $M = UU^\top$ be the Cholesky decomposition, the variety $\mathcal{V}$ is contained in the variety $V(I)$ where I is the ideal generated by the polynomials $b^\top q$ where Q is in the kernel of U. When $V(I)$ is zero-dimensional, it means that any measure μ that has the moments M is a sum of Dirac measures.

In practice, when M is computed numerically, we use the kernel of its low-rank factorization instead [HL05]. The variety is verified to be zero-

dimensional and the elements a_i of $V(I)$ are computed as in Section 1.5.1. We can then find the weights λ_i such that the moments of $\sum_i \lambda_i \delta_{a_i}$ are M by solving a linear system of equations.

Example 2.3.2. Consider the following moment matrix of [HL05, Section 4]:

```
using DynamicPolynomials
@polyvar x y
using MultivariateMoments
μ = measure([1/9,     0,     1/9,     0, 1/9,   0,
                0,     0,       0,   1/3,   0, 1/3,
                0,     0,       1],
             [x^4, x^3*y, x^2*y^2, x*y^3, y^4, x^3,
            x^2*y, x*y^2,     y^3,   x^2, x*y, y^2,
                x,     y,       1])
v = moment_matrix(μ, [1, x, y, x^2, x*y, y^2])
```

The variety $V(I)$, computed with a specific tolerance `ranktol`. The tolerance is used to transform the moment matrix into the variety $V(I)$ give the atoms. This transformation uses the SVD decomposition of the moment matrix and discards the polynomials of the generating basis of the ideal I corresponding to a singular value lower than `ranktol`.

```
MultivariateMoments.computesupport!(v, 1e-16)
```

is $V(\langle -y^2+1/3, -x^2+1/3\rangle)$. The elements of this variety are $(\pm 1/\sqrt{3}, \pm 1/\sqrt{3})$ and the atomic measure obtained as:

```
extractatoms(v, 1e-16)
```

is

$$\frac{1}{4}\delta_{(1/\sqrt{3},1/\sqrt{3})} + \frac{1}{4}\delta_{(1/\sqrt{3},-1/\sqrt{3})} + \frac{1}{4}\delta_{(-1/\sqrt{3},1/\sqrt{3})} + \frac{1}{4}\delta_{(-1/\sqrt{3},-1/\sqrt{3})}.$$

The classical application for this atom extraction is in polynomial optimization. The minimization of a polynomial $p(x)$ over a basic semialgebraic set K can be rewritten as the maximization of γ such that $p(x) - \gamma$ is nonnegative over K. A hierarchy of sufficient conditions for this nonnegativity can be obtained with Corollary 1.5.5 by increasing the degree considered in the polynomials of the certificate at each level of the hierarchy. The dual of the constraint associated with this certificate of nonnegativity is a matrix of moments M. If this matrix is found be the moments of Dirac measures, it means that the centers of the Diracs are minimizers and the value γ is the value of

$p(x)$ at these minimizers. We provide another application of atom extraction in Section 5.3.

Example 2.3.3. Consider the polynomial optimization problem of minimizing the polynomial $x^3 - x^2 + 2xy - y^2 + y^3$ over the polyhedron defined by the inequalities $x \geq 0, y \geq 0$ and $x + y \geq 1$. The value of p at $(1, 0)$, $(1/2, 1/2)$ and $(0, 1)$ are respectively 0, $1/4$ and 0:

```
using DynamicPolynomials
@polyvar x y
p = x^3 - x^2 + 2x*y -y^2 + y^3
using SemialgebraicSets
S = @set x >= 0 && y >= 0 && x + y >= 1
p(x=>1, y=>0), p(x=>1//2, y=>1//2), p(x=>0, y=>1)
```

As we will see below with Sum-of-Square Programming, the optimal solutions are $(x, y) = (1, 0)$ and $(x, y) = (0, 1)$ with objective value 0 but Ipopt [BZ09] only finds the local minimum $(1/2, 1/2)$ with objective value $1/4$. It is not surprising that Ipopt does not find the optimal solution, as this solvers only ensures to find a locally optimal solution.

```
using JuMP
using Ipopt
model = Model(Ipopt.Optimizer)
@variable(model, a >= 0)
@variable(model, b >= 0)
@constraint(model, a + b >= 1)
@NLobjective(model, Min, a^3 - a^2 + 2a*b -
                         b^2 + b^3)
optimize!(model)

@show termination_status(model)
@show value(a)
@show value(b)
@show objective_value(model)
```

A Sum-of-Squares certificate that $p \geq \alpha$ over the domain S, ensures that α is a lower bound to the polynomial optimization problem. The following program searches for the largest upper bound and finds zero.

```
model = SOSModel(optimizer)
@variable(model, α)
@objective(model, Max, α)
@constraint(model, c3, p >= α, domain = S)

optimize!(model)
@show termination_status(model) # OPTIMAL
@show objective_value(model) # -2.0092711e-10
```

Using the solution $(1/2, 1/2)$ found by Ipopt of objective value $1/4$ and this certificate of lower bound 0 we know that the optimal objective value is in the interval $[0, 1/4]$ but we still do not know what it is (if we consider that we did not try the solutions $(1, 0)$ and $(0, 1)$ as done in the introduction). If the dual of the constraint `c3` was atomic, its atoms would have given optimal solutions of objective value 0 but that is not the case.

```
using MultivariateMoments
v3 = moment_matrix(c3)
extractatoms(v3, 1e-3) # nothing
```

When the variety $V(I)$ extracted from the moment matrix is not zero-dimensional, `extractatoms` concludes that the measure is not atomic and return `nothing`. In this case, we can verify as follows:

```
v3.support
```

that the ideal I we extract from the moment matrix is

$$\langle x + y - 1 \rangle$$

hence the dimension of $V(I)$ is 1. This explains `extractatoms` returned `nothing`.

The constraint `c3` uses Proposition 1.5.3 to certify the nonnegativity of $p - a$ over `S` with:

$$p - \alpha = s_0 + s_1 x + s_2 y + s_3(x + y - 1)$$

where s_0, s_1, s_2, s_3 are SOS. By default, the basis of the SOS polynomials s_i are chosen so that the degrees of $s_1 x$, $s_2 y$ and $s_3(x + y - 1)$ match those of $p - \alpha$ and then the basis of s_0 is selected using Theorem 2.3.2 on the polynomial

$$p - \alpha - s_1 x - s_2 y - s_3(x + y - 1).$$

The maximum total degree (i.e. maximum sum of the exponents of x and y) of the monomials of p is 3 so the constraint in the program above is equivalent to `@constraint(model, p >= α, domain = S, maxdegree = 3)`. That is, since x, y and $x+y-1$ have total degree 1, the SOS polynomials s_1, s_2 and s_3 have been chosen with maximum total degree 2. Since these polynomials are SOS, their degree must be even so the next maximum total degree to try is 4. For this reason, the keywords `maxdegree = 4` and `maxdegree = 5` have the same effect in this example. In general, if the polynomials in the domain are not all odds or all even, each value of `maxdegree` has different effect in the choice of the maximum total degree of s_i. With `maxdegree` set to 5, the lower bound found is zero again.

```
model = SOSModel(optimizer)
@variable(model, α)
@objective(model, Max, α)
@constraint(model, c5, p >= α, domain = S,
            maxdegree = 5)

optimize!(model)
@show termination_status(model) # OPTIMAL
@show objective_value(model) # 8.70000606e-8
```

This time, the dual variable is atomic as it is the moments of the measure

$$0.5\delta_{(1,0)} + 0.5\delta_{(0,1)}.$$

Therefore the program provides both a certificate that 0 is a lower bound and a certificate that it is also an upper bound since it is attained at the global minimizers $(1,0)$ and $(0,1)$.

```
using MultivariateMoments
ν5 = moment_matrix(c5)
extractatoms(ν5, 1e-3)
```

The ideal extracted from the moment matrix is

$$I = \langle x + y - 1, y^2 - y, xy, x^2 + y - 1 \rangle$$

as obtained by:

```
ν5 = moment_matrix(c5)
MultivariateMoments.computesupport!(ν5, 1e-3)
```

We can compute the Gröbner basis with:

```
SemialgebraicSets.computegröbnerbasis!(
    ideal(v5.support))
v5.support
```

which gives the Gröbner basis:

$$I = \langle x + y - 1, y^2 - y \rangle.$$

From which the solution $(1, 0)$ and $(0, 1)$ are as obtained from the multiplication matrices as discussed in Section 1.5.1.

2.4 Parametrized program

We define in this section the concept of optimality and duality cuts that is used in Section 4.2.2, Section 6.4 and Section 7.8.

Consider the function $f(p)$ defined as the optimal objective value of a conic program parametrized by a vector of parameters p.

$$\begin{array}{rll} f(p) = \min\limits_{x \in \mathbb{R}^N} & c^\top x & \\ \text{subject to:} & Ax = b - Cp & \\ & x \in \mathcal{K}, & \end{array} \tag{2.17}$$

If strong duality holds then this function can be equivalently defined with the dual program where the convexity of $f(p)$ is apparent:

$$\begin{array}{rll} f(p) = \max\limits_{x \in \mathbb{R}^N} & \langle b, y \rangle - \langle Cp, y \rangle & \\ \text{subject to:} & c - A^\top y \in \mathcal{K}^* & \\ & y \text{ free.} & \end{array} \tag{2.18}$$

For a given value of the vector of parameters p, the dual optimal solution $y^\star(p)$ provides a subgradient of f at p given by the objective function of (2.18): $\langle b, y^\star(p) \rangle - \langle Cp, y^\star(p) \rangle$ hence

$$\forall \hat{p}, \quad \forall \hat{p}, f(\hat{p}) \geq \langle b, y^\star(p) \rangle - \langle C\hat{p}, y^\star(p) \rangle. \tag{2.19}$$

Moreover, if (2.17) is infeasible, from any infeasibility ray $y^\star(p)$, by Proposition 2.1.3, we have $-C^\top y^\star(p) \in N_{\operatorname{dom} f}(p)$ hence $f(p)$ is $+\infty$ over the halfspace $\mathcal{H}_{C^\top y^\star(p), \langle b, y^\star(p) \rangle}$ and

$$\operatorname{dom}(f) \subseteq \bigcap_p \mathcal{H}_{-C^\top y^\star(p), -\langle b, y^\star(p) \rangle}. \tag{2.20}$$

Given the dual solution $y^\star(p)$ for different values of p, (2.19) and (2.20) give a polyhedral lower-approximation of the function f as well as a polyhedron outer-approximation of its domain. The affine lower approximation

given by (2.19) for each value of p is called an *optimality cuts* and the halfspace given by (2.20) for each value of p is called a *feasibility cuts*.

Starting from an initial lower approximation of $f(p)$, we can iteratively minimize this lower approximation and then refine it at the optimal solution $p^\star$ using Algorithm 1.

Algorithm 1 Cut generation for a parametrized program.

Input Value p for the vector of parameters.
Solve the pair of program (2.17)/(2.18) with value p for the vector of parameters.
if the program is infeasible **then**
  Get an infeasibility ray $y^\star(p)$ of (2.18)
  Intersect the domain of $f(p)$ with the feasibility cut given by the halfspace $\mathcal{H}_{-C^\top y^\star(p), -\langle b, y^\star(p)\rangle}$.
else
  Get an feasible solution y of (2.18)
  Intersect the graph of $f(p)$ with the optimality cut $f(p) \geq \langle b, y^\star(p)\rangle - \langle C\hat{p}, y^\star(p)\rangle$.
end if

Systems and control

3

In this chapter, we define the different types of systems considered in this thesis. In Section 3.1, we define specific classes of hybrid systems with examples used throughout the text. In Section 3.2, we review the stability of the classes of autonomous systems and in Section 3.3 we detail the stabilizability of system with a continuous control input.

A system is defined either in continuous or discrete time. The continuous-time systems often originates from the discretization of continuous-time systems due to digitalization. A continuous-time linear system is defined by the iteration

$$\dot{x}(t) = Ax(t), x(t) \in \mathcal{X} \tag{3.1}$$

where $A \in \mathbb{R}^{n \times n}$ and $\mathcal{X} \subseteq \mathbb{R}^n$.

A discrete-time linear system is defined by the iteration

$$x_k = Ax_{k-1}, x \in \mathcal{X} \tag{3.2}$$

where $A \in \mathbb{R}^{n \times n}$ and $\mathcal{X} \subseteq \mathbb{R}^n$.

Definition 3.0.1 (Invariant set)**.** A set S is *invariant* for system (3.1) (resp. (3.2)) if for any state $x_0 \in S$, the trajectory of the system with initial state x_0 remains in S.

To represent uncertainty due to the imperfection of the model or actual randomness in the system being modeled, a disturbance is added to the system:

$$\dot{x}(t) = Ax(t) + Cw(t), x(t) \in \mathcal{X}, w(t) \in \mathcal{W} \tag{3.3}$$

$$x_k = Ax_{k-1} + Cw_{k-1}, x_k \in \mathcal{X}, w_k \in \mathcal{W} \tag{3.4}$$

where $A \in \mathbb{R}^{n_x \times n_x}, C \in \mathbb{R}^{n_x \times n_w}$.

We sometimes have a way to impact the state space x of the system. This is modeled using a control input u:

$$\dot{x}(t) = Ax(t) + Bu(t), x(t) \in \mathcal{X}, u(t) \in \mathcal{U} \tag{3.5}$$

$$x_k = Ax_{k-1} + Bu_{k-1}, x_k \in \mathcal{X}, u_k \in \mathcal{U} \tag{3.6}$$

where $A \in \mathbb{R}^{n_x \times n_x}, B \in \mathbb{R}^{n_x \times n_u}$.

We will see that there is a strong relation between these systems and both disturbance algebraic systems:

$$D\dot{x}(t) = Ax(t), x(t) \in \mathcal{X} \tag{3.7}$$
$$Dx_k = Ax_{k-1}, x_k \in \mathcal{X} \tag{3.8}$$

and control algebraic systems:

$$E\dot{x}(t) = Ax(t), x(t) \in \mathcal{X} \tag{3.9}$$
$$Ex_k = Ax_{k-1}, x_k \in \mathcal{X} \tag{3.10}$$

where $E, A \in \mathbb{R}^{n\times r}$. The integer r is usually smaller than n which gives a choice for $\dot{x}(t)$ or x_k.

Definition 3.0.2 (Robust invariant set)**.** A set S is *robust invariant* for the system (3.3) (resp. (3.4), (3.7) or (3.8)) if for any state $x_0 \in S$, all trajectories of the system with initial state x_0 remain in S.

Definition 3.0.3 (Controlled invariant set)**.** A set S is *controlled invariant* for system (3.5) (resp. (3.6), (3.9) or (3.10)) if for any state $x_0 \in S$, there exists a trajectory with initial state x_0 that remains in S.

Remark 3.0.1. The difference between systems (3.3), (3.4) and systems (3.5), (3.6) is the nature of the external input w or u. The disturbance w has an adversarial nature, it is considered arbitrarily chosen in $\mathcal{W}$ and cannot be controlled. On the other hand, we can choose the control u in $\mathcal{U}$ that fits our purpose. For this reason, the quantifier used in Definition 3.0.2 is "all" while the quantifier used in Definition 3.0.3 is "exists". A similar distinction applies for algebraic systems. For disturbance algebraic systems, the choice is arbitrary while for control algebraic systems, we can choose the vector $\dot{x}(t)$ or x_k among all of them with have the same image through E.

The *minimal invariant (convex) superset* of a set $\mathcal{S}$ is the intersection of all the invariant (convex) supersets of $\mathcal{S}$ and the *maximal invariant (convex) subset* of a set $\mathcal{S}$ is the union (resp. convex hull) of all the invariant (convex) subsets of $\mathcal{S}$.

The minimal (resp. maximal) invariant set is indeed an invariant set since the intersection (resp. union, convex hull) of invariant sets is invariant.

3.1 Hybrid Systems

In the previous section, the state space $\mathcal{X}$ considered is a subset of $\mathbb{R}^n$ which is usually convex. However, there is an increasing need to model also a state

q that belongs to a small union of discrete values. We refer to x as the continuous part and q as the discrete part of the state space. A system with a purely continous state space is adequately modeled by one of the systems defined in the previous section. A system with purely discrete state space can be modeled by an automaton.

Definition 3.1.1. We define the *automaton* $G(V, E)$ as a directed and labelled strongly connected graph with nodes V, edges E and possibly with parallel edges such that no node has zero ingoing or outgoing degree. Each edge $(q, q', \sigma) \in E$ represents a transition from node $q \in$ *node* to node $q' \in V$, where $\sigma \in [m]$ is the *label*, corresponding to the transition.

When the state space has both a continuous and discrete part, we say that it is *hybrid*. Arguably the simplest model of hybrid system is the discrete-time switched linear system. A discrete-time switched linear system is characterized by a finite set of matrices $\mathcal{A} \triangleq \{A_1, A_2, \dots, A_m\} \subset \mathbb{R}^{n\times n}$ and the iteration

$$x_k = A_{\sigma_k} x_{k-1}, \quad \sigma_k \in [m] \tag{3.11}$$

where $[m]$ denotes the set $\{1, \dots, m\}$.

In some applications the values that σ_k can take in (3.11) may depend on $\sigma_{k-1}, \sigma_{k-2}, \dots$. These constraints are often conveniently represented using a *finite automaton* and the JSR under such constraints is called *constrained joint spectral radius* (CJSR) [Dai12]; an example of constrained switched system is given by Example 3.1.1 and its automaton is illustrated by Figure 3.1. Constrained switched systems are used in a variety of applications including networked control [BL00; Zha+05] and coordination of a network of autonomous agents [JL+03]. Moreover, even if a switched system is *un*constrained, studying an associated *constrained* system generated by *path-complete* methods enhance our ability to analyze the stability [Ahm+14] or stabilize [Gom+18a] the original *un*constrained switched system.

The following will serve as a running example of constrained switched system.

Example 3.1.1 (Running example)**.** We borrow the example of [Phi+16, Section 4]. The set of matrices $\mathcal{A}$ is composed of the following four matrices

$$A_1 = A + B\begin{pmatrix} k_1 & k_2 \end{pmatrix}, \qquad A_2 = A + B\begin{pmatrix} 0 & k_2 \end{pmatrix},$$
$$A_3 = A + B\begin{pmatrix} k_1 & 0 \end{pmatrix}, \qquad A_4 = A.$$

where $k_1 = -0.49$, $k_2 = 0.27$,

$$A = \begin{pmatrix} 0.94 & 0.56 \\ 0.14 & 0.46 \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The corresponding automaton is represented by Figure 3.1.

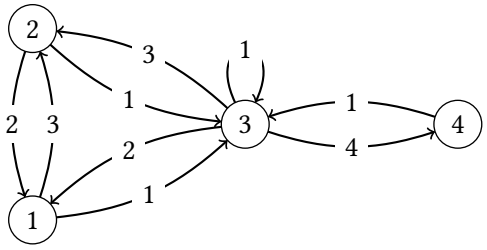


Figure 3.1: Automaton for the running example. The numbers on the edges are their respective labels.

The iteration 3.11 is rewritten as follows to take the automaton into account:

$$x_k = A_{\sigma_k} x_{k-1}, \qquad (\sigma_1, \ldots, \sigma_k) \text{ are the respective labels of a path in } G. \tag{3.12}$$

The *arbitrary switching* case (3.11) can be seen as the particular case when the automaton has only one node and m self-loops with labels $1, \ldots, m$.

Note that (3.12) is not time-invariant as the value of σ_k depends on previous values. This resolved by adding the discrete part of the state space explicitly:

$$x_k = A_{\sigma_k} x_{k-1}, \qquad (q_{k-1}, q_k, \sigma_k) \in G. \tag{3.13}$$

In general, the continuous part of the state space may exhibit both continuous-time and discrete-time dynamics. In order to model such behavior, a common approach is to use *hybrid automata*.

Definition 3.1.2 (Hybrid automaton [Alu+95; Joh+99])**.** A *hybrid automaton* is a collection

$$(G(V, E), \mathcal{X}, f, \text{Init}, \text{Guard}, R)$$

where:

- $G(V, E)$ is an automaton, as defined in Definition 3.1.1.
- $\mathcal{X}_q \subseteq \mathbb{R}^{n_q}$ is the state space constraint set for each node q;
- f_q is a time-invariant vector fields such that each $f_q(x)$ is Lipschitz continuous on $\mathcal{X}_q$ for each node q;
- Guard_σ defines a guard set for each transition.
- $R_\sigma \in \mathcal{X}_{q'}$ specifies how the continuous state is reset for a transition $(q, q', \sigma) \in E$.

A trajectory of the system is a piecewise continuous trajectory $x(t)$, a sequence $(q_k)_k \in V$ and a strictly increasing sequence $(t_k)_k$ such that for all k, we have

$$\begin{aligned} \dot{x}(t) &= f_{q_{k-1}}(x(t)), \quad x(t) \in \mathcal{X}_{q_{k-1}}, \quad t_{k-1} < t < t_k, \\ x(t_k^+) &= R_{\sigma_k}(x(t_k^-)), \quad x(t_k^-) \in \text{Guard}_{\sigma_k} \\ (q_{k-1}, q_k, \sigma_k) &\in G. \end{aligned}$$

We allow the state space of different nodes to differ by having different dimensions n_q as our analysis naturally extends to different state spaces but the reader may consider them to have identical dimension for simplicity.

When $\text{Guard}_\sigma(x) \neq \mathbb{R}^{n_q}$, the switching is called *state-dependent*. When $\mathcal{X}_q, \text{Guard}_\sigma$ are symmetric and f_q, R_σ are homogeneous for each node $q \in V$ and signal $\sigma \in [m]$, we say that the hybrid automaton is *homogeneous*.

The constrained switched system (3.13) is a special case of hybrid automaton where $\mathcal{X}_q = \mathbb{R}^n$ and $f_q(x) = 0$ (no continuous-time dynamics) for each $q \in V$, $\text{Guard}_\sigma(x) = \mathbb{R}^n$ (the switching is not state-dependent), and $R_\sigma(x) = A_\sigma$.

Another variant[1] of hybrid automaton we consider in this thesis is the class of Discrete-Time Affine Hybrid Control System. Similarly to constrained switched systems, these systems also do not have any continuous dynamic and the switching also not state-dependent.

Definition 3.1.3. A *Discrete-Time Affine Hybrid Control System (HCS)* is a system $S = (T, (A_\sigma, B_\sigma, c_\sigma)_{\sigma\in\Sigma}, (\mathcal{P}_q, U_q)_{q\in V})$ where $T = (V, \Sigma, \rightarrow)$, V is a finite set of nodes, Σ is a finite set of signals and $\rightarrow \subseteq V \times \Sigma \times V$ is a set of transitions. We denote $(q, \sigma, q') \in \rightarrow$ by $q \rightarrow_\sigma q'$.

Given a node $q \in V$, we denote the state dimension as $n_{q,x}$ and the input dimension as $n_{q,u}$. The set $\mathcal{P}_q \subseteq \mathbb{R}^{n_{q,x}}$ is the *safe set* corresponding to node q and the set $U_q \subseteq \mathbb{R}^{n_{q,u}}$ is the set of allowed inputs. For any transition $q \rightarrow_\sigma q'$, we have $A_\sigma \in \mathbb{R}^{n_{q',x} \times n_{q,x}}$, $B_\sigma \in \mathbb{R}^{n_{q',x} \times n_{q,u}}$ and $c_\sigma \in \mathbb{R}^{n_{q',x}}$.

A trajectory of S is a sequence $\{(x_k, u_k, \sigma_k)\}_{k\in\mathbb{N}}$ satisfying for all $k \in \mathbb{N}$:

$$\begin{aligned} x_{k+1} &= A_{\sigma_k} x_k + B_{\sigma_k} u_k + c_{\sigma_k}, \\ x_k \in \mathcal{P}_{q_k}, u_k &\in \mathcal{U}_{q_k}, q_k \rightarrow_{\sigma_k} q_{k+1}. \end{aligned}$$

We say that the HCS is homogeneous if $\mathcal{P}_q$ is symmetric for all $q \in V$ and c_σ is zero for every signal σ.

We illustrate this definition with the cruise control example of [RMT13].

Example 3.1.2 (see [RMT13, Section 6.1])**.** We consider a truck with M trailers as represented by Figure 3.2. There is a truck with mass m_0 and speed v_0

[1] This is not strictly a subclass of hybrid automaton since we add a control input.

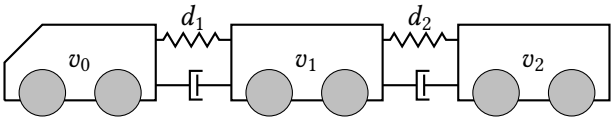


Figure 3.2: Illustration for Example 3.1.2 with two trailers.

followed by multiple trailers, each with mass m. The speed of the ith trailer is denoted v_i. There is a spring with stiffness k_s and elongation d_1 (resp. d_i) and a damper with coefficient k_d between the truck and the first trailer (resp. the $(i-1)$th trailer and the ith trailer). The scalar input u controls the speed v_0 of the truck by creating a force $m_0 u$. The dynamics of the system is given by the following equations:

$$\begin{aligned}
\dot{v}_0 &= \frac{k_d}{m_0}(v_1 - v_0) - \frac{k_s}{m_0} d_1 + u \\
\dot{v}_i &= \frac{k_d}{m}(v_{i-1} - 2v_i + v_{i+1}) + \frac{k_s}{m}(d_i - d_{i+1}) \qquad 1 \le i < M \\
\dot{v}_M &= \frac{k_d}{m}(v_{M-1} - v_M) + \frac{k_s}{m} d_M \\
\dot{d}_i &= v_{i-1} - v_i \qquad 1 \le i \le M.
\end{aligned} \tag{3.14}$$

The spring elongation should always remain between -0.5 m and 0.5 m and the speeds of the truck and trailers should remain between $5\,\mathrm{m\,s^{-1}}$ and $35\,\mathrm{m\,s^{-1}}$. Moreover, there are three speed limits $\bar{v}_a = 15.6\,\mathrm{m\,s^{-1}}$, $\bar{v}_b = 24.5\,\mathrm{m\,s^{-1}}$, $\bar{v}_c = 29.5\,\mathrm{m\,s^{-1}}$ and whenever the truck is informed of a new speed limit, it has 0.8 s to decrease v_i $(0 \le i \le M)$ below the speed limit.

We sample time with a period of 0.4 s and define an initial node q_{d0} and 6 nodes q_{ij} where $i \in \{a, b, c\}$ is the current speed limitation and $j \in \{0, 1\}$ is the number of sampling times left to satisfy the limit. The transitions are $q_{ij} \to_\sigma q_{\sigma 1}$ for each $i \in \{a, b, c, d\}$ and $\sigma \in \{a, b, c, d\} \setminus \{i\}$. The signal a (resp. b, c) represents that the truck sees a new speed limitation $\bar{v}_a$ (resp. $\bar{v}_b$, $\bar{v}_c$) and d represents that it does not see any new speed limitation. We suppose for simplicity that it is not possible to see a new speed limitation $\bar{v}_\sigma$ from a node $q_{\sigma j}$. The possible transitions are represented in Figure 3.3.

The reset maps $(A_\sigma, B_\sigma, c_\sigma)$ are simply the integration of the dynamical system (3.14) over 0.4 s with a zero-order hold input extrapolation.

Let

$$\begin{aligned}
P_0 &= \{\, (d, v) \in \mathbb{R}^{2M+1} \mid -0.5 \le d \le 0.5, 5 \le v \le 35 \,\}, \\
P_i &= \{\, (d, v) \in \mathbb{R}^{2M+1} \mid v \le \bar{v}_i \,\}, \quad i = a, b, c,
\end{aligned}$$

where $d = (d_1, \ldots, d_M)$, $v = (v_0, \ldots, v_M)$ and inequalities in the two equations above are entrywise. The safe sets are $\mathcal{P}_{q_{d0}} = P_0$ and for $i = a, b, c$, $P_{q_{ij}} = P_0$

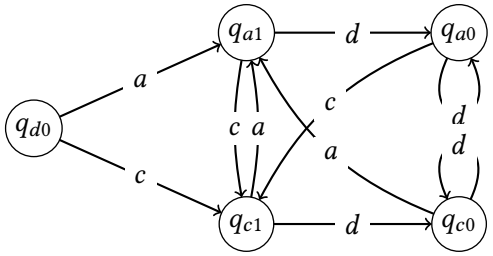


Figure 3.3: Transitions and switchings between the nodes for Example 3.1.2. Nodes q_{b1} and q_{b0} are not shown for clarity.

if $j > 0$ and $P_{q_{i0}} = P_0 \cap P_i$. The input set is $\mathcal{U}_{ij} = \{ u \in \mathbb{R} \mid -4 \leq u \leq 4 \}$ for each node q_{ij}.

Another variant of hybrid automaton is given by Discrete-Time Affine Hybrid Algebraic Systems.

Definition 3.1.4. A *Discrete-Time Affine Hybrid Algebraic System (HAS)* is a system $S = (T, (A_\sigma, E_\sigma, c_\sigma)_{\sigma\in\Sigma}, (\mathcal{P}_q)_{q\in V})$ where $T = (V, \Sigma, \rightarrow)$, V is a finite set of nodes, Σ is a finite set of signals and $\rightarrow \subseteq V \times \Sigma \times V$ is a set of transitions.

Given a node $q \in V$, we denote the state dimension as $n_{q,x}$. The set $\mathcal{P}_q \subseteq \mathbb{R}^{n_{q,x}}$ is the *safe set* corresponding to node q. For any transition $q \rightarrow_\sigma q'$, we have $A_\sigma \in \mathbb{R}^{n_{\sigma,p}\times n_{q,x}}$, $B_\sigma \in \mathbb{R}^{n_{\sigma,p}\times n_{q',x}}$ and $c_\sigma \in \mathbb{R}^{n_{\sigma,p}}$ for some natural number $n_{\sigma,p}$.

A trajectory of S is a sequence $\{(x_k, \sigma_k)\}_{k\in\mathbb{N}}$ satisfying for all $k \in \mathbb{N}$:

$$E_{\sigma_k} x_{k+1} = A_{\sigma_k} x_k + c_{\sigma_k},$$
$$x_k \in \mathcal{P}_{q_k}, q_k \rightarrow_{\sigma_k} q_{k+1}.$$

Definition 3.1.5. We say that a HAS is *homogeneous* if P_q is symmetric for all $q \in V$ and c_σ is zero for all signal σ.

3.2 Stability

We review in this section classical approaches for computing invariant ellipsoids and polysets for the systems defined above.

3.2.1 Continuous-time systems

The *Nagumo condition* provides necessary and sufficient conditions for invariance of continuous-time systems.

Theorem 3.2.1 (Nagumo condition [BM15, Theorem 4.7])**.** A closed set $\mathcal{S}$ is invariant for system (3.1) if and only if

$$\forall x \in \mathcal{S}, Ax \in T_{\mathcal{S}}(x). \tag{3.15}$$

For ellipsoids $\mathcal{E}_P$, the nagumo condition (3.15) is rewritten as

$$x^\top PAx \leq 0, \quad x \in \mathbb{R}^n.$$

which is equivalent to the LMI:

$$A^\top P + PA \preceq 0. \tag{3.16}$$

which allows to search for ellipsoidal invariant sets using semidefinite programming.

3.2.2 Discrete-time systems

Consider an ellipsoidal set $\mathcal{S} = \mathcal{E}_Q$, as defined in (1.14). Using Proposition 1.2.19, the invariant condition of Definition 3.0.1 for a discrete-time linear system (3.2) can be rewritten in terms of Q as: if $x^\top Px \leq 1$ then $x^\top A^\top PAx \leq 1$. By the Proposition 1.4.8, $x^\top Px \leq 1$ implies $x^\top A^\top PAx \leq 1$ if and only if the following LMI holds:

$$Q \succeq A^\top QA. \tag{3.17}$$

Robust stability

With the presence of a disturbance $w \in W = \mathcal{E}_Q$ in the system $x_{k+1} = Ax_k + Bw_k$, the condition becomes:

$$w^\top Qw \leq 1, x^\top Px \leq 1 \Rightarrow (Ax + Bw)^\top P(Ax + Bw) \leq 1. \tag{3.18}$$

and Proposition 1.5.6 can be applied to provide a robust invariance LMI; see [Boy+94, (6.5)].

3.2.3 Discrete-time switched systems

The maximal asymptotic growth rate of (3.11) is given by the *joint spectral radius* (JSR). The JSR $\rho(\mathcal{A})$ of a finite set of matrices $\mathcal{A}$ is defined as

$$\rho(\mathcal{A}) = \lim_{k\to\infty} \max_{\sigma\in[m]^k} \|A_{\sigma_k} \cdots A_{\sigma_2} A_{\sigma_1}\|^{1/k}.$$

This definition is independent of the norm used. The joint spectral radius is alternatively defined as the infimum value of σ such that the monoid generated by the matrices $A_1/\sigma, \ldots, A_m/\sigma$ with the matrix product is bounded; see Definition 1.1.3.

The JSR was introduced by Rota and Strang [RS60] and has many other applications such as wavelets, the capacity of some particular codes, zero-order stability of ordinary differential equations, congestion control in computer networks, curve design and networked and delayed control systems;

see [Jun09] for a survey on the JSR and its applications. Many algorithms exist for estimating the JSR but not much is known on how to generate an infinite sequence of matrices with an asymptotic growth rate close to the JSR. However generating such sequence can be of particular interest, depending on the application, such as exhibiting unstable trajectories for switched linear systems to prevent them from occuring [Gom+18a]. The currently known algorithms generate a sequence of matrices with high spectral radius using methods detailed in Section 5.3.3 and repeat this sequence infinitely [Gri96; GZ08; GP13; JCG14].

Approximating the JSR usually consists in certifying upper bounds $\overline{\gamma}$ to the JSR by exhibiting a Lyapunov function or an invariant set for the matrices $A_i/\overline{\gamma}$. The search for such Lyapunov functions can naturally be written as a convex optimization program; see Program 3.2.1.

The definition of the JSR is generalized as follows for constrained systems.

Definition 3.2.1 ([Dai12])**.** The *constrained joint spectral radius* (CJSR) of a finite set of matrices $\mathcal{A}$ constrained by an automaton G, denoted as $\rho(G, \mathcal{A})$, is

$$\limsup_{k\to\infty} \rho_k(G, \mathcal{A}) = \rho(G, \mathcal{A}) = \lim_{k\to\infty} \hat{\rho}_k(G, \mathcal{A}, \|\cdot\|) \tag{3.19}$$

where

$$\rho_k(G, \mathcal{A}) = \max\left\{ \rho(c) : c \in G_k, c \text{ is a cycle} \right\}, \quad \rho(c) = [\rho(A_c)]^{1/k}, \tag{3.20}$$

and

$$\hat{\rho}_k(G, \mathcal{A}, \|\cdot\|) = \max\left\{ \|A_s\|^{1/k} : s \in G_k \right\}. \tag{3.21}$$

We can readily see that $\rho_k(G, \mathcal{A}) \leq \hat{\rho}_k(G, \mathcal{A}, \|\cdot\|)$ for any k and *submultiplicative*[2] norm $\|\cdot\|$. Equality (3.19) is called the *Joint Spectral Radius Theorem* and was proved in 1992 by Berger and Wang [BW92] in the unconstrained case. Elsner [Els95] provided a somewhat simpler self contained proof in 1995. Both proofs use rather involved results on the joint spectral radius. In the general constrained case, the equality (3.19) was first proved in [Dai12] with the help of heavy-weighted machinery of ergodic theory, and later simpler proofs appeared.

Consider the space of bounded measurable functions on $\mathbb{S}^{n-1} = \{ x \in \mathbb{R}^n : \|x\|_2 = 1 \}$, denoted $\mathcal{B}$, where $\|\cdot\|_2$ is the *Euclidean* norm. Given a function $f(x) \in \mathcal{B}$, we can define the homogeneous[3] function $h(f) \triangleq x \mapsto \|x\|_2 f(x/\|x\|_2)$ on $\mathbb{R}^n$. We define

$$\mathcal{F} = \{ h(f) \mid f \in \mathcal{B} \}. \tag{3.22}$$

[2] A matrix norm $\|\cdot\|$ is *submultiplicative* if $\|AB\| \leq \|A\| \cdot \|B\|$ for all matrices A and B.
[3] A function f is homogeneous if $f(\alpha x) = \alpha f(x)$ for any scalar value α.

Let $\mathcal{F}_+$ (resp. $\mathcal{B}_+$) be the set of nonnegative functions of $\mathcal{F}$ (resp. $\mathcal{B}$) and $\mathcal{F}_{++}$ be the set of positive functions of $\mathcal{F}$. Given two functions $f, g \in \mathcal{F}$, $f \geq 0$ denotes $f \in \mathcal{F}_+$ and $f \geq g$ denotes $f - g \in \mathcal{F}_+$.

Program 3.2.1 (Primal)**.** *Input*: A finite set of matrices $\mathcal{A}$ and an automaton G.

Output: Functions f_v and a number $\overline{\gamma}$.

$$\begin{aligned} &\inf_{f_v \in \mathcal{F}, \overline{\gamma} \in \mathbb{R}} \overline{\gamma} \\ &\text{subject to } f_v(A_\sigma x) \leq \overline{\gamma} f_u(x), \quad \forall x \in \mathbb{R}^n, \forall (u, v, \sigma) \in E, \qquad (3.23) \\ &\qquad f_v(x) \in \mathcal{F}_{++}, \quad \forall v \in V, \\ &\sum_{v \in V} \int_{\mathbb{S}^{n-1}} f_v(x) \, \mathrm{d}x = 1. \qquad (3.24) \end{aligned}$$

Theorem 3.2.2. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V, E)$. We have $\lim_{k \to \infty} \hat{\rho}_k(G, \mathcal{A}, \|\cdot\|) \leq \overline{\gamma}^\star$. for the optimal solution $\overline{\gamma}^\star$ of Program 3.2.1.

Proof. Consider a norm $\|\cdot\|$ of $\mathbb{R}^n$ and its corresponding induced matrix norm of $\mathbb{R}^{n \times n}$. For each $v \in V$, we know by compactness of the unit ball in $\mathbb{R}^n$, continuity and strict positivity of $f_v(x)$ that there exist $0 < \alpha_v \leq \beta_v$ such that

$$\alpha_v \|x\| \leq f_v(x) \leq \beta_v \|x\|$$

for all $x \in \mathbb{R}^n$. Let $\alpha = \min_{v \in V} \alpha_v$ and $\beta = \max_{v \in V} \beta_v$.

For a G-admissible k-uple $(\sigma_1, \sigma_2, \ldots, \sigma_k)$, $\|A_{\sigma_k} \cdots A_{\sigma_1}\| = \sup_{x \neq 0} \frac{\|A_{\sigma_k} \cdots A_{\sigma_1} x\|}{\|x\|}$. Consider a path such that the ith edge has label σ_i for $i = 1, \ldots, k$ and denote the intermediary nodes of that path as $v_0, v_1, \ldots, v_k$. For any $x \in \mathbb{R}^n$, we have

$$\|A_{\sigma_k} \cdots A_{\sigma_1} x\| \leq \alpha_{v_k} f_{v_k}(A_{\sigma_k} \cdots A_{\sigma_1} x) \leq \alpha_{v_k} \overline{\gamma} f_{v_{k-1}}(A_{\sigma_{k-1}} \cdots A_{\sigma_1} x) \leq \alpha_{v_k} \overline{\gamma}^k f_{v_0}(x)$$

and $\|x\| \geq \beta_{v_0} p_{v_0}(x)$ hence $\|A_{\sigma_k} \cdots A_{\sigma_1}\| \leq \frac{\beta_{v_0}}{\alpha_{v_k}} \overline{\gamma}^k \leq \frac{\beta}{\alpha} \overline{\gamma}^k$. Taking the kth root, the limit $k \to \infty$ and using Definition 3.2.1 we obtain the result. □

The gauge function of polysets of degree $2d$ can be used as Lyapunov function.

Theorem 3.2.3 ([PJ08; Phi+16])**.** Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V, E)$. Suppose that there exist $|V|$ strictly positive homogeneous polynomials $p_v(x)$ of degree $2d$ such that

$$p_v(A_\sigma x) \leq \overline{\gamma}^{2d} p_u(x)$$

holds for all edge $(u, v, \sigma) \in E$. Then $\rho(G, \mathcal{A}) \leq \overline{\gamma}$.

Proof. Define $f_v(x) = [p_v(x)]^{\frac{1}{2d}}$ and use Theorem 3.2.2. □

We relax the positivity condition of Proposition 3.2.3 by the more tractable sum-of-squares (SOS) condition and define $\rho_{\text{SOS-}2d}(G, \mathcal{A})$ as the solution of the following SOS restriction of Program 3.2.1.

Program 3.2.2 (Primal)**.** *Input*: A finite set of matrices $\mathcal{A}$ and an automaton G.

Output: Polynomials $p_v(x)$ and a number $\overline{\gamma}$.

$$\inf_{p_v(x)\in\mathbb{R}_{\mathbf{x}},\overline{\gamma}\in\mathbb{R}} \overline{\gamma}$$

$$\text{subject to } \overline{\gamma}^{2d} p_u(x) - p_v(A_\sigma x) \text{ is SOS}, \quad \forall (u,v,\sigma) \in E, \tag{3.25}$$

$$p_v(x) \text{ is SOS}, \quad \forall v \in V, \tag{3.26}$$

$$p_v(x) \text{ is strictly positive}, \quad \forall v \in V, \tag{3.27}$$

$$\sum_{v\in V} \int_{\mathbb{S}^{n-1}} p_v(x)\, \mathrm{d}x = 1.$$

Remark 3.2.1. In practice we can replace (3.26) and (3.27) by "$p_v(x) - \epsilon\|x\|_2^{2d}$ is SOS" for any $\epsilon > 0$. This constrains $p_v(x)$ to be in the interior of the SOS cone, which is sufficient for $p_v(x)$ to be strictly positive. The bounds given in Section 5.2.4 are valid if $p_v(x)$ is in the interior of the SOS cone.

By Proposition 3.2.3, a feasible solution of Program 3.2.2 gives an upper bound for $\rho(G, \mathcal{A})$, and thus, for any positive degree $2d$,

$$\rho(G, \mathcal{A}) \leq \rho_{\text{SOS-}2d}(G, \mathcal{A}). \tag{3.28}$$

Example 3.2.1. Consider the unconstrained system [AP12b, Example 2.1] with $m = 3$:

$$\mathcal{A} = \{A_1 = e_1 e_2^\top, A_2 = e_2 e_3^\top, A_3 = e_3 e_1^\top\}$$

where e_i denotes the ith canonical basis vector.

For any d, a solution to Program 3.2.2 is given by

$$(p(x), \gamma) = (x_1^{2d} + x_2^{2d} + x_3^{2d}, 1).$$

Indeed, for example, with A_1 we have

$$p(e_1 e_2^\top x) = p(x_2, 0, 0) = x_2^{2d} + 0 + 0 \leq x_1^{2d} + x_2^{2d} + x_3^{2d}.$$

Example 3.2.2. Let us reconsider our running example; see Example 3.1.1. The optimal solution of Program 3.2.2 is represented by Figure 3.4 for $2d = 2$ and 12. We can see a big difference between the shape of the sublevel sets for $2d = 2$ and $2d = 4$ while between $2d = 4$ and $2d = 12$, the difference seems to be more subtle.

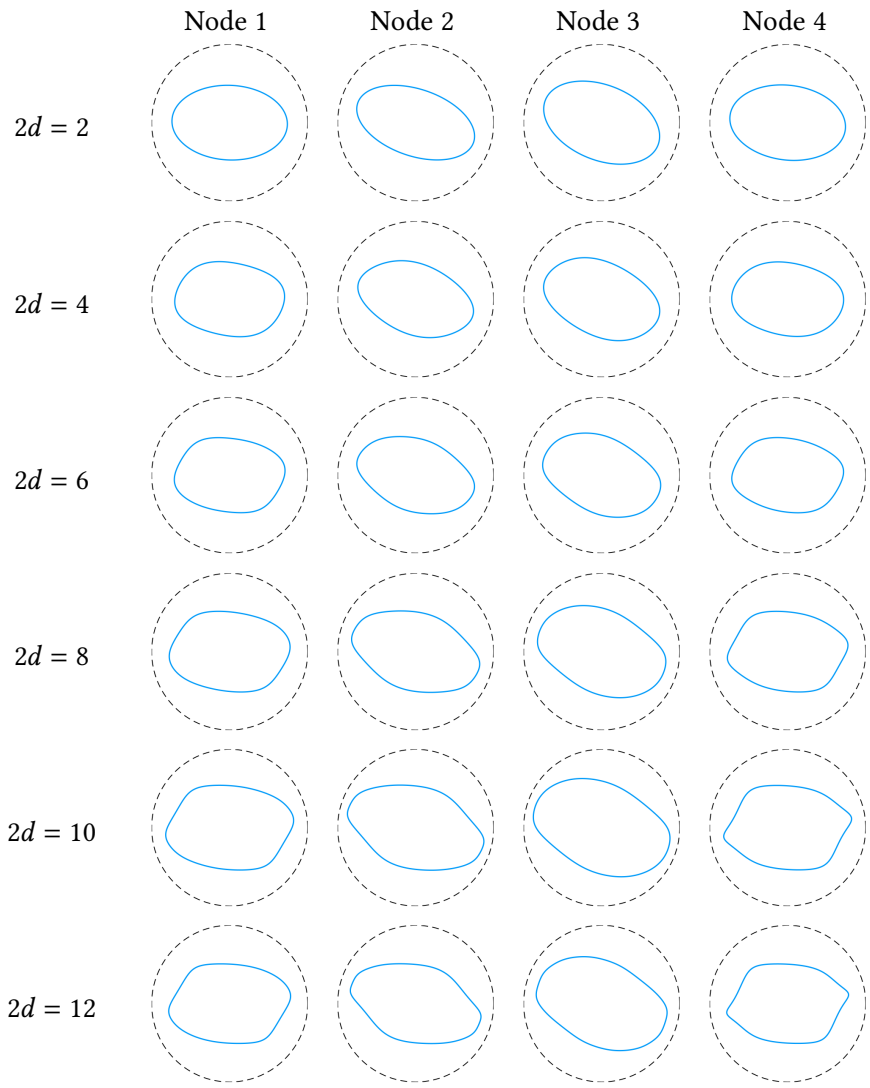


Figure 3.4: Representation of the solutions to Program 3.2.2 with different values of d for the running example. The blue curve represents the boundary of the polyset $\mathcal{P}_{p_v}$ where p_v is the optimal solution for each node $v \in V$. The dashed curve is the boundary of the unit circle. Observe that some sets are not convex.

3.3 State feedback stabilization

The problem of computing a controlled invariant set is a paradigmatic challenge in the broad field of Hybrid Systems control. Indeed, it is for instance crucial in safety-critical applications, such as the control of a platoon of vehi-

cles or air traffic management; see [TPS98], where firm guarantees are needed on our ability to maintain the state in a safe region (e.g., with a certain minimal distance between vehicles). In other situations, the dynamical system might be too complicated to analyze exactly in every point of the state space, yet it can be possible to confine the state within a guaranteed set. Such situations occur frequently in hybrid, embedded, event-triggered systems, because of the complexity of the dynamics.

A set is *controlled invariant* (sometimes also referred to as *viable*) if, any trajectory whose initial point is in the set can be kept inside it by means of a proper control action. Given a system with constraint specifications on the states and/or input, the controlled invariant set can be used to determine initial states such that trajectories with these initial conditions are guaranteed to meet the specifications. Moreover, in some situations, a state feedback control law can be derived from the knowledge of the controlled invariant set; see [Bla99] for a survey.

The existence of a controlled invariant set is equivalent to the stabilizability of a control system [Son83]. A (possibly nonlinear) stabilizable state feedback can be deduced from the controlled invariant set [Bar85].

The stabilizability of a linear time-invariant (LTI) control system is equivalent to the stability of its uncontrollable subspace (which is readily accessible in its Controllability Form) [Won85, Section 2.4]. Indeed, the eigenvalues of its controllable subspace can be fixed to any value by a proper choice of linear state feedback. The resulting controlled system is stable hence an invariant ellipsoid can be determined by solving a system of linear equations [Lia07]. This set is also controlled invariant for the control system. When a control system admits an ellipsoidal controlled invariant set, it is said to be *quadratically stabilizable*. When there exists a linear state feedback such that the resulting controlled system admits an ellipsoidal invariant set, it is said to be *quadratically stabilizable via linear control*.

While the stabilizability of LTI control systems is equivalent to their quadratic stabilizability via linear control, it is no longer the case for *uncertain* or *switched* systems [Pet85]. Furthermore, it is often desirable to find a controlled invariant set of maximal volume (or which is maximal in some direction [AG18]). For such problem, the method detailed above is not suitable as it does not take any volume consideration but more importantly, the maximal volume invariant set may not be an ellipsoid and may not be rendered stable via a linear control. For this reason, the Linear Matrix Inequality (LMI) (3.30) was devised to encapsulate the controlled invariance of an ellipsoid via linear control [Boy+94, Section 7.2.2] and the conservatism of the choice of linear control was analyzed [Son83]. As the linearity of the control was found to be conservative for uncertain systems [Pet85], the LMI (6.5) was found to encapsulate controlled invariance of an ellipsoid via *any* state-feedback [Bar85].

3.3.1 Continuous-time systems

With the presence of the control u in the system (3.5), the Nagumo condition (3.16) becomes:

$$\forall x, \exists u, x^\top P(Ax + Bu) \leq 0. \tag{3.29}$$

The control term u, or more precisely the existential quantifier $\exists$ prevents us to transform this into an LMI directly.

Fixing the control to a linear state feedback $u(x) = Kx$ for some matrix K allows to fallback to the case of uncontrolled system $x_{k+1} = (A + BK)x_k$. The invariance condition can be formulated as the *Bilinear Matrix Inequality* (BMI):

$$A^\top P + PA + K^\top B^\top P + PBK \preceq 0$$

While the matrix inequality is bilinear in K and P, and BMI's are NP-hard to solve in general [TO95], a clever algebraic manipulation allows to reformulate it as a Linear Matrix Inequality (LMI) in $Q := P^{-1}$ and $Y := KQ$, where the sought control-invariant ellipsoid is given by $\mathcal{E}_P$, see e.g. [Boy+94, Section 7.2.1, Section 7.2.2] and [BM15, Section 4.4.1] for more details. The linear matrix inequality is

$$QA^\top + AQ + Y^\top B^\top + BY \preceq 0. \tag{3.30}$$

As the algebraic manipulation which allows to reformulate the BMI into an LMI is done at the level of matrices, it is not clear how this approach can be generalized to sublevel sets of polynomials of higher degree. However, searching for ellipsoidal controlled invariant sets may be rather restrictive and the conservativeness is amplified for the class of hybrid system.

3.3.2 Discrete-time systems

The computation of invariant sets is usually achieved using either polyhedral computations or semidefinite programming. If the system contains a control input, the computational complexity of the problem becomes even more challenging. Indeed, this requires (see e.g., the procedure p. 201 in [BM15]) the computation of projections of polytopes when using polyhedral computations and as we show below semidefinite programming techniques are not directly applicable.

The semidefinite programming approach sacrifices exactness of the solution for the sake of algorithmic tractability. With the presence of the control u in the system $x_{k+1} = Ax_k + Bu_k$, the condition becomes:

$$x^\top Px \leq 1 \Rightarrow \exists u, (Ax + Bu)^\top P(Ax + Bu) \leq 1. \tag{3.31}$$

The control term u, or more precisely the existential quantifier $\exists$ prevents the S-procedure to be directly applied.

Methods based on polyhedral computations for hybrid control systems have been developed in [RMT13; SNO16; RT17]. Unfortunately, the problem of polyhedral projection is well known to severely suffer from the curse of dimensionality, see [ABS95], and the additional complexity of the discrete dynamics in hybrid systems makes the problem even less scalable for these systems. Parametrizations of the set have been proposed to improve the scalability of the polyhedral approach; see [RB10]. Polyhedral computations are typically restricted to affine constraint specifications but it has been recently shown that it can also be applied to algebraic constraints; see [AJ16].

In [KV05], the authors show how to compute an over- and under-approximation of the reachable sets of a hybrid control system. While they approximate *reachable sets* and do not compute *controlled invariant sets*, their approach bears similarities with the method presented in this Chapter 6. However, their technique does not rely on semidefinite programming as they propagate ellipsoidal sets and do not need to enforce any invariance property.

In [HK14], a semidefinite programming method is proposed for the computation of outer approximations of the *region of attraction*[4] (ROA) for polynomial control systems. Invariant set computation and ROA computation are different problems but the authors show how to adapt the method to the computation of outer approximations of the maximal controlled invariant set in [KHJ14]. While the set computed with this method can be a good approximation of the maximal controlled invariant set, it is an outer approximation and is not controlled invariant unless the approximation is exact. In [KHJ13], the authors show that in the uncontrolled case, an inner approximation of the ROA can be obtained as the complement of an outer approximation of the complement of the target set. The latter can be obtained using the technique developed in [HK14]. This may be extended to the computation of inner approximations of invariant sets in the uncontrolled case similarly to how [HK14] was adapted for invariance in [KHJ14]. However, this only applies to uncontrolled systems while our work tackles controlled systems.

Remark 3.3.1. The main challenge preventing the method developed in [HK14; KHJ14; KHJ13] to provide sufficient conditions for controlled invariance is the presence of the existential quantifier in (3.31). Indeed, adapting the method of [KHJ13] to invariance instead of ROA provides a generalization of (3.17) to polynomial systems and invariant polynomial sublevel sets. This technique can be generalized to systems with disturbance similarly to (3.18). In fact, given a control system for which we aim to compute a controlled invariant set inside a given target set, applying this generalization to the system with

[4] Given a time T and a target set, the *region of attraction* (ROA) is the set of all initial conditions such that there exists an admissible trajectory whose state belongs to the target set at time T. Note that the ROA is not necessarily invariant if the target set is not invariant.

control input replaced by a disturbance and the target set replaced by its complement gives exactly the method developed in [KHJ14]. Indeed, the complement of any outer approximation of the maximal controlled invariant set is invariant for the system where the control input is replaced by a disturbance. Replacing the control input by a disturbance has the effect of replacing the existential quantifier of (3.31) with the universal quantifier of (3.18). This shows that the methods of [HK14; KHJ14; KHJ13] reformulate robust invariance conditions into LMIs. For this reason, their technique does not provide sufficient conditions for control invariance and hence does not seem to be readily applicable to the computation of controlled invariant sets.

On a similar note, the complement of any outer approximation of the minimal robust invariant set is invariant for the system where the disturbance is replaced by a control input. Therefore, the method presented in this Chapter 6 can be used to obtain the complement of outer approximations of the minimal robust invariant set.

There is a well-known technique to circumvent the presence of the existential quantifier $\exists$ in (3.31), which allows to formulate the search for an ellipsoidal controlled invariant set of controlled linear discrete systems as a semidefinite program. We describe this technique in the following paragraph for completeness.

Fixing the control to a linear state feedback $u(x) = Kx$ for some matrix K allows to fallback to the case of uncontrolled system $x_{k+1} = (A + BK)x_k$. Using the Proposition 1.4.8 and Proposition 1.1.4, the invariance condition can be formulated as a *Bilinear Matrix Inequality* (BMI) which is NP-hard to solve in general [TO95]. While the matrix inequality is bilinear in K and P, a clever algebraic manipulation allows to reformulate it as a Linear Matrix Inequality (LMI) in $Q := P^{-1}$ and $Y := KQ$, where the sought control-invariant ellipsoid is given by $\mathcal{E}_P$, see e.g. [BPT12, Section 2.2.1]. The linear matrix inequality is

$$\begin{bmatrix} Q & QA^\top + Y^\top B^\top \\ AQ + BY & Q \end{bmatrix} \text{ is positive semidefinite.} \tag{3.32}$$

While the *uncontrolled* invariance LMI constraint (3.17) has size $n \times n$ where n is the state-space dimension, the *controlled* invariance LMI constraint (3.32) has size $(2n) \times (2n)$ (due to the Proposition 1.1.4) which negatively affects the scalability of the computation. Moreover, as the algebraic manipulation which allows to reformulate the BMI into an LMI is done at the level of matrices, it is not clear how this approach can be generalized to sublevel sets of polynomials of higher degree. However, searching for ellipsoidal controlled invariant sets may be rather restrictive and the conservativeness is amplified for the class of hybrid system. For instance, in [AS98] the authors exhibit a simple example of hybrid systems for which no ellipsoidal set is in-

variant although it is shown in [PJ08, Example 2.8] that there exists a quartic form with invariant sublevel set for this system.

The conservativeness may be reduced by considering intersection of ellipsoids with path-complete methods [Ahm+14]. Given the LMI (3.32) for invariance of an ellipsoid or the LMI developed in Chapter 6 for invariant polysets of higher degree, path-complete methods allow to generate LMIs for the invariance of intersection of such sets.

For piecewise semi-ellipsoidal sets, given a state vector $x \in \mathcal{P}_i$, the next iterate is $(A+BK)x$. Therefore, we need to somehow use (3.32) with Q_i and Q_j on the polyhedra $\mathcal{P}_i \cap (A+BK)^{-1}\mathcal{P}_j$. However, because of the reformulation into the decision variable Y, K is not a decision variable of the semidefinite program. It is therefore unclear how to formulate the controlled invariance of a piecewise semi-ellipsoidal set for a linear control system.

3.3.3 Discrete-time switched systems

In this section, we detail the relation between controlled invariant sets of HCS and invariant sets of HAS.

Definition 3.3.1 (Controlled invariant sets for a HCS)**.** Consider a HCS S as in Definition 3.1.3. We say that sets $C = (C_q)_{q\in V}$ are *controlled invariant* for S if $C_q \subseteq \mathcal{P}_q$ for each $q \in V$ and $\forall q \to_\sigma q', x \in C_q$, $\exists u \in \mathcal{U}_q$ such that

$$A_\sigma x + B_\sigma u + c_\sigma \in C_{q'}.$$

In view of Definition 3.3.1, a trajectory of a HCS should be interpreted as follows. Given initial conditions x_0, q_0, for each $k \geq 0$, a transition $q_k \to_{\sigma_k} q_{k+1}$ is first selected autonomously and then the input u_k can be controlled, knowing the selected transition.

Remark 3.3.2. It is important to distinguish three types of switching: *autonomous switching*, *controlled switching* and *stochastic switching*; see details on the difference between autonomous switching and controlled switching in [Lib12, Section 1.1.3]. Definition 3.3.1 is the definition of controlled invariance for autonomous systems and in this thesis we do not consider systems with controlled switching. With controlled switching, "$\forall q \to_\sigma q'$" is replaced by "$\exists q \to_\sigma q'$" in Definition 3.3.1, we do not investigate controlled switching in this thesis. Stochastic switching is considered in Section 6.4 where it is reduced to the case of autonomous switching in Theorem 6.4.1 for the purpose of invariance as the worst case is considered for safety.

Handling controller constraints

We say that the input of a HCS is *unconstrained* if $\mathcal{U}_q = \mathbb{R}^{n_{q,u}}$ for all $q \in V$, otherwise we say that the input is *constrained*. The computation of controlled

invariant sets for a HCS with constrained input can be reduced to the computation of invariant sets for a HCS with unconstrained input as shown by the following lemma.

Algorithm 2 Construct a HCS with unconstrained input given a HCS with constrained input

Require: A HCS $S = (T, (A_\sigma, B_\sigma, c_\sigma)_{\sigma\in\Sigma}, (\mathcal{P}_q, \mathcal{U}_q)_{q\in V})$
Ensure: A HCS $S' = (T', (A_\sigma, B_\sigma, c_\sigma)_{\sigma\in\Sigma'}, (\mathcal{P}'_q, \mathcal{U}'_q)_{q\in V'})$ where $T' = (V', \Sigma', \to')$.
  for all $q \in V$ **do**
    Add node q to V'
    Define $\mathcal{P}'_q := \mathcal{P}_q$
    Define $\mathcal{U}'_q := \mathbb{R}^{n_{q,u}}$
  end for
  for all $q \to_\sigma w$ **do**
    Add node q^σ to V'
    Define $\mathcal{P}'_{q^\sigma} := \mathcal{P}_q \times \mathcal{U}_q$
    Define $\mathcal{U}'_{q^\sigma} := \mathbb{R}^0$
    Add signal σ' to Σ'
    Define $A_{\sigma'} := \begin{bmatrix} A_\sigma & B_\sigma \end{bmatrix}$, $B_{\sigma'} \in \mathbb{R}^{n_{w,x}\times 0}$ and $c_{\sigma'} := c_\sigma$
    Add transition $q \to'_{q^0} q^\sigma$
    Add signal q^0 to Σ'
    Define $A_{q^0} := \begin{bmatrix} I & 0 \end{bmatrix}^\top$, $B_{q^0} := \begin{bmatrix} 0 & I \end{bmatrix}^\top$ and $c_{q^0} := 0$
    Add transition $q^\sigma \to'_{\sigma'} w$
  end for

Proposition 3.3.1. The sets $C = (C_q)_{q\in V}$ are *controlled invariant* for $S = (T, (A_\sigma, B_\sigma, c_\sigma)_{\sigma\in\Sigma}, (\mathcal{P}_q, U_q)_{q\in V})$ if and only if there exist controlled invariant sets $C' = (C'_q)_{q\in V'}$ such that $C'_q = C_q \ \forall q \in V$ for the system returned by Algorithm 2 with input S.

Proof. Consider controlled invariant sets C' for S' and let $C = (C'_q)_{q\in V}$. Given $x \in C_q$ and $q \to_\sigma w$, the controlled invariance of C' ensures that there exists u such that $(x, u) \in C'_{q^\sigma} \subseteq \mathcal{P}_q \times \mathcal{U}_q$ and $A_\sigma x + B_\sigma u + c_\sigma \in C'_w = C_w$. Hence C is controlled invariant for S.

Consider now controlled invariant sets C for S and let $C' = (C'_q)_{q\in V'}$ where $C'_q = C_q$ for each $q \in V$. Given $q \to_\sigma w$, for each $x \in C'_q = C_q$ the controlled invariance of C ensures that there exists $u \in \mathcal{U}_q$ such that $A_\sigma x + B_\sigma u + c_\sigma \in C_w = C'_w$, setting C'_{q^σ} to be the union of these pairs (x, u) makes C' controlled invariant for S'. □

Note that even if the increase of number of nodes in Proposition 3.3.1 induces more sets to compute, hence an increased computation time, the number of nodes added is at most the number of transitions, and thus this increase is limited.

Remark 3.3.3. For a given node q, let $\Sigma_q = \{ \sigma \mid \exists q', q \to_q q' \}$. If Σ_q is a singleton $\{\sigma\}$, we can merge q and q^σ into one state hence have $\mathcal{P}'_q = \mathcal{P}_q \times \mathcal{U}_q$. In that case, C_q will be the projection of C'_q in its state space. Even if Σ_q is not a singleton, we can pick a single $\sigma \in \Sigma_q$ and merge q and q^σ into one state and use the reset map

$$A_{q^0} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \qquad B_{q^0} = \begin{bmatrix} 0 \\ I \end{bmatrix} \qquad c_{q^0} = 0$$

so that switchings $\sigma' \in \Sigma_q \setminus \{\sigma\}$ ignore the part of the state of q that corresponds to the input to be used for σ.

We start by applying Proposition 3.3.1 on a simple example in Example 3.3.1 and then in Example 3.3.2 we detail its application to the system introduced in Example 3.1.2.

Example 3.3.1. Consider the discrete-time linear control system $x_{k+1} = x_k + u_k/2$ where the safe set for x_k is $[-1, 1]$ and the control u_k is constrained to be between -1 and 1. This system can be viewed as a HCS as defined in Definition 3.1.3 with only one node and one transition. As mentioned in Remark 3.3.3, as there is only one transition, we do not have to create an intermediate node hence the system constructed by Proposition 3.3.1 can also represented by a non-hybrid system. This system is planar and has the following dynamics

$$\begin{aligned} x_{k+1} &= x_k + \frac{u_k}{2} \\ u_{k+1} &= u'_k \end{aligned}$$

where the safe set for (x_k, u_k) is $[-1, 1]^2$ and the control u'_k is unconstrained. By Proposition 3.3.1 we know that a set C is controlled invariant for the scalar system if and only if it is the projection into the x-axis of a controlled invariant set C' of the planar system.

Example 3.3.2. We represent in Figure 3.5 the application of the transformation described in Proposition 3.3.1 to the system of Example 3.1.2. We can use Remark 3.3.3 to avoid creating q^d for each q. Moreover, since $(A_\sigma, B_\sigma, c_\sigma)$ does not depend on σ, we can merge all the nodes q^a (resp. q^b, q^c) together into a common state that we name q_{a2} (resp. q_{b2}, q_{c2}).

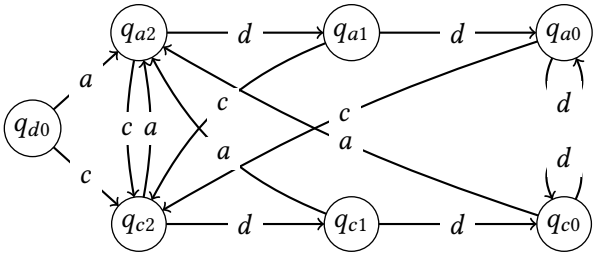


Figure 3.5: Transitions and switchings between the nodes for Example 3.3.2. Nodes q_{b2}, q_{b1} and q_{b0} are not shown for clarity.

Discrete-Time Affine Hybrid Algebraic System

Definition 3.3.2 (Invariant sets for a HAS)**.** Consider a HAS S as in Definition 3.1.4. We say that sets $C = (C_q)_{q\in V}$ are *invariant* for S if $C_q \subseteq \mathcal{P}_q$ for each $q \in V$ and for all $q \to_\sigma q'$,

$$A_\sigma C_q + c_\sigma \subseteq E_\sigma C_{q'}. \tag{3.33}$$

In view of Definition 3.3.2, a trajectory of a HAS should be interpreted as follows. Given initial conditions x_0, q_0, for each $k \geq 0$, a transition $q_k \to_{\sigma_k} q_{k+1}$ is first selected autonomously and then the state x_k such that $E_{\sigma_k} x_{k+1} = A_{\sigma_k} x_k + c_{\sigma_k}$ can be controlled, knowing the selected transition.

Remark 3.3.4. Definition 3.3.2 can be interpreted as stating that C is invariant if for each transition $q \to_\sigma q'$ and $x \in C_q$,

> there *exists* $y \in C_{q'}$ such that $A_\sigma x + c_\sigma = E_\sigma y$.

A similar definition exists, see for instance [LT12], where this last part is replaced by

> for *each* y such that $A_\sigma x + c_\sigma = E_\sigma y$, y must belong to $C_{q'}$.

This is not equivalent to Definition 3.3.2 if A_σ and E_σ are not full rank as discussed in Remark 3.0.1. Moreover, computing ellipsoidal invariant sets according to this definition is much easier: it simply amounts to finding positive definite matrices Q_q such that $A_\sigma^\top Q_q A_\sigma \preceq E_\sigma^\top Q_{q'} E_\sigma$; see [OD85].

With this alternative definition of invariant sets, the invariance of sets of HAS would be equivalent to the robust invariance of sets for a system with disturbance. This differs from Proposition 3.3.2 for which the equivalence is with controlled invariance for a control system. This shows that, similarly to [HK14; KHJ14; KHJ13], the LMI found in [OD85] is related to robust invariance (3.18) and not control invariance (3.31).

We now show that the computation of controlled invariant sets of a HCS can be reduced to the computation of invariant sets of a HAS.

Proposition 3.3.2. The sets $C = (C_q)_{q\in V}$ are *controlled invariant* for the HCS $S = (T, (A_\sigma, B_\sigma, c_\sigma)_{\sigma\in\Sigma}, (\mathcal{P}_q, \mathbb{R}^{n_{q,u}})_{q\in V})$ if and only if they are invariant sets for the HAS $S' = (T, (E_\sigma A_\sigma, E_\sigma, E_\sigma c_\sigma)_{\sigma\in\Sigma}, (\mathcal{P}_q)_{q\in V})$ where $E_\sigma = \pi_{\mathrm{Im}(B_\sigma)^\perp}$.

Proof. By Proposition 1.1.5, as the input is unconstrained, for each $q \to_\sigma q'$ and $x \in \mathcal{P}_q$, there exists $u \in \mathbb{R}^{n_{q,u}}$ such that $A_\sigma x + B_\sigma u + c_\sigma \in C_{q'}$ if and only if $E_\sigma A_\sigma x + E_\sigma c_\sigma \in E_\sigma C_{q'}$. □

In the following example, we apply Proposition 3.3.2 to Example 3.3.1.

Example 3.3.3. Consider the planar system introduced in Example 3.3.1. In this system, $\mathrm{Im}(B)$ is the u-axis hence $\mathrm{Im}(B)^\perp$ is the x-axis. Hence we have $A = \begin{bmatrix}1 & 1/2\end{bmatrix}$ and $E = \begin{bmatrix}1 & 0\end{bmatrix}$.

Part II

Contributions

Set programming

4

In this chapter, we extend Chapter 2 to optimization programs for which decision variables are sets. It is important to distinguish conic programs (2.1), for which the decision variables are *vectors* that belongs to *given cones*, from set programs where the decision variables are *sets* that belongs to *given families* of sets, called *templates*.

This distinction is similar to the extension from first-order logic to second-order logic. In first-order logic, propositions quantify only over variables that represent elements of given sets while in second-order logic, propositions can quantifies over variables that can represent sets that are elements of given families of sets. When considering sets inside propositions, Russell exhibited the famous Russell's paradox: "Let $\mathcal{S} = \{ x \mid x \notin x \}$, then $\mathcal{S} \in \mathcal{S} \Leftrightarrow \mathcal{S} \notin \mathcal{S}$". This paradox is resolved in second-order logic by using a hierarchy of sets that can only contain sets of the lower order. For instance, in second-order logic, the variables are sets that cannot contain sets themselves. The same applies for set programming, the decision variables represent subsets of $\mathbb{R}^n$ or even convex subsets of $\mathbb{R}^n$ hence we shall not encounter Russell's paradox.

A key difference with the set programs we study in this section with the optimization of *submodular functions* is that the later focuses on discrete sets while we focus on subsets of $\mathbb{R}^n$. Given a finite set Ω and a set function $h : 2^\Omega \to \mathbb{R}$, the function f is *submodular* if the condition[1] holds for all subsets $J, K \subseteq \Omega$:

$$h(J) + h(K) \geq h(J \cup K) + h(J \cap K).$$

The concept of *submodular continuous functions* was introduced recently but the functions are defined for continuous vector, not subset of a continuous domain. Given a submodular function h, we can define the function $f : \{0,1\}^{|\Omega|} \to \mathbb{R}$ with $f(x) = h(\{ \omega \mid x_\omega = 1 \})$. The submodularity condition (7.7) is then rewritten as (see e.g. [Bac19, (1)])

$$f(x) + f(y) \geq f(\min(x, y)) + f(\max(x, y))$$

where min and max are applied component-wise. This condition is used to define submodular continuous functions $f : [0,1]^{|\Omega|} \to \mathbb{R}$. If f is twice

[1] It is defined again later in (7.7) but we include it here as well for convenience.

differentiable, it is equivalent to all off-diagonal entries of the hessian being nonpositives, see e.g. [Bia+16, (2)].

A generic set program is defined as follows:

$$\begin{aligned} \min_{\mathcal{S}_1 \subseteq \mathbb{R}^{n_1}, \ldots, \mathcal{S}_N \subseteq \mathbb{R}^{n_N}} \quad & f(\mathcal{S}_1, \ldots, \mathcal{S}_N) \\ \text{subject to:} \quad & g_i(\mathcal{S}_1, \ldots, \mathcal{S}_N) \subseteq h_i(\mathcal{S}_1, \ldots, \mathcal{S}_N), \quad i = 1, 2, \ldots, M. \end{aligned} \tag{4.1}$$

where $\mathcal{S}_i \subseteq \mathbb{R}^{n_i}$ is a set decision variable, $f(\mathcal{S}_1, \ldots, \mathcal{S}_N)$ is the objective function and the inclusion constraints are given by $g_i(\mathcal{S}_1, \ldots, \mathcal{S}_N) \subseteq h_i(\mathcal{S}_1, \ldots, \mathcal{S}_N)$ for $i = 1, \ldots, M$. Note that this form also encapsulates membership constraints $x \in \mathcal{S}$ as it can be encoded as an inclusion constraint $\{x\} \subseteq \mathcal{S}$.

In this chapter, we give the following approach to set programs that is illustrated in Figure 4.1:

1. First, given the properties of gauge and support functions and the program constraints, we determine whether the set variables should be represented with the gauge or support function.

2. Second, we consider the different templates and analyze the program obtained by formulating the program in terms of its gauge or support function, depending on what was determined in the previous section.

The advantage of this approach is that the first step is independent of the actual template and hence it gives a generic geometric approach to the computation of sets that are solutions to the set program instead of the algebraic template-dependent approaches presented in Chapter 3.

For ellipsoids, piecewise semi-ellipsoids (resp. polysets and piecewise polysets), specific classes of set programs can be reformulated as a semidefinite program (resp. Sum-of-Squares programs). The reformulation is done in the second step and interestingly, the interdependence of the different constraints and the objective only appear in the choice of the representation, which is already carried out in a template-independent fashion in the first step. Therefore, the second-step can be done independently for the objective function and for each constraint. For this reason, we consider in each section, the constraints and the objective functions in isolation. Moreover, similarly to the bridges defined in Section 2.1.2, the since each constraint can be reformulated independently, implementing only a few constraint reformulation enables the reformulation of programs made of any of their combinations, as long as there is a valid choice of representation.

Example 4.0.1. Consider the following generic set program modeling the controlled invariance of the control system considered in Example 3.3.1 and

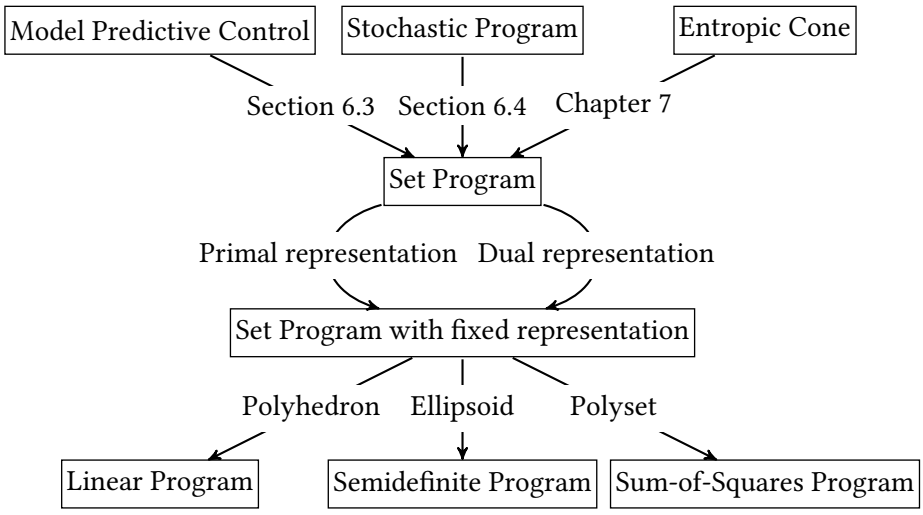


Figure 4.1: Illustration of the advantages of the set programming interface. It first allows different applications such as Model Predictive Control, Stochastic Programming and the Entropic Cone to formulate the set program that should be solved in a template independent manner. Second, the choice between primal representation (e.g. gauge function) or dual representation (e.g. support function) is done in a template independent manner. Lastly, once the representation is determined for each set, each inclusion constraint can be reformulated independently of the rest of the set program.

Example 3.3.3:

$$\begin{aligned} \underset{\mathcal{S}}{\text{maximize}}\ & \operatorname{vol}(\mathcal{S}) \\ & \mathcal{S} \subseteq [-1,1]^2 && (4.2) \\ & C\mathcal{S} \subseteq E\mathcal{S}. && (4.3) \end{aligned}$$

where $C = \begin{bmatrix} 1 & 1/2 \end{bmatrix}$ and $E = \begin{bmatrix} 1 & 0 \end{bmatrix}$.

As discussed in Section 4.4, due to (4.3), it is more appropriate to choose the dual representation (i.e. support function) for this program, independently of the template.

For the ellipsoids template, the dual representation is given by $\delta^*(y|\mathcal{S}) = \sqrt{x^\top Q x}$ as shown in Proposition 1.4.3. By (1.6), the inclusing (4.2) is equivalent to $(1,1), (-1,1), (1,-1), (-1,-1) \in \mathcal{S}^\circ = \mathcal{E}_Q$ which can be encoded as a linear constraint with (1.19). As we will see in Section 4.4, (4.3) can be encoded as the LMI (4.18). In this case, as C, E have only one row, this LMI has dimension 1×1 so it is the linear inequality $Q_{1,2} + Q_{2,2}/4 \leq 0$. The optimal ellipsoid $\mathcal{E}_{Q^{-1}}$ and its polar $\mathcal{E}_Q$ are given respectively in Figure 4.2a and Figure 4.2b.

For the polyset template of degree $2d$, the dual representation is given by $\delta^*(y|\mathcal{S}) = \sqrt[2d]{p(x)}$ where $p(x)$ is a homogeneous polynomial of degree $2d$. We use the volume heuristic consisting in maximizing the integral of $p(x)$ over the square $[-1,1]^2$. This is the method of [DHL17a] discussed in Section 4.2.1. The optimal solution is given in Figure 4.4.

For the piecewise semi-ellipsoidal template, the volume is not directly maximized. Instead, for each cone of the partition, we compute the sum s of the normalized rays and consider the polytope obtained by intersecting the cone with the halfspace $s^\top x \leq \|s\|_2^2$. We integrate the quadratic form corresponding to this cone, i.e. $h^2(\mathcal{S}, x)$ over the polytope. The sum of the integrals over each polytope is the objective function we use. This can be seen as the generalization of the sum of the squares of the length of the semi-axes of the polar of the ellipsoid.

The conic partition is obtained by considering the conic hull of each facets of a given polytope. We first consider the conic partition corresponding to the polar of the square $[-1,1]^2$. This gives the four quadrants as cones of the conic partition. The maximal piecewise semi-ellipsoidal control invariant set with this partition has the following support function:

$$\delta^*(x|\mathcal{S})^2 = \begin{cases} (x_1 - x_2)^2 & \text{if } x_1 x_2 \leq 0, \\ x_1^2 - x_1 x_2/2 + x_2^2 & \text{if } x_1 x_2 \geq 0. \end{cases}$$

For the cones $x_1 x_2 \leq 0$, the semi-ellipsoid matches the square and the maximal control invariant set. For the cones $x_1 x_2 \geq 0$, the semi-ellipsoid matches the optimal solution for the ellipsoid of maximal volume. This illustrates one

key feature of piecewise semi-ellipsoidal sets, they can combine advantages of both polyhedra and ellipsoids. It can be polyhedral on the directions where the maximal control invariant set is polyhedral and be ellipsoidal on the directions where the maximal control invariant set is smooth or requires many halfspaces in its representation. We can use the polar of the polytope resulting from the first fixed point iteration to generate a refined conic partition. With this partition, the optimal solution for piecewise semi-ellipsoids matches the optimal solution of the generic set program. The support function is given by:

$$\delta^*(x|\mathcal{S})^2 = \begin{cases} (x_1 - x_2)^2 & \text{if } x_1 x_2 \leq 0, \\ x_1^2 & \text{if } x_2(x_1 - 2x_2) \geq 0, \\ (x_1/2 + x_2)^2 & \text{if } x_1(2x_2 - x_1) \geq 0. \end{cases}.$$

For the polyhedral template, there are several choices of parametrization of the representation. We consider two cases here and highlight that in both cases, it is more appropriate to use the dual representation, that is, the support function of $\mathcal{S}$ which is given by the V-representation of $\mathcal{S}$ or equivalently the H-representation of $\mathcal{S}^\circ$. This supports the template-independence of this choice of representation.

The V-representation of both $C\mathcal{S}$ and $E\mathcal{S}$ depends linearly on the V-representation of $\mathcal{S}$ as shown in Proposition 1.3.8. The inclusing is ensured by constraining each element of the V-representation of $C\mathcal{S}$ to belong to $E\mathcal{S}$ using Proposition 1.3.10 and Proposition 1.3.12. If the coordinates of each element of the V-representation of $\mathcal{S}$ is fixed, this give a linear program to verify that $\mathcal{S}$ is a feasible solution. Otherwise, if the coordinates of the elements are to be determined it gives a bilinear program. This condition is given in [BM15, Theorem 4.44]. Note that there is no corresponding result for the H-representation in [BM15]. This is in accordance with the claim of Section 4.4 that the dual representation should be used for such inclusion.

Alternatively, given a fixed partition, the support function can be given by

$$\delta^*(y|\mathcal{S}) = \langle a_i, y \rangle \text{ if } y \in \mathcal{P}_i.$$

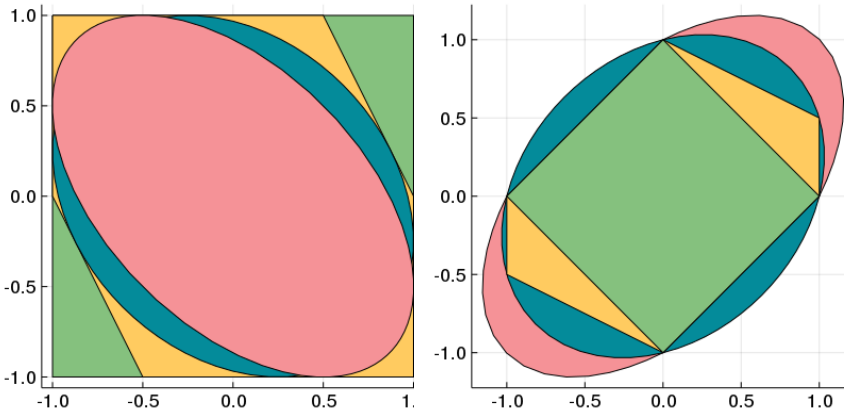
In this case, by Proposition 1.2.20, we have

$$\delta^*(y|C\mathcal{S}) = \langle Ca_i, y \rangle \text{ if } y \in C^\top \mathcal{P}_i.$$

Therefore, $C\mathcal{S} \subseteq E\mathcal{S}$ is equivalent to

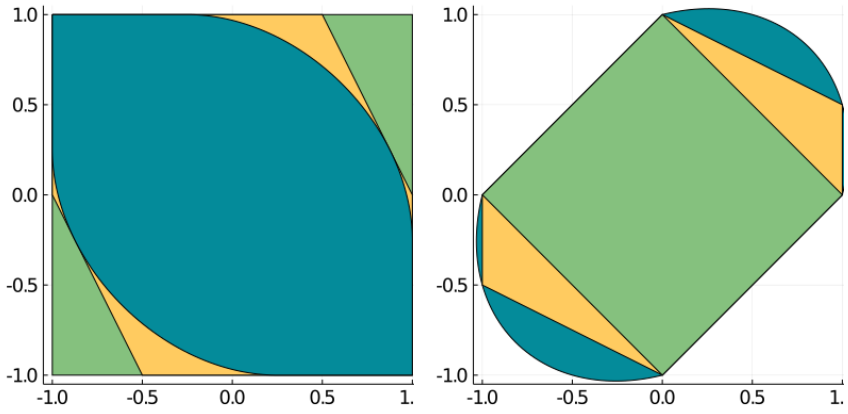
$$\forall i, j, (Ea_j - Ca_i) \in C^\top \mathcal{P}_i \cap E^\top \mathcal{P}_j.$$

If the partition is fixed, this is rewritten as linear constraints using Proposition 1.3.14. This representation of polyhedra is studied in [Rak20].



(a) Primal space (b) Polar space

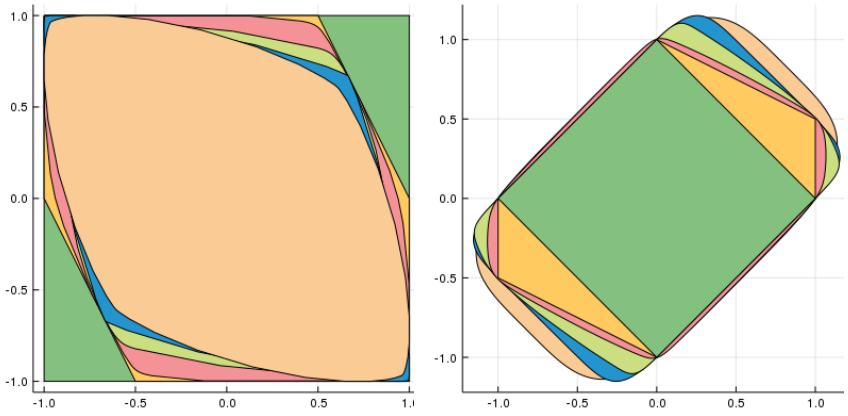
Figure 4.2: Optimal solutions for the set program of Example 4.0.1. The green set is the square $[-1, 1]^2$, the yellow set is the optimal solution of the generic set program, the blue ellipsoid is the optimal solution of the set program for ellipsoids and the red ellipsoid is the optimal solution of the set program for ellipsoids with the objective replaced by the sum of squares of the length of the semi-axes of the polar.



(a) Primal space (b) Polar space

Figure 4.3: Optimal solutions for the set program of Example 4.0.1. The green set is the square $[-1, 1]^2$ and the blue and yellow sets are the optimal solutions of the set program for piecewise semi-ellipsoids with the objective replaced by the integral of $\delta^*(y|\mathcal{S})^2$ over the polar of the square and with different conic partitions.

In Section 4.1, we discuss the choice of representation depending on the objective and constraints. In Section 4.2, we discuss different types of ob-



(a) Primal space **(b) Polar space**

Figure 4.4: Optimal solutions for the set program of Example 4.0.1. The green set is the safe set $[-1,1]^2$, the yellow set is the optimal solution of the generic set program, the optimal solution of the set program for polysets of degree 4, (resp. 8, 18 and 22) is represented in orange (resp. blue, green and red). Note that, as discussed in [DHL17a, Remark 2], their heuristic coincides with the sum of squares of semi-axes for quadratic forms. For this reason, the optimal controlled invariant sublevel set of quadratic forms is the red ellipsoid in Figure 4.2.

jective functions and different ways to handle each type depending on the template of the sets involved in the objective function.

In Section 4.3 and Section 4.4, we investigate the following four types of functions f_i, g_i in the inclusion constraints of the set program.

$$\begin{aligned}
&(\mathcal{S}_1,\dots,\mathcal{S}_N) \mapsto \mathcal{S}_j, && \text{where } j \in [N]\\
&(\mathcal{S}_1,\dots,\mathcal{S}_N) \mapsto A\mathcal{S}_j, && \text{where } j \in [N], A \in \mathbb{R}^{m\times n_j}\\
&(\mathcal{S}_1,\dots,\mathcal{S}_N) \mapsto A^{-1}\mathcal{S}_j, && \text{where } j \in [N], A \in \mathbb{R}^{n_j\times m}\\
&(\mathcal{S}_1,\dots,\mathcal{S}_N) \mapsto \pi_{[n_k],m}A^{-1}\mathcal{S}_j, && \text{where } j,k \in [N], A \in \mathbb{R}^{n_j\times m}.
\end{aligned}$$

In particular, in Section 4.3 we focus on constraints of the form

$$A\mathcal{S}_i \subseteq \mathcal{S}_j, \tag{4.4}$$
$$\mathcal{S}_i \subseteq A^{-1}\mathcal{S}_j. \tag{4.5}$$

with $A \in \mathbb{R}^{n_j\times n_i}$. In Section 4.4, we focus on constraints of the form

$$\mathcal{S}_i \subseteq \pi_{[n_i],n_i+m}\begin{bmatrix}A & B\end{bmatrix}^{-1}\mathcal{S}_j, \tag{4.6}$$
$$C\mathcal{S}_i \subseteq E\mathcal{S}_j. \tag{4.7}$$

with $A \in \mathbb{R}^{n_j\times n_i}$, $B \in \mathbb{R}^{n_j\times m}$, $C \in \mathbb{R}^{r\times n_i}$ and $E \in \mathbb{R}^{r\times n_j}$.

In Section 4.5, we study two variants of set programs consisting in finding the minimal or maximal scaling of a hypersphere in a hypercube. We show small changes in the set program can have dramatic effect on complexity.

When g_i is a union or h_i is an intersection, the constraint can be split into several constraints. When g_i is an intersection of h_i is a union, one typically needs to employ Proposition 1.5.6 for ellipsoids or more generally Corollary 1.5.5 for polysets. In Section 4.6, we study the problem of finding the minimal value of γ and an ellipsoid $\mathcal{E}_P$ such that a given intersection of ellipsoids includes $\mathcal{E}_P$ and is included in $\gamma\mathcal{E}_P$. Note that, by Lemma 1.2.2 and Proposition 1.4.2, this is equivalent to having the convex hull of their union include $\mathcal{E}_{P^{-1}}/\gamma$ and be included in $\mathcal{E}_{P^{-1}}$.

We assume in this chapter that the sets $\mathcal{S}_i$ contain the origin. It might not always be the case, and a translation might not always make it possible. For instance, we might have constraints $\mathcal{S}_1 \subseteq \mathcal{P}_1$ and $\mathcal{S}_2 \subseteq \mathcal{P}_2$ where $\mathcal{P}_1, \mathcal{P}_2$ are two disjoint given polyhedra. In this case, there is no tranlation of the state space that ensure that both $\mathcal{S}_1$ and $\mathcal{S}_2$ contain the origin. In Section 4.7, we discuss how to lift the state space to handle this non-homogeneity.

4.1 Representation

As highlighted in the introduction, for each variable of the set program, we should choose whether the variable is represented using its primal representation (i.e. gauge function) or dual representation (i.e. support function). To determine the representation to use for each variable, we analyse the impact of the choice for the objective and each constraint.

For the objective, if the integral heuristic discussed in Section 4.2.1 is used then either the primal or dual representation can be used as the integral can be done either on the gauge or support function with either maximization or minimization as the integral depends linearly on the parameters of the representation. However, for the $\log\det\cdot$ and $\sqrt[n]{\det\cdot}$ objective for the ellipsoidal template, the representation is determined by the sense of the objective as discussed in Section 4.2.1.

For constraints, as discussed in Section 4.3, constraints of the form (4.4) and (4.5) requires that the sets have the same representation and, as discussed in Section 4.3, constraints of the form (4.6) and (4.7) requires that the sets both have the dual representation.

Some constraints may also have "soft preferences" instead of "hard constraints". For instance, consider a constraint $\mathcal{S} \subseteq \mathcal{P}$ where $\mathcal{S}$ is an ellipsoid and $\mathcal{P}$ is a polyhedron. If the primal representation is used for $\mathcal{S}$ and the polyhedron $\mathcal{P}$ is fixed, the LMI (1.24) should be used. If the polyhedron has a parametrized H-representation then (1.23) should be used instead. If the dual representation is used for $\mathcal{P}$ and the polyhedron $\mathcal{P}$ is fixed, then the linear

constraint (1.22). We see here that if the polyhedron $\mathcal{P}$ is fixed, then the dual representation is preferred as it creates a linear constraints while the primal representation creates an LMI. A similar discussion applies, for a constraint $\mathcal{P} \subseteq \mathcal{S}$ with Proposition 1.4.9. If there is no hard constraint on the representation of $\mathcal{S}$, these soft preferences can be used to generate a semidefinite program that is less computationally expensive to solve.

4.2 Objective

There can be multiple different types of objective functions to a set program.

- First, a *feasibility* set program has no objective function, this is the case in Chapter 5.
- Second, we might want to maximize or minimize a linear combination of the volumes of the set variables. This is the objective chosen in Chapter 6.
- Third, we might want to maximize a set variable $\mathcal{S}$ in a specific direction (see [AG18] for a detailed analysis for a special case of set program with directional objective). For instance, we might to maximize γ such that $\gamma x \subseteq \mathcal{S}$ for a given vector x, or minimize γ such that $\mathcal{S}\gamma \subseteq \mathcal{H}_{a,\alpha}$. This is the case of the *Ingleton score* studied in Chapter 7, see (7.12).

The second type of objective is detailed in Section 4.2.1 and the third one is discussed in Section 4.2.2.

4.2.1 Volume objective

In view of Proposition 1.2.5 (resp. Proposition 1.2.6), minimizing (resp. maximizing) the integral of the gauge-like (resp. support) function over a given set is a sensible heuristic for maximizing the volume. In fact, the volume can be computed directly from the gauge function with the following proposition.

Proposition 4.2.1 ([CG06])**.** The n-dimensional volume of a convex body $C \subseteq \mathbb{R}^n$ is given by

$$\frac{1}{n} \int_{\mathbb{S}^{n-1}} \mathrm{g}^{-n}(C, x)\, \mathrm{d}x. \tag{4.8}$$

However, the expression (4.8) may not be easy to optimize on as it may not be a convex function in terms of the parametrization of a set for a given template.

Ellipsoid volume

As shown in Proposition 1.4.12, in order to maximize (resp. minimize) the volume of an ellipsoid $\mathcal{E}_P$, the determinant $\det P$ should be minimized (resp. maximized). Remark 2.2.1 discusses how to optimize $\det P$ depending on the SDP solver capabilities. If the $\ln\det P$ is not natively supported by the solver and need to be reformulated, a $(2n)\times(2n)$ LMI constraint is added. In view of Remark 2.2.2, this might substantially increase the solve time of the semidefinite program. Moreover, this objective is reformulated into exponential cone constraints. If these constraints are not supported, the objective $\sqrt[n]{\det P}$ can be used instead which can be reformulated into a $(2n)\times(2n)$ LMI constraint and rotated second-order cone constraints. As the $\ln\det P$ and $\sqrt[n]{\det P}$ functions are concave in P, they should be maximized in order for the program to be convex. Therefore, if the volume of an ellipsoid is maximized (resp. minimized), it should be represented as $\mathcal{E}_{P^{-1}}$ (i.e. the dual representation) (resp. $\mathcal{E}_{P^{-1}}$ (i.e. the primal representation)).

Alternatively, the integral of the gauge-like function of degree 2 over a given set can be minimized. By linearity, the integral can be expressed as a linear combination of the entries of Q, multiplied by the integral of the corresponding monomial. It turns out that if this set is the hypercube symmetric around the origin, then this integral is exactly twice the trace of P. In view of Proposition 1.4.13, this is twice the sum of the squares of the length of the semi-axes of $\mathcal{E}_P$.

These two ways to maximize the "size" of the set are discussed in [DPW96]. Even if the trace of P does not correspond to the volume, this objective may render the set "larger" than using the volume for some instances of set programs. We compare these as well in Example 4.0.1, Figure 4.2.

Polyset volume

We discuss in this section three methods for maximizing the volume of a polyset $\mathcal{P}_p$. A first method is to minimize the integral of the gauge-like function of degree $2d$ over a given set. By linearity, the integral can be expressed as a linear combination of the coefficient of the polynomials, multiplied by the integral of the corresponding monomial. The linearity of the expression is appealing as it directly fits in the linear objective of (2.9). This method is discussed further in [DHL17a],

A second (resp. third) method is to minimize the volume of $\mathcal{E}_Q$ where Q is the positive semidefinite matrix certifying that $p(x)$ is SOS-convex (resp. SOS). Note that both methods correspond to the minimization of $\det P$ for an ellipsoid $\mathcal{E}_P$ that was discussed in Section 4.2.1. The second method is developed in [MLB05] and we detail the third method in the remaining of this section.

As discussed in Section 2.3, if we choose some polynomial basis b, there exists some positive semidefinite matrix $Q \in \mathcal{S}^n_+$ such that $p(x) = \langle Qb, b\rangle$ so $p(x)$ can be seen as an ellipsoid in a lifted space [Phi+15].

More precisely, consider the basis $b = x^{[d]}$ and the following operator on sets $\mathcal{S}^{[d]} = \{ x^{[d]} \mid x \in \mathcal{S} \}$. Given an SOS polynomial $p(x) = \langle x^{[d]}, Qx^{[d]}\rangle$ where $Q \in \mathcal{S}^n_+$, we have the following identity

$$(\mathcal{P}_p)^{[d]} = \mathcal{E}_Q \cap (\mathbb{R}^n)^{[d]}.$$

That is, $(\mathcal{P}_p)^{[d]}$ is the intersection of a semi-ellipsoid and the *Veronese variety* $(\mathbb{R}^n)^{[d]}$.

Example 4.2.1. Consider the polynomial $p(x) = (x_1^2 + x_2^2)^2 = x_1^4 + 2x_1^2x_2^2 + x_2^4$. We consider the scaled monomial basis introduced in Section 2.3. The Veronese variety is given by

$$(\mathbb{R}^2)^{[2]} = \{ (x_1^2, x_2^2, x_1x_2) \mid x \in \mathbb{R}^2 \} = \{ y \mid y_3^2 = y_1y_2 \}.$$

The set $(\mathcal{P}_p)^{[d]}$ is the intersection of the ellipsoid $\{ y \mid y_1^2 + y_2^2 + 2y_3^2 \leq 1 \}$ with $(\mathbb{R}^2)^{[2]}$. This is illustrated by Figure 4.5.

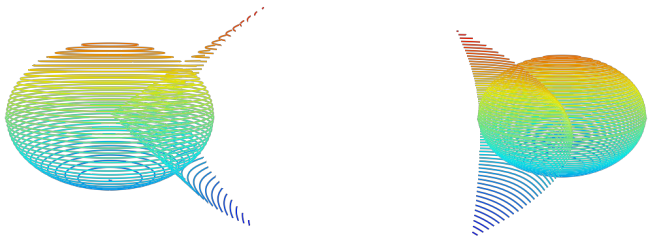


Figure 4.5: Illustration for the Veronese variety.

In view of the preceding discussion, instead of minimizing the volume of $(\mathcal{P}_p)^{[d]}$, it does not seem unreasonable to minimize the volume of the ellipsoid $\mathcal{E}_Q$ which is easily computable thanks to Proposition 1.4.12. Note however that there is a whole affine space of matrices Q such that $p(x) = \langle x^{[d]}, Qx^{[d]}\rangle$ so Q can be chosen in the intersection of this affine space with the cone $\mathcal{S}^n_+$ of semidefinite matrices.

4.2.2 Directional objective

Suppose the objective is to maximize γ such as $\gamma x \in \mathcal{S}$ for a fixed vector x. For polysets, ellipsoids and their piecewise versions as template for $\mathcal{S}$, since the approach is to reformulate the set program as an SOS program, we simply set the linear objective function to γ and add the constraint $\gamma x \in \mathcal{S}$.

If the template for $\mathcal{S}$ is polyhedral, then the method used to compute $\mathcal{S}$ might be completely different from the classical iterative approaches. One drawback of these iterative approaches is that it refines the polyhedral approximation of the optimal value of $\mathcal{S}$ in the generic program in every direction. However, an accurate polyhedral approximation of this set might need a polyhedral representation of prohibitive size. This curse of dimensionality can be circumvented in the case of a directional objective. Algorithm 1, introduced in Section 2.4, can be used to generate cuts to refine a polyhedral approximation in some desired directions only. Consider for instance the following set program with inclusion constraints of the form (4.7) and a directional objective.

$$\begin{array}{rl} \min\limits_{\mathcal{S}_1 \subseteq \mathcal{P}_1, \ldots, \mathcal{S}_N \subseteq \mathcal{P}_N} & \gamma \\ \text{subject to:} & \gamma x \in \mathcal{S}_1 \\ & A_k \mathcal{S}_{i_k} \subseteq E_k \mathcal{S}_{j_k}, k = 1 \ldots, M \end{array} \tag{4.9}$$

We can start with the solutions $\mathcal{S}_i \leftarrow \mathcal{P}_i$ and we can iteratively find the maximum γ such that $\gamma x \in \mathcal{S}_1$, then check whether this satisfy all constraints and if it does not, generate an infeasibility cut as in Algorithm 1. This procedure is detailed in Algorithm 3. It searches over the tree of depth D of all the paths of length D starting at node 1 in the graph underlying the set program where the constraint k is a directed edge from node i_k to j_k. In the context of *Stochastic Programming* introduced in Section 6.4, this tree is made of the *forward pass*, d represents the *time* or *stage* (see Remark 6.4.1), i represents the current node of the markov chain, the integers k such that $i_k = i$ are the transitions index of constraints of (4.9) corresponding to transitions with source i and j_k is the target of the transition.

In Stochastic Programming, instead of generating the cuts "on the fly" during the forward pass as done in Algorithm 3, they are generated in a *backward pass* that start from the leaves of the tree, and generate for each leaf the cut for the node before the leaf in the path ending at this leaf. Then, it generates the cut in the nodes preceding these nodes where a cut was added. The advantage of this approach is that the newly generated cut is used when generating the cuts of the previous nodes in the tree. Indeed, if the cuts where generated forward instead of backward, we would need D pass for a cut in the end of a path of length D to impact a node in the start of the path! Moreover, once an infeasibility is found somewhere in the tree, a packward pass can generate infeasibility cuts up to the node 1 at the start of the path and then the rest of the search tree can be aborted. Indeed, there is no need for generating more than one infeasibility cut for ensuring that $\gamma^\star x$ will not be found again. These two improvements: backward pass and early abortion should be applied to Algorithm 3 in order to improve its practical efficiency.

Another important practical aspect is the order in which the elements are chosen in the set I of Algorithm 3. Indeed, with the early abortion approach presented in the previous paragraph, it is derisable to choose an element in I for that will generate an infeasibility cut as early as possible. Depending on the nature of the instance of the set program (4.9), different ordering may be more appropriate. For some instances, it may be relevant to record the paths in the tree that generate infeasibility cuts more frequently and visit them in priority. It is important to average the number of cuts generated over the number of times they are visited. Indeed, the paths that are visited in priority might generate more cuts hence be even more priorized, this could lead to some paths being neglected.

Algorithm 3 Solving a set program with polyhedral set variables and directional objective.

Input A program (4.9) and a maximal depth D.
Set $\mathcal{S}_i \leftarrow \mathcal{P}_i$.
repeat
  Set $\gamma^\star \in \arg\max\{\, \gamma \mid \gamma x \in \mathcal{S}_1 \,\}$.
  Set $x_1 \leftarrow \gamma^\star x$.
  Set $I = \{(1, \gamma^\star x, 0)\}$.
  while $I \neq \emptyset$ **do**
    Choose $(i, y, d) \in I$.
    if $d < D$ **then**
      for all $k = 1 \ldots, M$ **do**
        if $i_k = i$ **then**
          Search for $z \in \mathcal{S}_{j_k}$ such that $A_k y = E_k z$.
          if This is feasible, a feasible solution z is found **then**
            Set $I \leftarrow I \cup \{(j_k, z, d+1)\}$.
          else
            Generate an infeasibility cut $\mathcal{H}_{a,\alpha}$ from the infeasibility ray as in Algorithm 1.
            Set $\mathcal{S}_i \leftarrow \mathcal{S}_i \cap \mathcal{H}_{a,\alpha}$.
          end if
        end if
      end for
    end if
    Set $I \leftarrow I \setminus \{(i, y, d)\}$.
  end while
until No infeasibility cut is generated in the loop

4.3 Inclusion with one linear image or a preimage

We can see with Proposition 4.3.1 that the constraint (4.4) and the constraint (4.5) are equivalent.

Proposition 4.3.1. Consider a matrix $A \in \mathbb{R}^{n_1 \times n_2}$, a set $\mathcal{S} \subseteq \mathbb{R}^{n_2}$ and a set $\mathcal{T} \subseteq \mathbb{R}^{n_1}$. The inclusion $A\mathcal{S} \subseteq \mathcal{T}$ holds if and only if the inclusion $\mathcal{S} \subseteq A^{-1}\mathcal{T}$ holds.

Proof. The inclusion $A\mathcal{S} \subseteq \mathcal{T}$ is equivalent to $x \in \mathcal{S} \Rightarrow Ax \in \mathcal{T}$. Since $Ax \in \mathcal{T}$ is equivalent to $x \in A^{-1}\mathcal{T}$, we have the desired result. □

.

Remark 4.3.1. Note that $\mathcal{S} \subseteq A\mathcal{T}$ does not imply $A^{-1}\mathcal{S} \subseteq \mathcal{T}$. For instance, if $A = \begin{bmatrix} 1 & 0 \end{bmatrix}$, $\mathcal{S} = [-1, 1]$ and $\mathcal{T} = [-1, 1]^2$ then $\mathcal{S} = [-1, 1] = A\mathcal{T}$ but $A^{-1}\mathcal{S} = [-1, 1] \times \mathbb{R} \not\subseteq \mathcal{T}$.

We can formulate inequalities in terms of either the support (resp. gauge) function of $\mathcal{S}_i$ and $\mathcal{S}_i$ with (4.10) (resp. (4.11)) or we can write them in terms of the gauge (resp. support) function of the polar of $\mathcal{S}_i$ and $\mathcal{S}_j$ with (4.12) (resp. (4.13)).

Theorem 4.3.1. Consider closed convex sets $\mathcal{S}_i, \mathcal{S}_i$ containing the origin. The equivalent constraints (4.4) and (4.5) are equivalent to each of the following equivalent inequalities:

$$\forall y \in \mathbb{R}^{n_j}, \qquad \delta^*(A^\top y | \mathcal{S}_i) \leq \delta^*(y | \mathcal{S}_j). \tag{4.10}$$

$$\forall x \in \mathbb{R}^{n_i}, \qquad \mathrm{g}(\mathcal{S}_i, x) \geq \mathrm{g}(\mathcal{S}_j, Ax). \tag{4.11}$$

$$\forall y \in \mathbb{R}^{n_j}, \qquad \mathrm{g}(\mathcal{S}_i^\circ, A^\top y) \leq \mathrm{g}(\mathcal{S}_j^\circ, y). \tag{4.12}$$

$$\forall x \in \mathbb{R}^{n_i}, \qquad \delta^*(x | \mathcal{S}_i^\circ) \geq \delta^*(Ax | \mathcal{S}_j^\circ). \tag{4.13}$$

Proof. Using (4.4) and Proposition 1.2.20, we obtain (4.10). Using (4.5) and Proposition 1.2.19, we obtain (4.11). Using (4.10) and Proposition 1.2.1, we obtain (4.12). Using (4.11) and Proposition 1.2.1, we obtain (4.13). □

4.3.1 Ellipsoid template

For ellipsoidal sets $\mathcal{S}_i = \mathcal{E}_{Q_i}$, as defined in (1.14), the inequality (4.11) can be rewritten in terms of the matrices Q_i as:

$$Q_i \succeq A^\top Q_j A. \tag{4.14}$$

This LMI could alternatively be obtained in a more template-specific way from the inclusion (4.5) with Proposition 1.4.4 and Proposition 1.4.8.

On the other hand, the inequality (4.10) can be rewritten in terms of the matrices Q_i^{-1} as:

$$AQ_i^{-1}A^\top \preceq Q_j^{-1}.$$

This LMI could alternatively be obtained in a more template-specific way from the inclusion (4.4) with equation (1.16) of Proposition 1.4.5 and Proposition 1.2.6.

4.3.2 Piecewise semi-ellipsoid template

For a piecewise semi-ellipsoidal set, (4.11) is rewritten into:

> for all $i, j = 1, \dots, m$, for all $x \in \mathcal{P}_i \cap A^{-1}\mathcal{P}_j$ such that $x^\top Q_i x \leq 1$, we have $x^\top A^\top Q_j A x \leq 1$.

This is equivalent to

> for all $i, j = 1, \dots, m$, for all $x \in \mathcal{P}_i \cap A^{-1}\mathcal{P}_j$ such that $x^\top (Q_i - A^\top Q_j A)x \geq 0$.

When $\mathcal{P}_i \cap A^{-1}\mathcal{P}_j$ is the nonnegative orthant, this is equivalent to the copositivity of $Q_i - A^\top Q_j A$. See Proposition 1.5.5 for an LMI constraint encoding a sufficient condition for checking the nonnegativity of the quadratic form $x \mapsto x^\top (Q_i - A^\top Q_j A)x$ over the polyhedral cone $\mathcal{P}_i \cap A^{-1}\mathcal{P}_j$.

4.4 Inclusion with a linear image and a preimage

We can see that the inclusion (4.6) can be rewritten in terms of the inclusion (4.7).

Proposition 4.4.1. The inclusion (4.6) is equivalent to the inclusion (4.7) where $C = \pi_{\mathrm{Im}(B)^\perp} A$ and $E = \pi_{\mathrm{Im}(B)^\perp}$.

Proof. A vector $x \in \pi_{[n_i],n_i+m} \begin{bmatrix} A & B \end{bmatrix}^{-1} \mathcal{S}_j$ if and only if there exists a vector u such that $Ax + Bu \in \mathcal{S}_j$. This is equivalent to $(A\{x\} + B\mathbb{R}^m) \cap \mathcal{S}_j \neq \emptyset$. By Proposition 1.1.5, this can be rewritten as

$$\pi_{\mathrm{Im}(B)^\perp}{}^{-1} \pi_{\mathrm{Im}(B)^\perp} Ax \cap \mathcal{S}_j \neq \emptyset$$

or equivalently

$$\pi_{\mathrm{Im}(B)^\perp} Ax \cap \pi_{\mathrm{Im}(B)^\perp} \mathcal{S}_j \neq \emptyset.$$

Therefore, (4.6) is equivalent to

$$\pi_{\mathrm{Im}(B)^\perp} A\mathcal{S}_i \subseteq \pi_{\mathrm{Im}(B)^\perp} \mathcal{S}_j.$$

□

Moreover, as we show below, the inclusion (4.6) does not seem to be easily rewritten in terms of the gauge or support function of $\mathcal{S}_i$ and $\mathcal{S}_j$ without using Proposition 4.4.1. First note that as there is both an image and a preimage in the right-hand side. Therefore, we cannot get an inequality in terms of $\mathrm{g}(\mathcal{S}_j, \cdot)$ because of the image and cannot get an inequality in terms of $\delta^*(\cdot|\mathcal{S}_j)$ because of the preimage. On the other hand, the inclusion $\pi^{-1}_{[n_i],n_i+m}\mathcal{S}_i \subseteq \begin{pmatrix} A & B \end{pmatrix}^{-1} \mathcal{S}_j$ can be rewritten as an inequality in terms of $\mathrm{g}(\mathcal{S}_i, \cdot)$ and $\mathrm{g}(\mathcal{S}_j, \cdot)$ using Proposition 1.2.19 but this inclusion is not equivalent to the inclusion (4.6) as we saw in Remark 4.3.1.

By Proposition 1.2.21, the inclusion (4.6) is equivalent to

$$\mathcal{S}_i^\circ \supseteq \pi^{-1}_{[n_i],n_i+m} \begin{pmatrix} A & B \end{pmatrix}^\top \mathcal{S}_j^\circ \tag{4.15}$$

This inclusion is generalized in [Rak20] for robust control invariant sets of discrete-time control linear systems with disturbances.

Again, there is both an image and a preimage in the right-hand side of the inclusion (4.15), and it is not equivalent to $\pi^\top_{[n_i],n_i+m}\mathcal{S}_i^\circ \supseteq \begin{pmatrix} A & B \end{pmatrix}^\top \mathcal{S}_j^\circ$ by Remark 4.3.1. Hence there does not seem to be any way to use this inclusion to write an inequality in terms of either the gauge or support function of $\mathcal{S}_i^\circ$ and $\mathcal{S}_j^\circ$.

We now consider the inclusion (4.7). This inclusion can be rewritten in terms of the support function of $\mathcal{S}_i$ and $\mathcal{S}_j$ or the Minkowski function of $\mathcal{S}_i^\circ$ and $\mathcal{S}_j^\circ$ as shown in the following theorem.

Theorem 4.4.1. Consider a nonempty closed convex sets $\mathcal{S}_i$, $\mathcal{S}_j$, the inclusion (4.7) is equivalent to the following equivalent inequalities

$$\forall y \in \mathbb{R}^r, \quad \delta^*(C^\top y|\mathcal{S}_i) \leq \delta^*(E^\top y|\mathcal{S}_j) \tag{4.16}$$

$$\forall y \in \mathbb{R}^r, \quad \mathrm{g}(\mathcal{S}_i^\circ, C^\top y) \leq \mathrm{g}(\mathcal{S}_j^\circ, E^\top y). \tag{4.17}$$

Proof. By Proposition 1.2.6, (4.7) is equivalent to

$$\forall y \in \mathbb{R}^r, \quad \delta^*(y|C\mathcal{S}) \leq \delta^*(y|E\mathcal{S}).$$

By Proposition 1.2.20, this inequality is equivalent to (4.16). By Proposition 1.2.1, the inequalities (4.16) and (4.17) are equivalent. □

4.4.1 Ellipsoid template

For ellipsoidal sets $\mathcal{S}_i = \mathcal{E}_{Q_i}$, as defined in (1.14), the inequality (4.17) can be rewritten in terms of the matrices Q_i as:

$$CQ_iC^\top \preceq EQ_jE^\top. \tag{4.18}$$

This LMI could alternatively be obtained in a more template-specific way from the inclusion (4.4) with equation (1.16) of Proposition 1.4.5 and Proposition 1.2.6.

4.4.2 Polyset template

For polar of polysets $\mathcal{S}_i = \mathcal{P}_{p_i}^\circ$ of degree $2d$, the inequality (4.17) can be rewritten in terms of the polynomials Q_i as:

$$\forall x \in \mathbb{R}^r, \quad p_i(C^\top x) \leq p_j(E^\top x).$$

This polynomial inequality can be ensured with the following SOS certificate:

$$p_j(E^\top x) - p_i(C^\top x) \in \Sigma_{2\mathbf{d}}.$$

4.4.3 Piecewise semi-ellipsoid template

For the polar of a piecewise semi-euclidean set, the inequality (4.17) is rewritten in terms of the matrices Q_i and the polyhedra $\mathcal{P}_i$ as

$$x^\top C Q_i C^\top x \leq x^\top E Q_j E^\top x, \forall x \in C^{-\top}\mathcal{P}_i \cap E^{-\top}\mathcal{P}_j.$$

This inclusion can be implemented either with the computational cheap sufficient condition provided by Proposition 1.5.5 or with the more computationally expensive necessary and sufficient condition mentioned in Remark 1.5.1. In both cases, the resulting constraint is an LMI if the polyhedra $\mathcal{P}_i$ are fixed.

Once the piecewise semi-ellipsoidal function is computed, we have the support function of $\mathcal{S}$ or equivalently the Minkowski function of $\mathcal{S}^\circ$. Depending on the application, the Minkowski function of $\mathcal{S}$ might be needed instead. In this case, it can be computed using Proposition 1.4.14. While Proposition 1.4.14 allows to easily go from the piecewise semi-ellipsoidal representation of a set and its polar, the computation of the set satisfying the inclusion (4.7) seems to be more naturally achieved with the piecewise semi-ellipsoidal representation of the polar as detailed in this section.

4.4.4 Piecewise polyset template

For the polar of a piecewise polyset, the inequality (4.17) is rewritten in terms of the matrices Q_i and the polyhedra $\mathcal{P}_i$ as

$$p_i(C^\top x) \leq p_j(E^\top x), \forall x \in C^{-\top}\mathcal{P}_i \cap E^{-\top}\mathcal{P}_j.$$

Suppose the polyhedra $\mathcal{P}_i$ and $\mathcal{P}_j$ are fixed, and let consider an homogeneous H-representation $(a_i)_{i=1}^s \in \mathbb{R}^n$ of $C^{-\top}\mathcal{P}_i \cap E^{-\top}\mathcal{P}_j$. The polynomial inequality over the polyhedral cone can be either with the Schmüdgen's certificate (see Theorem 1.5.6) as follows:

$$p_j(E^\top x) - p_i(C^\top x) = \sum_{\substack{I \subseteq [s], \\ |I| \equiv 0 \pmod 2}} s_I \prod_{i \in I} \langle a_i, x \rangle, \text{ where } s_I \in \Sigma_{2\mathbf{d}-|\mathbf{I}|}$$

Note that we only keep the subsets I of $[s]$ with an even number of elements and take $s_I \in \Sigma_{2\mathbf{d}-|\mathbf{I}|}$ so that so that all terms have degree $2d$. The family of subsets I considered can be reduced to improve scalability at the expense of increased conservatism.

4.4.5 Conclusion

We have shown that the inclusion (4.6) and inclusion (4.7) are equivalent, morover, they can be written as an inequality in terms of the support functions of $\mathcal{S}_i$ and $\mathcal{S}_j$, suggesting that this type of inclusion requires a "dual" representation for the sets.

4.5 Special cases

In this section, we illustrate a special case of set program to illustrate the complexity implication of relatively small changes in the model.

4.5.1 Chebyshev radius and center

Given a set $\mathcal{S} \subseteq \mathbb{R}^n$ and the Euclidean unit ball $\mathcal{E}_{I_n}$, consider the following set program.

Program 4.5.1 (Chebyshev radius and center)**.**

$$\begin{aligned} &\underset{\gamma \geq 0, c \in \mathbb{R}^n}{\text{maximize}} \ \gamma \\ &\{c\} + \gamma \mathcal{E}_{I_n} \subseteq \mathcal{S}. \end{aligned}$$

The optimal value $\gamma^\star$ is the radius of the largest sphere included in $\mathcal{S}$ and $c^\star$ is its center. The values $\gamma^\star$ (resp. $c^\star$) is commonly referred to as a Chebyshev radius (resp. center). In general, the set of Chebyshev centers is not a singleton as shown by the following example.

Example 4.5.1. Consider the rectangle $\mathcal{S} = [-2, 2] \times [-1, 1]$, the Chebyshev radius of $\mathcal{S}$ is 1 and the set of Chebyshev centers is $[-1, 1] \times 0$.

For sets with a non-full-dimensional affine hull, the chebyshev radius is zero. We define for these sets the *relative chebyshev center and radius* as the Chebyshev centers and radius of the set in the affine hull as ambient space.

For a polyhedron

$$\mathcal{S} = \bigcap_{i=1}^{r} \overline{\mathcal{H}}_{a_i, \alpha_i} \cap \bigcap_{i=1}^{s} \mathcal{H}_{b_i, \beta_i},$$

Program 4.5.1 is adapted as in Program 4.5.2 for relative Chebyshev radius and center.

Program 4.5.2 (Relative Chebyshev radius and center of polyhedron)**.**

$$\begin{aligned} &\underset{\gamma\geq 0, c\in\mathbb{R}^n}{\text{maximize}}\ \gamma \\ &\langle a_i, c\rangle = \alpha_i, \qquad \forall i \in [r] \\ &\langle b_i, c\rangle + \gamma\|b_i, 2\| \leq \beta_i, \qquad \forall i \in [s]. \end{aligned}$$

The polyhedron $\mathcal{S}$ is a polytope if and only if Program 4.5.2 is bounded. If $\mathcal{S}$ is a polytope, then the set of optimal solutions for c is a polytope of dimension strictly lower than $\mathcal{S}$.

We define the *proper Chebyshev center* by induction on the dimension of $\mathcal{S}$. If $\mathcal{S}$ has dimension 0 then it is a singleton and its proper Chebyshev center is the only element of $\mathcal{S}$. Otherwise, the dimension of the set $\mathcal{S}'$ of Chebyshev centers of $\mathcal{S}$ is smaller than the dimension of $\mathcal{S}$ and the proper Chebyshev center of $\mathcal{S}$ is the proper Chebyshev center of $\mathcal{S}'$.

For computing the proper Chebyshev center of a polyhedron, the affine hull of the the set of optimal solution of Program 4.5.2 can be computed using the method detailed in Section 1.3.2. Note the relevance of Remark 1.3.1 in the context of the computation of proper Chebyshev centers.

4.5.2 Max-Cut and chebyshev radius

We have seen in Section 4.5.1 that finding the maximum hypersphere in a polytope is a linear program hence it is computationally tractable if the size of the H-representation is reasonable.

Another common interpretation of Chebyshev radius and center is the minimum hypersphere that includes a polytope. These programs are polar to each other hence this problem is computationally tractable if the size of the V-representation of the polytope is reasonable. We call this radius and center the *V-Chebyshev radius and center* to distinguish it from the *H-Chebyshev radius and center* corresponding to the maximum hypersphere in a polytope.

The same conclusions apply if instead of searching for ellipsoids of the form $\gamma\mathcal{E}_I$, we are searching for ellipsoids of the form $\gamma\mathcal{E}_Q$ for a fixed Q. Indeed, consider the Cholesky decomposition of Q, i.e. U such that $U^\top U = Q$. We can apply a linear transformation of the space with $y = Ux$. As U is invertible, the new polyhedron can be seen both as an image with U and a preimage with U^{-1} hence Proposition 1.3.5 and Proposition 1.3.8 ensure that the size of the H-representation and V-representation are unaffected.

The Max-Cut problem can be reduced to finding the minimal γ such that $\{-1,1\}^n \subseteq \gamma\mathcal{E}_Q$ for a fixed positive definite matrix Q. By convexity of $\mathcal{E}_Q$, the inclusion can be replaced by $[-1,1]^n \subseteq \gamma\mathcal{E}_Q$. Here, the polytope under consideration is the hypercube which has a H-representation of size $2n$ and a V-representation of size 2^n. Therefore, the Max-Cut problem is equivalent to the computation of the V-Chebyshev radius of the zonotope obtained as the image of the hypercube $[-1,1]^n$ under U where $U^\top U = Q$. As the V-representation has size 2^n, the number of inequalities of Program 4.5.2 is exponential in n. It therefore does not provide a polynomial time algorithm for the Max-Cut problem. This is to be expected as the Max-Cut problem is NP-hard.

4.6 Löwner-John ellipsoid of intersection of ellipsoids

In this section, we explore the approximation capability of an intersection of ellipsoids by an ellipsoid.

The accuracy of approximation is measured as follows.

Definition 4.6.1. A set $\mathcal{S}$ is a *γ-scaling inner approximation* of a set $\mathcal{T}$ of

$$\gamma = \inf\{\, \gamma > 0 \mid \mathcal{S} \subseteq \mathcal{T} \subseteq \gamma\mathcal{S} \,\}.$$

A set $\mathcal{S}$ is a *γ-scaling outer approximation* of a set $\mathcal{T}$ if $\mathcal{T}$ is a γ-scaling inner approximation of $\mathcal{S}$.

We study the solution of the following set program:

Program 4.6.1.

$$\begin{aligned} \underset{\mathcal{S}}{\text{maximize}}\ & \operatorname{vol}(\mathcal{S}) \\ & \mathcal{S} \subseteq \mathcal{T}. \end{aligned} \tag{4.19}$$

where $\mathcal{T} \subseteq \mathbb{R}^n$ is a given set.

Note that this problem can be reformulated as a semidefinite program for the ellipsoid template when $\mathcal{T}$ is a polyhedron or an intersection of ellipsoids. If $\mathcal{T}$ is a polyhedron, then the inclusion (4.19) is reformulated as linear constraints as shown in Proposition 1.4.10. If $\mathcal{T}$ is an intersection of ellipsoids, then the inclusion (4.19) is reformulated as an LMI with Proposition 1.4.8. The different options for implementing the objective are discussed in Section 4.2.1.

4.6.1 Arbitrary convex body

Bounds are known on the approximation capability of ellipsoids for an arbitrary convex body under the name of Löwner-John theorem. John showed in

[Joh14] that a convex body contains a unique ellipsoid of maximal volume. This ellipsoid is called the *John ellipsoid*. The ellipsoid of minimal volume containing a given convex body is unique as well and is referred to as the *Löwner ellipsoid*. Note that the polar of the John ellipsoid of a convex body $\mathcal{S}$ containing the origin in its interior is the Löwner ellipsoid of the polar of $\mathcal{S}$.

Given a set $\mathcal{S}$ such that the John ellipsoid is $\mathcal{E}_Q$, the John ellipsoid of the set $Q^{1/2}\mathcal{S}$ is the Euclidean ball. The following theorem characterizes the intersection point between the boundary of the John ellipsoid and $Q^{1/2}\mathcal{S}$.

Theorem 4.6.1 ([Bal92])**.** The Euclidean ball $\mathcal{E}_{I_n}$ is the John ellipsoid of a symmetric convex body $\mathcal{S} \subseteq \mathbb{R}^n$ if and only if $\mathcal{E}_{I_n} \subseteq \mathcal{S}$ and there exists $m \geq n$ Euclidean unit vectors $u_1, \ldots, u_m$, on the boundary of $\mathcal{S}$ and positive numbers $c_1, \ldots, c_m$ such that

$$\sum_{i=1}^{m} c_i u_i u_i^\top = I_n. \tag{4.20}$$

It follows from Theorem 4.6.1 that the John ellipsoid is a $\sqrt{n}$-scaling approximation of $\mathcal{S}$. More precisely, we have the following result.

Theorem 4.6.2. The Euclidean ball $\mathcal{E}_{I_n}$ is the John ellipsoid of a symmetric convex body $\mathcal{S} \subseteq \mathbb{R}^n$ if and only if there exists a polyhedron $\mathcal{P}$ of $\mathcal{H}$-representation $((u_i, 1))_{i=1}^m$ with $m \geq n$ Euclidean unit vectors $u_1, \ldots, u_m$ such that

$$\mathcal{E}_{I_n} \subseteq \mathcal{S} \subseteq \mathcal{P} \subseteq \sqrt{n}\mathcal{E}_{I_n}.$$

Proof. We have $\mathcal{E}_{I_n} \subseteq \mathcal{S}$ by definition of the John ellipsoid. By Theorem 4.6.1, there exists $m \geq n$ Euclidean vectors $u_1, \ldots, u_m$ the boundary of $\mathcal{S}$. By Proposition 1.4.6, the normal cone of the Euclidean ball at u_i is $\mathbb{N}_{\mathcal{E}_{I_n}}(u_i) = \operatorname{ray}(\{u_i\})$. As $\mathcal{E}_{I_n} \subseteq \mathcal{S}$, the normal cone of $\mathcal{S}$ is included in the normal cone of $\mathcal{E}_{I_n}$ at x. That is, $\mathbb{N}_{\mathcal{S}}(u_i) \subseteq \operatorname{ray}(\{u_i\})$. As $\mathcal{S}$ is convex, the normal cone is not empty hence $\mathbb{N}_{\mathcal{S}}(u_i) = \operatorname{ray}(\{u_i\})$. This shows that $\mathcal{E}_{I_n} \subseteq \mathcal{S} \subseteq \overline{\mathcal{H}}_{u_i,1}$ hence $\mathcal{E}_{I_n} \subseteq \mathcal{S} \subseteq \mathcal{P}$. Taking the trace on both sides of (4.20), we obtain $\sum_{i=1}^m c_i = n$. Let $(\lambda_i, \ldots, \lambda_m) \in \Delta^{m-1}$ such that $\lambda_i/n = c_i$. The equation (4.20) is rewritten

$$\sqrt{\sum_{i=1}^{m} \lambda_i |\langle u_i, x\rangle|^2} = \frac{\|x\|_2}{\sqrt{n}}.$$

That is, the gauge function of $\mathcal{E}_{I_n/n}$ is the λ-weighted quadratic mean of the nonnegative number $|\langle u_i, x\rangle|$. By Proposition 1.3.3, the polytope $\mathcal{P}$ is the maximum of these same numbers. By Proposition 1.2.4 and Proposition 1.1.3, we conclude that $\mathcal{P} \subseteq \mathcal{E}_{I_n/n}$. By Proposition 1.2.19, $\mathcal{E}_{I_n/n} = \sqrt{n}\mathcal{E}_{I_n}$ hence we have the desired inclusions. □

We define the quantity $\tilde{\beta}(n)$ as the minimum value of γ such that any symmetric convex body $\mathcal{S} \subseteq \mathbb{R}^n$ has a γ-scaling ellipsoidal approximation. By Theorem 4.6.2, we have $\tilde{\beta}(n) \leq \sqrt{n}$. This bound can be shown to be tight with the hypercube as worst case. That is, $\tilde{\beta}(n) = \sqrt{n}$.

Proposition 4.6.1. There is no γ-scaling ellipsoidal approximation of the n-dimensional hypercube with $\gamma < \sqrt{n}$.

4.6.2 Intersection of ellipsoids

In this section, we prove that any intersection of m ellipsoids $\mathcal{E}_{P_1}, \ldots, \mathcal{E}_{P_m}$ can be $\sqrt{\min(m,n)}$-scaling approximated by an ellipsoid.

Consider the following functions:

$$\beta(P_1, \ldots, P_m) = \inf\{\, \gamma \mid \exists Q, \mathcal{E}_Q \subseteq \cap_{i=1}^m \mathcal{E}_{P_i} \subseteq \gamma \mathcal{E}_Q \,\}, \tilde{\beta}(m,n) = \max_{P_1,\ldots,P_m \in \mathcal{S}^n_{++}} \beta(P_1, \ldots, P_m).$$

By Theorem 4.6.2, $\tilde{\beta}(m,n) \leq \sqrt{n}$, we show next that $\tilde{\beta}(m,n) \leq \sqrt{m}$.

Lemma 4.6.1. For any natural numbers m, n,

$$\tilde{\beta}(m,n) \leq \sqrt{m}.$$

Proof. Consider m ellipsoids $\mathcal{E}_{P_1}, \ldots, \mathcal{E}_{P_m}$ and let $Q = \sum_{i=1}^m P_i$. For any vector x, consider the m nonnegative numbers $x^\top P_i x$ for $i = 1, \ldots, m$. By Proposition 1.1.2 with $p = 1$, we have

$$\frac{1}{m} x^\top Q x \leq \max_{i=1}^m x^\top P_i x.$$

By Proposition 1.1.1 with $p = 1$, we have

$$\max_{i=1}^m x^\top P_i x \leq x^\top Q x.$$

By Proposition 1.2.7,

$$\mathrm{g}_2(\bigcap_{i=1}^m \mathcal{E}_{P_i}, x) = \max_{i=1}^m x^\top P_i x/2$$

and by Proposition 1.2.19,

$$\mathrm{g}_2(\sqrt{m}\mathcal{E}_Q, x) = \frac{1}{m} x^\top Q x.$$

Therefore, by Proposition 1.2.4, we have

$$\mathcal{E}_Q \subseteq \bigcap_{i=1}^m \mathcal{E}_{P_i} \subseteq \sqrt{m}\mathcal{E}_Q$$

hence $\tilde{\beta}(m,n) \leq \sqrt{m}$. □

Lemma 4.6.2. For any natural numbers m, n,

$$\tilde{\beta}(m,n) \geq \sqrt{\min(m,n)}.$$

Proof. We give examples with semi-ellipsoids. By continuity, this can be adapted to a sequence of examples with ellipsoids with β converging to the desired value.

If $\min(m,n) = n$, consider $P_i = e_i e_i^\top$ for $i = 1, \ldots, n$ and $P_i = 0$ for $i = n+1, \ldots, m$. The intersection of $\mathcal{E}_{P_i}$ for $i = 1, \ldots, m$ is the symmetric n-dimensional hypercube. We conclude with Proposition 4.6.1.

If $\min(m,n) = m$, consider $P_i = e_i e_i^\top$ for $i = 1, \ldots, m$. The projection of the intersection of $\mathcal{E}_{P_i}$ for $i = 1, \ldots, m$ onto the first m dimensions is the m-dimensional symmetric m-dimensional hypercube. We conclude with Proposition 4.6.1. □

4.6.3 Intersection of ellipsoids with given joint condition number

The worst case example is given by the hypercube which is defined by the intersection of semi-ellipsoids. That is, to reach this example with ellipsoids, the condition number of the ellipsoids need to tend to zero. We investigate in this section whether bounding the condition number allows to prove better approximation guarantees.

Note that $\beta(P_1, \ldots, P_m)$ is invariant under a congruence (i.e. change of basis) but the condition number of $P_1, \ldots, P_m$ is not. This leads us to the following definition.

Definition 4.6.2. The *joint condition number* of matrices $Q_1, \ldots, Q_m$, is given by

$$\kappa(Q_1, \ldots, Q_m) = \inf_{T \in \mathrm{GL}(\mathbb{R}^n)} \sup_{i=1,\ldots,m} \kappa(T^\top Q_i T).$$

Conjecture 4.6.1. Given matrices $P_1, \ldots, P_m \in \mathcal{S}_+^n$ of joint condition number $\kappa = \kappa(P_1, \ldots, P_m)$, let $l = \min(n,m)$, we have

$$\beta(P_1, \ldots, P_m) \leq \sqrt{\frac{l\kappa^2}{l - 1 + \kappa^2}}.$$

The conjecture is proved in Theorem 4.6.3 in the special case of ellipsoids with aligned axes. It also shows that the bound cannot be improved as it is attained when the ellipsoids have all eigenvalues that are either 1 or κ^2 where κ is their joint condition number.

Theorem 4.6.3. Consider the positive semidefinite matrices $P_1, \ldots, P_m \in \mathcal{S}_+^n$, the mutually disjoint subsets $J_1, \ldots, J_m \subseteq [n]$ and the positive numbers $\alpha_{i,j}$ for $i \in [m], j \in I_j$ such that

$$P_i = I_n + \sum_{j in J_i} \alpha_{i,j} e_j e_j^\top.$$

Let $\kappa = \kappa(P_1, \ldots, P_m)$ and l be the number of nonempty set J_i, we have

$$\beta(P_1, \ldots, P_m) \leq \sqrt{\frac{l\kappa^2}{l - 1 + \kappa^2}}$$

with equality if and only if all $\alpha_{i,j}$ are equal.

Proof. The optimizer of $\beta(P_1, \ldots, P_m)$ is

$$Q = I_n + \sum_{i=1}^{n} \sum_{j \in J_i} \alpha_{i,j} e_j e_j^\top.$$

Given a vector x, consider the nonnegative numbers for $i = 1, \ldots, m$ given by

$$a_i = \sum_{j \in J_i} \alpha_j x_j^2.$$

Applying Proposition 1.1.2 with $p = 1$ on the nonnegative numbers $(a_i)_{i=1}^m$, we obtain

$$\|x\|_2^2 + \max_{i=1,\ldots,m} a_i \geq \|x\|_2^2 + \frac{1}{l} \sum_{i=1}^{m} a_i$$

with equality over the variety $\mathcal{V}_1$ defined by equal a_i for all i such that J_i is nonzero. The variety $\mathcal{V}_i$ is nonempty as each a_i is a positive polynomial over different variables. We have $\kappa^2 = 1 + \max_{i,j}(\alpha_{i,j})$ and

$$\begin{aligned}
\|x\|_2^2 + \frac{1}{l} \sum_{i=1}^{m} a_i &= \frac{l-1}{l} \|x\|_2^2 + \frac{1}{m} \left(\|x\|_2^2 + \sum_{i=1}^{m} a_i \right) \\
&= \frac{l-1}{\kappa^2 l} (\|x\|_2^2 + (\kappa^2 - 1)\|x\|_2^2) + \frac{1}{l} \left(\|x\|_2^2 + \sum_{i=1}^{m} a_i \right) \\
&\geq \frac{l-1}{\kappa^2 l} \left(\|x\|_2^2 + \sum_{i=1}^{m} a_i \right) + \frac{1}{l} \left(\|x\|_2^2 + \sum_{i=1}^{m} a_i \right) \\
&= \frac{l - 1 + \kappa^2}{l\kappa^2} x^\top Q x
\end{aligned}$$

with equality when $\alpha_{i,j}$ are equal and x belongs to the variety

$$\mathcal{V}_2 = \{ x \in \mathbb{R}^n \mid x_j = 0, j \in [n] \setminus \cup_{i=1}^m J_i \}.$$

Therefore, by Proposition 1.2.19 and Proposition 1.2.5, we have

$$\bigcap_{i=1}^{m} \mathcal{E}_{Q_i} \subseteq \sqrt{\frac{l\kappa^2}{l - 1 + \kappa^2}} \mathcal{E}_Q$$

and the sets intersect over the nonempty variety $\mathcal{V}_1 \cap \mathcal{V}_2$ if all $\alpha_{i,j}$ are equal. Applying Proposition 1.1.1 with $p = 1$ on the nonnegative numbers a_i, we obtain

$$\|x\|_2^2 + \max_{i=1,\dots,m} a_i \leq \|x\|_2^2 + \sum_{i=1}^{m} a_i = x^\top Q x$$

with equality when at most one of the numbers $a_1, a_2, \dots, a_m$ are nonzero. Hence, by Proposition 1.2.5, we have

$$\mathcal{E}_Q \subseteq \bigcap_{i=1}^{m} \mathcal{E}_{Q_i}.$$

□

The following example exhibits simple instances showing that the bound of Conjecture 4.6.1 cannot be improved using Theorem 4.6.3.

Example 4.6.1. Given natural numbers n, m, let $l = \min(n, m)$ and consider the matrices $Q_i = I_n + (\kappa^2 - 1) e_i e_i^\top$ for $i = 1, \dots, l$ and $Q_i = I_n$ for $i = l+1, \dots, m$. We have

$$\max_{i=1,\dots,m} x^\top Q_i x = \|x\|_2^2 + (\kappa^2 - 1) \max_{i=1,\dots,l} x_i^2.$$

By Theorem 4.6.3, the ellipsoid $\mathcal{E}_Q$ with

$$Q = \begin{bmatrix} \kappa^2 I_l & 0 \\ 0 & I_{n-l} \end{bmatrix}$$

is the optimum in the definition of $\tilde{\beta}$ and is such that

$$\mathcal{E}_Q \subseteq \bigcap_{i=1}^{m} \mathcal{E}_{Q_i} \subseteq \sqrt{\frac{l - 1 + \kappa^2}{l\kappa^2}} \mathcal{E}_Q.$$

The intersection for the first inclusion occurs when at most one of $x_1^2, \dots, x_l^2$ is nonzero. The intersection for the second inclusion occurs when $x_1^2 = \dots = x_l^2$ and $x_{l+1} = \dots = x_n = 0$.

In the case $m = 2$, the invariance under congruence can reduce any case to the case covered by Theorem 4.6.3.

Theorem 4.6.4. Given two positive definite matrices $P_1, P_2 \in \mathbb{R}^{n \times n}$, let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of $P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}}$, we have

$$\beta(P_1, P_2) \leq \sqrt{\frac{2\kappa^2}{1 + \kappa^2}}$$

where $\kappa = \kappa(P_1, P_2) = \max(\max(\lambda), 1/\min(\lambda))$ with equality if and only if $\lambda_i \in \{\kappa^2, 1/\kappa^2\}$ for $i = 1, \dots, n$.

Proof. We have

$$\beta(P_1, P_2) = \beta(P_1^{-\frac{1}{2}} P_1 P_1^{-\frac{1}{2}}, P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}}) = \beta(I, P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}}).$$

Since $P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}}$ is symmetric, there exists U orthogonal such that $P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}} = U\Lambda U^\top$. We therefore

$$\beta(I, P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}}) = \beta(U^\top U, U^T P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}} U) = \beta(I, \Lambda).$$

Let $\Lambda = \mathrm{Diag}(\lambda_1, \dots, \lambda_n)$. Consider the matrix $\underline{\Lambda} = \mathrm{Diag}(\min(1, \lambda_1), \dots, \min(1, \lambda_n))$. We have

$$\beta(I, \Lambda) = \beta(\underline{\Lambda}^{-1/2} I \underline{\Lambda}^{-1/2}, \underline{\Lambda}^{-1/2} \Lambda \underline{\Lambda}^{-1/2}).$$

We conclude with Theorem 4.6.3.

□

Remark 4.6.1. On can have $\beta(P_1, P_2)$ arbitrarily close to 2 by make λ_i tend to 0 or ∞ for each i.

We see with Theorem 4.6.4 that for $n \geq 2$, $\beta(P_1, P_2) < \min(2, n)$.

4.6.4 Conclusion and open questions

In Example 4.6.1, the optimizer of β is the John ellipsoid. It seems reasonable geometrically to expect it to be the case for any instance of m ellipsoids but the questions remains open.

Open Question 4.6.1. Given ellipsoids $\mathcal{E}_{P_1}, \mathcal{E}_{P_2}, \dots, \mathcal{E}_{P_m}$, is the John ellipsoid always an optimizer of $\beta(P_1, P_2, \dots, P_m)$?

Conjecture 4.6.1 is proved for $m = 2$ in Theorem 4.6.4. The case $m > 2$ remains open.

Open Question 4.6.2. Is Conjecture 4.6.1 true for $m > 2$?

The dependence of the bounds not only on the number of matrices m but also on they *joint condition number* could allow to refine the bound obtained in Section 5.2 and provide a way to tailor the design of *path-complete graphs* (see [Ahm+14]) to a specific instance of switched system so as to guarantee the decrease the associated joint condition number and hence ensure the best possible guarantee.

4.7 Handling non-homogeneity

If the sets C does not contain the interior, the gauge function $\mathrm{g}(C, \cdots$ and the polar set C° and not defined and we cannot rely on most of the results of this chapter. We handle this non-homogeneity by taking the conic hull of the lifted sets $C \times \{1\}$. More precisely, we define

$$\tau(C) = \{\, (\lambda x, \lambda) \mid \lambda \geq 0, x \in C \,\} \tag{4.21}$$
$$r(A, c) = \begin{bmatrix} A & c \\ 0 & 1 \end{bmatrix}.$$

It is easy to see that for any set C, vector c and linear map A,

$$\tau(AC + c) = r(A, c)\tau(C). \tag{4.22}$$

Moreover, for any half-space $a^\top x \leq \beta$,

$$a^\top x \leq \beta, \forall x \in C \Leftrightarrow (-a, \beta) \in \tau(C)^*. \tag{4.23}$$

We define the following notation for ellipsoids not necessarily centered at the origin.

$$\mathcal{E}_{Q,c} = \{\, x \mid (x - c)^\top Q(x - c) \leq 1 \,\}$$
$$\mathcal{E}_{D,d,\delta} = \{\, x \mid x^\top Dx + 2d^\top x + \delta \leq 0 \,\}.$$

The following lemma shows the relation between the two notations.

Proposition 4.7.1. Let $Q, D \in \mathcal{S}^n$, $c, d \in \mathbb{R}^n$, $\delta \in \mathbb{R}$ with $Q \succ 0$. We have $\mathcal{E}_{Q,c} = \mathcal{E}_{D,d,\delta}$ if and only if $D \succ 0$ and there exists $\lambda > 0$ such that

$$\lambda = d^\top D^{-1} d - \delta \tag{4.24}$$
$$c = -D^{-1} d \tag{4.25}$$
$$Q = D/\lambda. \tag{4.26}$$

Proof. Substituting Q and c using (4.25) and (4.26) in $(x-c)^\top Q(x-c)-1$ gives $(x^\top Dx + 2d^\top x + d^\top D^{-1} d - \lambda)/\lambda$. We can conclude the "if" part of the proof with (4.24). We now show the "only if" part.

By Proposition 1.5.8, for $\mathcal{E}_{Q,c} = \mathcal{E}_{D,d,\delta}$ to hold, there must exist $\lambda > 0$ such that

$$x^\top Dx + 2d^\top x + \delta = \lambda((x - c)^\top Q(x - c) - 1).$$

This implies that

$$\delta = \lambda c^\top Qc - \lambda \tag{4.27}$$
$$d = -\lambda Qc \tag{4.28}$$
$$D = \lambda Q. \tag{4.29}$$

Equations (4.28) and (4.29) directly give (4.25) and (4.26). It remains to show (4.24). Equation (4.28) is equivalent to $Q^{-1/2}d = -\lambda Q^{1/2}c$ which implies

$$d^\top Q^{-1} d = \lambda^2 c^\top Q c. \tag{4.30}$$

Combining (4.30) with (4.29), we get $\lambda c^\top Q c = d^\top D^{-1} d$ which, combined with (4.27), gives (4.24). □

We use the following corollary to represent the cones $\tau(C_q)^*$ as the 0-sublevel set of quadratic forms $p(y) = p(x,z) = x^\top D_q x + 2d_q^\top xz + \delta_q z^2$.

Corollary 4.7.1. Let $\mathcal{K} = \{ (x,z) | x^\top Dx + 2d^\top xz + \delta z^2 \leq 0, z \geq 0 \}$ be a cone that has a nonempty interior and no intersection with the hyperplane $\{ (x,0) | x \in \mathbb{R}^n \}$ except the origin. The cone $\mathcal{K}$ is convex if and only if $D \succ 0$.

Proof. Let $C = \mathcal{E}_{D,d,\delta}$. Since every point of the cone satisfies $z > 0$ except the origin, we have $\tau(C) = \mathcal{K}$. Therefore, $\mathcal{K}$ is convex if and only if C is convex. Since $\mathcal{K}$ is nonempty,

$$\delta - d^\top D d = \min_{x \in \mathbb{R}^n} x^\top Dx + 2d^\top x + \delta < 0.$$

We conclude with Proposition 4.7.1. □

In Corollary 4.7.1, we require the cone to have no intersection with a particular hyperplane (except the origin). However, the cone $\tau(C_q)^*$ has no intersection with the hyperplane $\{ (x,0) | x \in \mathbb{R}^n \}$ if and only if the origin is contained in C_q which may not be the case. In order to alleviate this, the approach we suggest is to suppose that we know one point h_q in the interior of each C_q and we use Corollary 4.7.1 in a transformed space where h_q is mapped to the z-axis, i.e. the axis with direction vector $(0,1)$. For this transformation we use the *Householder reflection* [GV12, Section 5.1.2]

$$H_h = I - \frac{2}{h^\top h} h h^\top.$$

Observe that the householder reflection is symmetric and orthogonal. In practice, we want a point to be far from the boundary of the sets C_q to improve the numerical conditioning of the problem. An appropriate choice is therefore the proper Chebyshev center of C_q that was defined in Section 4.5.1.

4.8 Conclusion

We have motivated the definition of a Set Programming interface. We argued that the only coupling between the different inclusion constraints of the set program lies in the choice of representation for the sets. That is whether

each set is represented with its gauge or support function. We showed in Section 4.3 and Section 4.4 that this choice of representation is essentially template-independent. Therefore, given a set program, the choice of representation can be done in a template-independent way as a first step. Then, as a second step, the objective and each constraint and be treated in isolation.

In view of this, for each type of inclusion contraint, the choice of representation can be analyzed independently of the other constraints and of a given template. Moreover, the way to reformulate an inclusion constraint in terms of a given representation of the sets in a given template can be studied in complete isolation of the rest of the set program. This is in our opinion a compelling property as it allows to decouple the problem and hence, instead of having to study the reformulation of all the set program obtained with all possible combination of the different type of constraints and objective and all different templates, each subtask can be worked on separately. In addition, without the definition of set program, each combination would also combined with all the different contexts and applications encoutering this set program.

This decoupling property is common in optimization. For instance, in conic programs such as (2.1), one only needs to find a self-concordant barrier for a convex cone in order to be able to mix this cone with a conic program involving arbitrary other cones [Nes04, Section 4].

Stability of switched systems

5

The invariance condition of a set $\mathcal{S}$ for the linear autonomous system (3.2) can be encoded in a set program as constraint (4.4) or (4.5). For a convex set $\mathcal{S}$, depending on the choice of representation of $\mathcal{S}$, it can be rewritten as a inequality between the gauge or support function $\mathcal{S}$.

Theorem 5.0.1. Consider an autonomous system (3.2). The invariance of a closed convex set $\mathcal{S}$ containing the origin is equivalent to each of the following inequalities:

$$\forall y \in \mathbb{R}^n, \qquad \delta^*(A^\top y|\mathcal{S}) \leq \delta^*(y|\mathcal{S}). \tag{5.1}$$
$$\forall x \in \mathbb{R}^n, \qquad \mathrm{g}(\mathcal{S}, x) \geq \mathrm{g}(\mathcal{S}, Ax). \tag{5.2}$$
$$\forall y \in \mathbb{R}^n, \qquad \mathrm{g}(\mathcal{S}^\circ, A^\top y) \leq \mathrm{g}(\mathcal{S}^\circ, y). \tag{5.3}$$
$$\forall x \in \mathbb{R}^n, \qquad \delta^*(x|\mathcal{S}^\circ) \geq \delta^*(Ax|\mathcal{S}^\circ). \tag{5.4}$$

Proof. Apply Theorem 4.3.1 with $\mathcal{S}_i = \mathcal{S}_j = \mathcal{S}$. □

For an ellipsoidal set $\mathcal{S} = \mathcal{E}_Q$, as defined in (1.14), the LMI (4.14) reduces to the LMI (3.17).

The similarity between (5.2) and (5.3) or between (5.1) and (5.4) is reminiscent of the fact that the stability of the system (3.2) is equivalent to the stability of the polar system.

Corollary 5.0.1. A convex set $\mathcal{S}$ is invariant for the autonomous system (3.2) if and only if its polar $\mathcal{S}^\circ$ is invariant for the following polar autonomous system:

$$x_{k+1} = A^\top x_k.$$

This result is generalized in [Rak17] for robust positively invariant sets of discrete-time linear systems with disturbances.

Dropping the convexity requirement, and using (4.11), the stability of a switched system (3.12) can be certified with Program 3.2.1. Selecting the polyset template with degree $2d$ and using the SOS certificate of nonnegativity, we obtain Program 3.2.2.

The stability of the switched system is not equivalent to the feasibility of Program 3.2.2 but guarantees are available for the degree $2d$ needed to certify stability. In Section 5.1, we show that this guarantee is tight for switched systems defined with real matrices. In Section 5.2, we provide a guarantee based on the entropy of the language of admissible switching sequences. In Section 5.3, we introduce a rounding algorithm for extracting an infeasibility certificate for the switched system given an infeasibility certificate of the set program for polyset of fixed degree $2d$. In Section 5.4, we discuss how to minimally restrict the possible switching so as to render the switching system stable, based on the entropy measure of Section 5.2 and the rounding algorithm of Section 5.3. In Section 5.5, we show that if the system is defined by low-rank matrices, its stability can be verified by a set program with inclusing constraints embedded in a low dimensional space, which shall reduce the solve time of the set program in view of Remark 2.2.2.

5.1 Guarantees for unconstrained switched systems

5.1.1 Introduction

Approximating the JSR usually consists in certifying upper bounds $\overline{\gamma}$ to the JSR by exhibiting Lyapunov functions or invariant sets for the matrices $A_i/\overline{\gamma}$. The search for such Lyapunov functions can naturally be written as a convex optimization program using sum-of-squares (SOS) programming [PJ08].

As shown by the following example, exhibited in [AS98], a switched system may not admit any invariant ellipsoid although it is stable.

Example 5.1.1. Consider the switched system with the matrices

$$A_1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}. \tag{5.5}$$

The symmetric cube $\mathrm{conv}(\{(-1,-1),(-1,1),(1,1),(1,-1)\})$ is invariant for this system. The circle of unit radius centered at the origin is feasible solution of Program 3.2.2 with $\overline{\gamma} = \sqrt{n}$. In fact, it is shown in [AS98] that there is no solution for $\overline{\gamma} < \sqrt{n}$ with $d = 1$.

It turns out that these Lyapunov methods cannot produce an arbitrarily bad JSR approximation: bounds are known on the accuracy of the estimate they deliver. Indeed, the following two bounds have been proved in the *unconstrained* case for the lowest upper bound $\overline{\gamma}$ that can be certified using sum-

of-squares polynomials[1] of degree $2d$, denoted $\rho_{\text{SOS-}2d}(\mathcal{A})$:

$$\rho_{\text{SOS-}2d}(\mathcal{A}) \leq \binom{n+d-1}{d}^{\frac{1}{2d}} \rho(\mathcal{A}) \tag{5.6}$$

$$\rho_{\text{SOS-}2d}(\mathcal{A}) \leq m^{\frac{1}{2d}} \rho(\mathcal{A}). \tag{5.7}$$

The bound (5.6) simplifies to

$$\rho_{\text{SOS-}2d}(\mathcal{A}) \leq \sqrt{n}\rho(\mathcal{A}) \tag{5.8}$$

$$\tag{5.9}$$

for ellipsoids, i.e. $d = 1$.

In [AS98], for any positive integer n, switched systems of n matrices of $\mathcal{C}^{n\times n}$ are shown not to admit any invariant ellipsoid even if the system is stable. This shows that guarantee (5.8) is tight for the general class of switched system with matrices of complex entries. Considering the n matrices of $\mathbb{R}^{(2n)\times(2n)}$ corresponding to these complex matrices, we can show that the bound is at most $\sqrt{2n}$ for real matrices but not that it is tight. In this section, we show that the bound (5.8) is tight for real matrices as well.

5.1.2 Tight example with real matrices

We search for matrices $A_\sigma \in \mathbb{R}^{n\times n}$ such that the symmetric hypercube is invariant and a solution of Program 5.3.2 with $\gamma = \sqrt{n}$.

The set of matrices leaving the hypercube invariant is the convex hull of matrices of the form $be_i^\top$ where b is an extreme point of the symmetric hypercube, i.e. $b \in \{-1, 1\}^n$. Indeed, notice that any point exposed by the e_i (resp. $-e_i$) is mapped by $be_i^\top$ to the extreme point b (resp. $-b$). There are $n2^n$ matrices of this form but we consider only matrices such that $b_1 = 1$ to avoid considering matrices opposite to each other. That leaves $n2^{n-1}$ matrices. The number of matrices used for the example of this section is the lowest common denominator of n and 2^{n-1}. We first show it with $n2^{n-1}$ for simplicity.

We first exhibit the following fact.

Lemma 5.1.1. For any positive integer n, we have

$$\sum_{b\in\{1\}\times\{-1,1\}^{n-1}} bb^\top = 2^{n-1} I_n \tag{5.10}$$

Proof. Let Q be the sum of the left-hand side of (5.10). For any i, the ith diagonal entry is 1 for each term so $Q_{ii} = 2^{n-1}$. For $i \neq j$, the entry at row i and column j is 1 if $b_i = b_j$ and -1 otherwise for each term. By symmetry, it is 1 in 2^{n-2} terms and -1 in 2^{n-2} terms so $Q_{ij} = 0$. □

[1] A polynomial $p(x)$ is a *sum-of-squares* if there exists some natural number k and k polynomials $q_i(x)$ such that $p(x) = q_1^2(x) + \cdots + q_k^2(x)$.

Example 5.1.2. For $n = 3$, the four matrices $bb^\top$ are given by

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

We observe that the diagonal entries are 1 and each diagonal entry is 1 for exactly 2 matrices and -1 for the 2 other matrices.

Theorem 5.1.1. Given a positive integer n, consider the switched system with the $n2^{n-1}$ matrices A_σ of the form be_i with $b \in \{1\} \times \{-1, 1\}^{n-1}$ and $i \in \{1, \dots, n\}$. The hypercube is invariant for the switched system but $\rho_{\text{SOS-1}}(G, \mathcal{A}) \geq \sqrt{2}$.

Proof. Let Q_σ denote the moment matrix of the pseudo-measure $\widetilde{\mu}_{uv\sigma}$ for $2d = 1$ and in the unconstrained case. The dual constraints (5.28) is then rewritten as

$$\sum_{\sigma=1}^{m} A_\sigma Q_\sigma A_\sigma^\top \succeq n \sum_{\sigma=1}^{m} Q_\sigma \tag{5.11}$$

Searching for Q_σ of the form $e_{i_\sigma} e_{i_\sigma}^\top$, we have

$$\sum_{\sigma=1}^{m} A_\sigma e_{i_\sigma} (A_\sigma e_{i_\sigma})^\top \succeq n \sum_{\sigma=1}^{m} e_{i_\sigma} e_{i_\sigma}^\top$$

With matrices A_σ of the form $b_\sigma e_{i_\sigma}^\top$, the constraint becomes

$$\sum_{\sigma=1}^{m} b_\sigma b_\sigma^\top \succeq n \sum_{\sigma=1}^{m} e_{i_\sigma} e_{i_\sigma}^\top \tag{5.12}$$

For the choice of matrices of the statement, the right-hand side is $n2^{n-1} I_n$. By Lemma 5.1.1, the left-hand side is equal to $n2^{n-1} I_n$ as well. Hence this is a feasible solution of Program 5.3.2 so $\rho_{\text{SOS-1}}(G, \mathcal{A}) \geq \sqrt{n}$. □

As the terms $b_\sigma b_\sigma^\top$ in the left-hand side of (5.12) are independent of i_σ, we can avoid having duplicated b_σ for i_σ in some situations and use fewer matrices as shown in the following theorem.

Theorem 5.1.2. Given a positive integer n, let $2^d = \gcd(n, 2^{n-1})$ and consider the switched system with the $n2^{n-d-1}$ matrices A_σ of the form be_i with $b \in \{1\} \times s_i \times \{-1, 1\}^{n-d-1}$ and $i \in \{1, \dots, n\}$ where s_i is the singleton containing the last d digits of i. The hypercube is invariant for the switched system but $\rho_{\text{SOS-1}}(G, \mathcal{A}) \geq \sqrt{2}$.

Proof. There are 2^{n-d-1} identical i_σ for each value of i_σ so the right-hand side of (5.12) is $n2^{n-d-1}I_n$. In the left-hand side of (5.12), there are $n/2^d$ identical terms $b_\sigma b_\sigma^\top$ for each value of b_σ. Applying Lemma 5.1.1 to each of the $n/2^d$ groups of different terms, the left-hand side sum up to $n2^{n-d-1}I_n$. Hence this is a feasible solution of Program 5.3.2 so $\rho_{\text{SOS-1}}(G, \mathcal{A}) \geq \sqrt{n}$. □

For $n = 2$, we find the same example as already exhibited by [AS98].

Example 5.1.3. For $n = 2$, the matrices considered in Theorem 5.1.2 are the matrices given in (5.5). The feasible dual solution is given by $i_1 = 1$ and $i_2 = 2$ and the constraint (5.11) becomes

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = 2(e_1 e_1^\top + e_2 e_2^\top)$$

For $n = 3$, the example given by [AS98] uses 3 complex matrices. With our approach, we obtain $3 \cdot 2^{3-1} = 12$ real matrices as 3 and 2^{3-1} are coprime. For $n = 4$, we have 8 different matrices as 8 is the least common multiplier of 4 and 2^{4-1}.

5.1.3 Conclusion

We showed that the bound (5.8) is tight in the real case as well as the complex case by exhibiting a switched system for any positive integer n that is stable but does not admit any invariant ellipsoid. The number of real matrices used is much larger than the number of complex matrices used in [AS98]. The question of whether there exists examples of stable switched systems that do not admit any invariant ellipsoid and are made of less matrices is open.

5.2 Entropy-based bound for constrained switched systems

5.2.1 Introduction

The two guarantees (5.6) and (5.7) are incomparable, as (5.6) depends on the dimension, and (5.7) depends on the number of matrices. However, only (5.6) has been generalized in the constrained case yet; see Theorem 5.2.4. The main result of this section is a generalization of the second guarantee: we relate the accuracy of the SOS-based approximation algorithm with the combinatorial complexity of the automaton. This complexity is measured by the *entropy* of the language of allowed switching signals. This new estimate of the accuracy of the SOS technique is always better than the previously existing one for sufficiently large sum-of-squares degree. According to the new estimate, the more constrained the system is, the smaller the entropy is and the better the

accuracy of the method is. This shows that, in some sense, it is easier to analyze stability of *constrained* switched systems than *unconstrained* switched systems because the entropy of the language of allowed switching signals is smaller.

Constrained switched systems may also be useful to analyze *abstraction techniques* for complex control systems. Given a nonlinear system, an *abstraction* of the system can be constructed by a discretization of the state-space, such abstraction may enhance our ability to analyze the system [Tab09]. The entropy of the language of allowed switching signals of the abstraction is related[2] to the *topological entropy* of the nonlinear system [AKK65; Bow71]. This suggests that the computational complexity of the abstraction is intrinsically related to the topological entropy of the nonlinear system and not to the specific choice of discretization, e.g. the value of ε. In [YM10], the authors use the Kullback-Leibler divergence of the uncertainty induced by a model to measure its fidelity. They measure the entropy of the *uncertainty* of the noise representing the part of the plant that is not accounted for in the model. This is similar to our work which measures the entropy of the *uncertainty* induced by an uncontrolled switching representing the loss of information due to the discretization. However, it is fundamentally different as we use this entropy to measure the computational complexity of the model and not the fidelity of the abstraction. Indeed, as we have seen, in our work this entropy is related to the topological entropy of the plant and not to the accuracy of the abstraction. Other appearances of the entropy in systems and control theory include [BGL01; FPR08]; see [PF13] for an overview.

5.2.2 Entropy

We denote the set of all words accepted by the automaton as $G^* = \bigcup_{k=1}^{\infty} G_k$. The entropy of a regular language is defined as follows.

Definition 5.2.1 (Entropy [LM95, Definition 4.1.1])**.** Given a regular language $\mathcal{L}$ recognized by an automaton G, we define the *entropy* of the language as

$$h(G) = \lim_{k\to\infty} \frac{1}{k} \log_2 |G_k|. \tag{5.13}$$

The entropy of a language generated by an automaton is easily computable, as we now recall. The logarithm of the spectral radius of the adjacency matrix of an *irreducible*[3] automaton gives the entropy of its *edge shift*.

[2] The entropy of the abstraction with an ε-discretization measures the growth rate of the number of cells in which the state could be [LM95, Example 6.3.4] while the topological entropy is the limsup, with $\varepsilon \to \infty$, of the growth rate with n of the cardinality of the largest (n, ε)-*separated* (or the smallest (n, ε)-*spanning*) set; see [Bow71] for precise definitions.

[3] An automaton is *irreducible* if for every pair of nodes u, v, there exists a path from u to v accepted by the automaton.

Definition 5.2.2 ([LM95, Definition 2.2.5])**.** The *edge shift* of an automaton $G = (V, E)$ is the language recognized by the automaton $G' = (E, E')$ with the transitions $((u, v, \sigma), (v, w, \sigma'), (v, w, \sigma')) \in E'$ for each $(u, v, \sigma), (v, w, \sigma') \in E$. We denote the entropy of the edge shift of G as $h(E) = h(G')$.

Particularizing equation (5.13) to the edge shift gives

$$h(E) = \lim_{k\to\infty} \frac{1}{k} \log_2 |E_k|. \tag{5.14}$$

It turns out that the entropy of the edge shift is equal to the entropy of the language recognized by the automaton if the automaton is *right-resolving* [LM95, Proposition 4.1.13].

Definition 5.2.3 ([LM95, Definition 3.3.1])**.** An automaton G is *right-resolving* if for every vertex v, the outgoing edges have different symbols.

Every regular language is recognized by a right-resolving automaton. Moreover, there are automated ways to obtain such an automaton from a starting representation of a language with an automaton that is not right-resolving [LM95, Section 3.3].

5.2.3 Constrained p-radius

The constrained p-radius is defined as follows.

Definition 5.2.4. The *constrained p-radius* of a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V, E)$, denoted as $\rho_p(G, \mathcal{A})$, is

$$\rho_p(G, \mathcal{A}) = \lim_{k\to\infty} \left[|E_k|^{-1} \sum_{v\in V} \hat{\rho}_{p;k;v}(G, \mathcal{A}) \right]^{\frac{1}{pk}}$$

where

$$\hat{\rho}_{p;k;v}(G, \mathcal{A}) = \sum_{s\in E_k^+(v)} \|A_s\|^p.$$

Thus, the CJSR can be defined as the constrained p-radius for $p = \infty$.

Remark 5.2.1. Since G is assumed to be strongly connected, we could give the following equivalent definition

$$\rho_p(G, \mathcal{A}) = \lim_{k\to\infty} \left[\max_{v\in V} [d_k^+(v)]^{-1} \hat{\rho}_{p;k;v}(G, \mathcal{A}) \right]^{\frac{1}{pk}} \tag{5.15}$$

or the same definition with "$d_k^-(v)$" instead of "$d_k^+(v)$" and "$s \in E_k^-(v)$" instead of "$s \in E_k^+(v)$" in the definition of $\hat{\rho}_{p;k;v}(G, \mathcal{A})$.

By the equivalence of norms, the definition of the p-radius does not depend on the norm used.

We can show that the p-radius is well defined using the following classical result, known as *Fekete's Lemma* [Fek23].

Lemma 5.2.1. Let $\{a_n\} : n \geq 1$ be a sequence of real numbers such that

$$a_{m+n} \leq a_m + a_n.$$

Then the limit

$$\lim_{n\to\infty} \frac{a_n}{n}$$

exists and is equal to $\inf\left\{\frac{a_n}{n}\right\}$.

Lemma 5.2.2. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V, E)$ and the sequence $(a_k)_k = \max_{v\in V} \hat{\rho}_{p;k;v}(G, \mathcal{A})$ with a submultiplicative norm. The sequence $\sqrt[k]{a_k}$ converges when $k \to \infty$. Moreover,

$$\lim_{k\to\infty} \sqrt[k]{a_k} = \inf\{\sqrt[k]{a_k}\}.$$

Proof. By submultiplicativity, for any $v \in V$, k and any $k_1, k_2 \geq 0$ such that $k_1 + k_2 = k$,

$$\begin{aligned}
\hat{\rho}_{p;k;v}(G, \mathcal{A}) &= \sum_{u\in V} \sum_{s_1\in E_{k_1}(v,u), s_2\in E^+_{k_2}(u)} \|A_{s_2}A_{s_1}\|^p \\
&\leq \sum_{u\in V} \sum_{s_1\in E_{k_1}(v,u), s_2\in E^+_{k_2}(u)} \|A_{s_2}\|^p \|A_{s_1}\|^p \\
&= \sum_{u\in V} \hat{\rho}_{p;k_2;u}(G, \mathcal{A}) \sum_{s_1\in E^-_{k_1}(u), s_1(1)=v} \|A_{s_1}\|^p \\
&\leq a_{k_2} \sum_{u\in V} \sum_{s_1\in E^-_{k_1}(u), s_1(1)=v} \|A_{s_1}\|^p \\
&\leq \hat{\rho}_{p;k_1;v}(G, \mathcal{A}) a_{k_2}
\end{aligned}$$

hence, in particular, $a_k \leq a_{k_1}a_{k_2}$ and $\log a_k \leq \log a_{k_1} + \log a_{k_2}$. We can conclude by Lemma 5.2.1. □

Corollary 5.2.1. The following holds

$$\lim_{k\to\infty} \left[\max_{v\in V} \hat{\rho}_{p;k;v}(G, \mathcal{A})\right]^{\frac{1}{k}} = \lim_{k\to\infty} \left[\sum_{v\in V} \hat{\rho}_{p;k;v}(G, \mathcal{A})\right]^{\frac{1}{k}}$$

and, in particular, the limit on the right-hand side converges.

Proof. For a finite set of nonnegative numbers, their maximum is always between their average and their sum:

$$\frac{1}{|V|}\sum_{v\in V}\hat{\rho}_{p;k;v}(G,\mathcal{A}) \le \max_{v\in V}\hat{\rho}_{p;k;v}(G,\mathcal{A}) \le \sum_{v\in V}\hat{\rho}_{p;k;v}(G,\mathcal{A})$$

or equivalently

$$\max_{v\in V}\hat{\rho}_{p;k;v}(G,\mathcal{A}) \le \sum_{v\in V}\hat{\rho}_{p;k;v}(G,\mathcal{A}) \le |V|\max_{v\in V}\hat{\rho}_{p;k;v}(G,\mathcal{A}).$$

By Lemma 5.2.2, $\max_{v\in V}\hat{\rho}_{p;k;v}(G,\mathcal{A})$ converges for $k\to\infty$ hence $\sum_{v\in V}\hat{\rho}_{p;k;v}(G,\mathcal{A})$ converges too. Taking the kth root and the limit $k\to\infty$ gives the identity. □

Theorem 5.2.1 shows a relation between entropy of the switching signals and the p-radius.

Theorem 5.2.1. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton G. The following relation holds

$$\rho_p(G,\mathcal{A}) = 2^{-h(E)/p}\lim_{k\to\infty}\left[\sum_{s\in E_k}\|A_s\|^p\right]^{\frac{1}{pk}}.$$

Proof. By Corollary 5.2.1,

$$\lim_{k\to\infty}\left[\sum_{s\in E_k}\|A_s\|^p\right]^{\frac{1}{pk}}$$

converges and by (5.14), $\lim_{k\to\infty}|E_k|^{-\frac{1}{pk}} = 2^{-h(E)/p}$. □

Let $x^{[d]}$ denote the *scaled monomial* basis. The elements of this basis are

$$\frac{d!}{\alpha_1!\alpha_2!\cdots\alpha_n!}x_1^{\alpha_1}\cdots x_n^{\alpha_n}.$$

for each n-tuples of nonnegative integers α such that $\alpha_1+\cdots+\alpha_n = d$. For this basis, $\|x^{[d]}\|_2 = \|x\|_2^d$ where $\|\cdot\|_2$ is the Euclidean norm.

For any matrix $A\in\mathbb{R}^{n\times n}$, the map $x\mapsto x^{[d]}$ induces an associated map $A^{[d]}\in\mathbb{R}^{N_d\times N_d}$ which is the unique matrix that satisfies $(Ax)^{[d]} = A^{[d]}x^{[d]}$. We also denote $\mathcal{A}^{[d]} \triangleq \{A_1^{[d]},\ldots,A_m^{[d]}\}$.

Since $\|Ax\|^{[d]} = \|A\|^{[d]}\|x\|^{[d]}$, we have the following Lemma that is known in the unconstrained case or for the contrained case with $p=\infty$.

Lemma 5.2.3. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton G, then

$$\rho_p(G, \mathcal{A}) = \rho_1(G, \mathcal{A}^{[p]})^{\frac{1}{p}}$$

and

$$\rho(G, \mathcal{A}) = \rho(G, \mathcal{A}^{[p]})^{\frac{1}{p}}.$$

We say that a cone $\mathcal{K}$ is *proper* if it is closed, solid, convex and pointed. We say that a matrix A *leaves a set* S *invariant* if $AS \subseteq S$ and we say that a set of matrices $\mathcal{A}$ *leaves a proper cone invariant* if there exists a proper cone $\mathcal{K}$ such that each matrix of $\mathcal{A}$ leaves $\mathcal{K}$ invariant.

Lemma 5.2.4 ([BN05; Pro97])**.** If a set of m matrices leaves a proper cone $\mathcal{K}$ invariant, then

$$\begin{aligned}\rho_1(\mathcal{A}) &= \frac{1}{m} \lim_{k\to\infty} \left\| \sum_{s\in[m]^k} A_s \right\|^{\frac{1}{k}} \\ &= \frac{1}{m} \rho\left(\sum_{A\in\mathcal{A}} A \right).\end{aligned}$$

We deduce the following corollary of Lemma 5.2.3 and Lemma 5.2.4.

Corollary 5.2.2. If $\mathcal{A}^{[p]}$ leaves a proper cone $\mathcal{K}$ invariant, then

$$\begin{aligned}\rho_p(\mathcal{A}) &= \frac{1}{m^{\frac{1}{p}}} \lim_{k\to\infty} \left\| \sum_{s\in[m]^k} A_s^{[p]} \right\|^{\frac{1}{pk}} \\ &= \frac{1}{m^{\frac{1}{p}}} \rho\left(\sum_{A\in\mathcal{A}} A^{[p]} \right)^{\frac{1}{p}}.\end{aligned}$$

We generalize it to the constrained case using the lifting procedure introduced independently by Kozyakin [Koz14] and Wang [Wan+14].

Lemma 5.2.5. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V, E)$. The following identity holds for any $p \in [1, +\infty]$

$$\sum_{s\in E_k} \|A_s\|^p = \sum_{s\in E^k} [\|A'_s\|']^p$$

where

$$\mathcal{A}' = \{ A'_{uv\sigma} = (e_v e_u^\top) \otimes A_\sigma \mid (u, v, \sigma) \in E \}.$$

Proof. Consider a vector norm $\|\cdot\|$ of $\mathbb{R}^n$ and the vector norm $\|\cdot\|'$ of $\mathbb{R}^{n|V|}$ such that

$$\|e_1 \otimes x_1 + \cdots + e_{|V|} \otimes x_{|V|}\| = \|x_1\| + \cdots + \|x_{|V|}\|.$$

Consider the induced matrix norms $\|\cdot\|$ and $\|\cdot\|'$. It is easy to see that for any nodes $u, v \in V$ and any matrix $B \in \mathbb{R}^{n\times n}$, $\|(e_v e_u^\top) \otimes B\|' = \|B\|$. In particular, given a path $s \in E_k$,

$$\begin{aligned}\|A'_s\|' &= \left\| \prod_{i=1}^{k} (e_{s(i+1)} e_i^\top) \otimes A_{s[i]} \right\|' \\ &= \|(e_{s(k+1)} e_{s(1)}^\top) \otimes A_s\|' = \|A_s\|\end{aligned}$$

and given $s \notin E_k$, $\|A'_s\| = 0$. □

It is easy to see that if $\mathcal{A}$ leaves the proper cone $\mathcal{K}$ invariant then the set of matrices $\mathcal{A}'$ of Lemma 5.2.5 leaves the proper cone $\mathcal{K}^{|V|}$ invariant.

Lemma 5.2.6. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V, E)$. If $\mathcal{A}$ leaves a proper cone invariant, then

$$\begin{aligned}\rho_1(G, \mathcal{A}) &= 2^{-h(E)/p} \lim_{k\to\infty} \left\| \sum_{s\in E_k} (e_{s(k+1)} e_{s(1)}^\top) \otimes A_s \right\|^{\frac{1}{k}} \\ &= 2^{-h(E)/p} \rho\left(\sum_{(u,v,\sigma)\in E} (e_v e_u^\top) \otimes A_\sigma \right).\end{aligned}$$

Proof. Combine Theorem 5.2.1, Lemma 5.2.5 and Lemma 5.2.4. □

Theorem 5.2.2. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton G. If $\mathcal{A}^{[p]}$ leaves a proper cone invariant, the following identities hold

$$\begin{aligned}\rho_p(G, \mathcal{A}) &= 2^{-h(E)/p} \lim_{k\to\infty} \left\| \sum_{s\in E_k} (e_{s(k+1)} e_{s(1)}^\top) \otimes A_s^{[p]} \right\|^{\frac{1}{pk}} \\ &= 2^{-h(E)/p} \rho\left(\sum_{(u,v,\sigma)\in E} (e_v e_u^\top) \otimes A_\sigma^{[p]} \right)^{\frac{1}{p}}.\end{aligned}$$

Theorem 5.2.2 shows that when there is an invariant proper cone, $\rho_p(G, \mathcal{A})$ is as easy to obtain as computing a spectral radius.

It turns out that if p is even then there exists an invariant proper cone.

Lemma 5.2.7. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton G. For any positive integer d, $\mathcal{A}^{[2d]}$ leaves an invariant proper cone. Moreover this cone is the cone of SOS polynomials in the scaled monomial basis.

Proof. Consider an homogeneous SOS polynomial $p(x)$ of degree $2d$ and its coordinates p in the scaled monomial basis. That is, $p(x) = \langle p, x^{[2d]} \rangle$. For any matrix A, we have

$$\langle A^{[2d]} p, x^{[2d]} \rangle = \langle p, (A^{[2d]})^\top x^{[2d]} \rangle = \langle p, (A^\top x)^{[2d]} \rangle = p(A^\top x).$$

Therefore if p is the coordinate vector of an SOS polynomial then $A^{[2d]} p$ is also the coordinate vector of an SOS polynomial. □

5.2.4 Performance guarantees

In this section, we provide a new bound that relates the accuracy of Program 3.2.2 to the entropy of the switching signal and the p-radius of the switched system.

An important property of the p-radius is that it is increasing in p.

Lemma 5.2.8. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton G. For any integers $p \leq q$,

$$\rho_p(G, \mathcal{A}) \leq \rho_q(G, \mathcal{A}) \leq \rho(G, \mathcal{A}) \\ \leq 2^{h(E)/q} \rho_q(G, \mathcal{A}) \leq 2^{h(E)/p} \rho_p(G, \mathcal{A}). \quad (5.16)$$

Proof. This is a consequence of Proposition 1.1.2 and Proposition 1.1.1. □

This Lemma is already known in the unconstrained case where $2^{h(E)} = m$ [Zho02].

Remark 5.2.2. Lemma 5.2.8 shows that the p-radius provides an upper and lower bound on the CJSR. See [BN05; PJ08] for methods based on the *veronese liftings* computing the $2d$-radius either by computing a spectral radius or by solving a linear program (see [OPJ16] for computation algorithms when p is not an even integer).

We show the following bound stating that the solution found by Program 3.2.2 is at least as good as the bound obtained by computing the $2d$-radius (see Lemma 5.2.8).

Theorem 5.2.3. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton G. For any positive integer d, the approximation given by Program 3.2.2 using homogeneous polynomials of degree $2d$ satisfies:

$$\rho_{\text{SOS-}2d}(G, \mathcal{A}) \leq 2^{h(E)/2d} \rho_{2d}(G, \mathcal{A}) \leq 2^{h(E)/2d} \rho(G, \mathcal{A}). \quad (5.17)$$

Note that the second inequality in (5.17) is simply (5.16). Theorem 5.2.3 is proven at the end of this section.

We can see with (5.17) that if $h(E) = 0$, the approximation is exact. This corresponds to the case where every node of G has indegree and outdegree 1. In that case, the graph forms a cycle of some length k and the CJSR is simply the kth root of the spectral radius of the product of the matrices along this cycle.

If the automaton $G(V, E)$ is not strongly connected then the entropy of the language recognized by G is equal to the maximum of the entropy of the language recognized by each of its strongly connected components. Similarly, the CJSR is equal to the maximum of the CJSR of each of the switched system constrained by each of the strongly connected components.

Corollary 5.2.3. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton G. Let $G_1(V_1, E_1), \ldots, G_m(V_m, E_m)$ be the strongly connected components of G. For any positive integer d, the approximation given by Program 3.2.2 using homogeneous polynomials of degree $2d$ satisfies:

$$\max_{i=1}^{m}(2^{-h(E_i)/2d}\rho_{\text{SOS-}2d}(G_2, \mathcal{A})) \leq \rho(G, \mathcal{A}). \tag{5.18}$$

As shown by the following example, the bound Theorem 5.2.3 can be improved when it is applied to each separate component.

Example 5.2.1. Consider the constrained switched system with $\mathcal{A} = \{A_1, A_2, A_3\}$ and automaton $G(V, E)$ where $V = \{1, 2\}$ and $E = \{(1, 1, 1), (1, 1, 2), (2, 2, 3)\}$. There are two strongly connected components $G_1(V_1, E_1)$ with $V_1 = \{1\}$ and $E_1 = \{(1, 1, 1), (1, 1, 2)\}$ and $G_2(V_2, E_2)$ with $V_2 = \{2\}$ and $E_2 = \{(2, 2, 3)\}$ and The entropy of the language recognized by G (resp. G_1, G_2) is 1 (resp. 1, 0). Applying Corollary 5.2.3 on each connected component gives

$$\max(\rho_{\text{SOS-}2d}(G_1, \mathcal{A})2^{\frac{-1}{2d}}, \rho_{\text{SOS-}2d}(G_2, \mathcal{A})) \leq \rho(G, \mathcal{A}) \leq \max(\rho_{\text{SOS-}2d}(G_1, \mathcal{A}), \rho_{\text{SOS-}2d}(G_2, \mathcal{A})).$$

If $\rho_{\text{SOS-}2d}(G_2, \mathcal{A})) \geq \rho_{\text{SOS-}2d}(G_1, \mathcal{A})$, then we can conclude that

$$\rho(G, \mathcal{A}) = \rho_{\text{SOS-}2d}(G_2, \mathcal{A}).$$

In general, if the lower bounds provided by a connected component is greater than the upper bound provided by a second connected component, we can deduce that the CJSR of this second connected component is not equal to the CJSR of the full constrained switched system. To refine the CJSR approximation, we can then for instance drop these connected components and solve Program 3.2.2 for larger degree only for the remaining ones.

For the unconstrained switching case, $2^{h(E)}$ is equal to the number of matrices m. Theorem 5.2.3 is therefore the generalization of (5.7) to the constrained case. A generalization of (5.6) to the constrained case was already known (note that the bound does not take into account the particular structure of the automaton):

Theorem 5.2.4 ([Phi+16, Theorem 3.6])**.** Consider a finite set of matrices $\mathcal{A} \subset \mathbb{R}^{n\times n}$ constrained by an automaton G and a positive integer d. The approximation $\rho_{\text{SOS-}2d}(G, \mathcal{A})$ given by Program 3.2.2 using homogeneous polynomials of degree $2d$ satisfies:

$$\rho_{\text{SOS-}2d}(G, \mathcal{A}) \leq \binom{n+d-1}{d}^{\frac{1}{2d}} \rho(G, \mathcal{A}).$$

The results of Theorem 5.2.3, Theorem 5.2.4 and (3.28) are summarized by the following corollary.

Corollary 5.2.4. Consider a finite set of matrices $\mathcal{A} \subset \mathbb{R}^{n\times n}$ constrained by an automaton G and a positive integer d, the approximation given by Program 3.2.2 using homogeneous polynomials of degree $2d$ satisfies:

$$\max\left\{\binom{n+d-1}{d}^{-\frac{1}{2d}}, 2^{-h(E)/2d}\right\}\rho_{\text{SOS-}2d}(G, \mathcal{A}) \leq \rho(G, \mathcal{A}) \leq \rho_{\text{SOS-}2d}(G, \mathcal{A}).$$

We see that we can have arbitrary accuracy by increasing d.

Our proof technique for Theorem 5.2.3 relies on the analysis of an iteration in the vector space of polynomials of degree $2d$. When this iteration converges, it converges to a feasible solution of Program 3.2.2. By analyzing this iteration as affine iterations in this vector space, we derive a sufficient condition for its convergence and thus an upper bound for $\rho_{\text{SOS-}2d}(G, \mathcal{A})$.

Consider the iteration

$$\begin{aligned} p_{v,0}(x) &= 0, \\ p_{v,k+1}(x) &= q_v(x) + \frac{1}{\tau}\sum_{(u,v,\sigma)\in E} p_{u,k}(A_\sigma x), \quad v \in V \end{aligned} \tag{5.19}$$

for fixed homogeneous polynomials $q_v(x)$ of degree $2d$ in n variables (not necessarily different) and a constant $\tau > 0$.

When this iteration converges, it converges to a feasible solution of Program 3.2.2.

Lemma 5.2.9. Consider a constant $\tau > 0$. If there exist homogeneous polynomials $q_v(x)$ in the interior of the SOS cone such that iteration (5.19) converges then $\rho_{\text{SOS-}2d}(G, \mathcal{A}) \leq \tau^{\frac{1}{2d}}$.

Proof. Suppose the iteration converges to the polynomials $p_{v,\infty}(x)$. It is easy to show by induction that $p_{v,k}(x)$ is SOS for all k. It is trivial for $k = 0$ and if it is true for k then it is also true for $k + 1$ by (5.19). Since the SOS cone is closed, $p_{v,\infty}$ is SOS. Now by (5.19), for each $v \in V$,

$$p_{v,\infty}(x) = q_v(x) + \frac{1}{\tau} \sum_{(u,v,\sigma)\in E} p_{u,\infty}(A_\sigma x)$$

so $p_{v,\infty}(x)$ is also in the interior of the SOS cone. For each edge (u, v, σ), by manipulating the above equation, we have

$$\tau p_{v,\infty}(x) - p_{u,\infty}(A_\sigma x) = \tau q_v(x) + \sum_{\substack{(u',v,\sigma')\in E,\\ (u',\sigma')\neq(u,\sigma)}} p_{u',\infty}(A_{\sigma'} x)$$

so $\tau p_{v,\infty}(x) - p_{u,\infty}(A_\sigma x)$ is SOS. Therefore $(\{\, p_{v,\infty}(x) \,:\, v \in V \,\}, \tau^{\frac{1}{2d}})$ is a feasible solution of Program 3.2.2. □

In view of Lemma 5.2.9, it is thus natural to analyze under which condition iteration (5.19) converges.

Proof of Theorem 5.2.3. Iteration (5.19) is an affine map on the vector space of homogeneous polynomials of degree $2d$. It is well known that if the convergence is guaranteed when we only retain the linear part of the affine map then it is also guaranteed for the affine iteration.

Therefore we can analyze instead the following iteration

$$\begin{aligned} p_{v,0}(x) &= q_v(x), \\ p_{v,k+1}(x) &= \frac{1}{\tau} \sum_{(u,v,\sigma)\in E} p_{u,k}(A_\sigma x), \quad v \in V \end{aligned}$$

We can see that

$$p_{v,k}(x) = \frac{1}{\tau^k} \sum_{s\in E_k^-(v)} q_{s(1)}(A_s x)$$

where $s(1)$ denotes the first node of the path s.

Consider a norm $\|\cdot\|$ of $\mathbb{R}^n$ and its corresponding induced matrix norm of $\mathbb{R}^{n\times n}$. For each $v \in V$, we know by continuity of $q_v(x)$ that there exist $\beta_v > 0$

such that $q_v(x) \leq \beta_v \|x\|^{2d}$ for all $x \in \mathbb{R}^n$. Let $\beta = \max_{v \in V} \beta_v$, then

$$\begin{aligned} p_{v,k}(x) &\leq \frac{1}{\tau^k} \sum_{s \in E_k^-(v)} \beta_{s(1)} \|A_s\|^{2d} \|x\|^{2d} \\ &\leq \frac{\beta}{\tau^k} \|x\|^{2d} \sum_{s \in E_k^-(v)} \|A_s\|^{2d} \\ \sum_{v \in V} p_{v,k}(x) &\leq \frac{\beta}{\tau^k} \|x\|^{2d} \sum_{s \in E_k} \|A_s\|^{2d} \end{aligned}$$

By Theorem 5.2.1, if $\tau > 2^{h(E)} \rho_{2d}(G, \mathcal{A})^{2d}$, then $\lim_{k \to \infty} \sum_{v \in V} p_{v,k}(x) = 0$ hence $\lim_{k \to \infty} p_{v,k}(x) = 0 \ \forall v \in V$ since the polynomials $p_{v,k}$ belong to a proper cone. We obtain the result by Lemma 5.2.9. □

5.2.5 Improving the automaton-dependent bounds

If strong duality holds for a convex problem, its feasibility is equivalent to the non-existence of an *infeasibility certificate* (see [BV04, Section 5.8]). An infeasibility certificate contains one entry per constraint and if this entry is zero for a given constraint then the infeasibility certificate remains valid if the constraint is removed from the problem. In this section, we show how this fact allows to improve the guarantee given by Theorem 5.2.3 using the sparsity of the infeasibility certificate.

Lemma 5.2.10 (No duality gap)**.** For a fixed γ,

Weak duality If Program 3.2.1 (resp. Program 5.3.1) is feasible for $\overline{\gamma} = \gamma$ (resp. $\underline{\gamma} = \gamma$) then Program 5.3.1 (resp. Program 3.2.1) is infeasible for all $\underline{\gamma} < \gamma$ (resp. $\overline{\gamma} > \gamma$).

Strong duality If Program 3.2.1 (resp. dual) is infeasible for $\overline{\gamma} = \gamma$ (resp. $\underline{\gamma} = \gamma$) then Program 5.3.1 (resp. Program 3.2.1) is feasible for $\underline{\gamma} = \gamma$ (resp. $\overline{\gamma} = \gamma$).

In other words, there exists a value γ^* such that for every $\gamma > \gamma^*$, there exists a feasible solution to Program 3.2.1 and for every $\gamma < \gamma^*$, there exists a feasible solution to Program 5.3.1. Moreover, either Program 3.2.1, Program 5.3.1 or both have a feasible solution with $\gamma = \gamma^*$.

Proof. Consider the hyperplane $C \triangleq \left\{ (f_v : v \in V) \in \mathcal{F}^{|V|} \,\middle|\, \sum_{v \in V} \int_{\mathbb{S}^{n-1}} f_v(x) \, \mathrm{d}x = 1 \right\}$ and the map $\mathcal{D}_\gamma \colon \mathcal{F}^{|V|} \to \mathcal{F}^{|E|} : (f_v : v \in V) \mapsto (\gamma f_u(x) - f_v(A_\sigma x) : (u, v, \sigma) \in E)$.

Given a fixed γ, Program 3.2.1 has no solution for $\overline{\gamma} = \gamma$ if and only if $\mathcal{D}_\gamma(\mathcal{F}_{++}^{|V|} \cap C) \cap \mathcal{F}_+^{|E|} = \emptyset$. Since $\mathcal{F}_{++}^{|V|} \cap C$ is compact, so is $\mathcal{D}_\gamma(\mathcal{F}_{++}^{|V|} \cap C))$.

We know that a compact set and a closed set have no intersection if and only if there exist a strict separating hyperplane separating the two sets. That is, a measure $\mu \in \mathcal{M}$ such that $\langle \mu, f \rangle \geq 0$ for all $f \in \mathcal{F}_+^{|E|}$ and $\langle \mu, f \rangle < 0$ for all $f \in \mathcal{D}_\gamma(\mathcal{F}_{++}^{|V|} \cap C)$. The first condition is simply $\mu \in \mathcal{M}_+$. For the second condition, we remark that $\mathcal{D}_\gamma(\mathcal{F}_{++}^{|V|} \cap C) = \mathcal{D}_\gamma(\operatorname{int}(\mathcal{F}_+^{|V|}) \cap C) = \operatorname{ri} \mathcal{D}_\gamma(\mathcal{F}_+^{|V|} \cap C)$ where ri denotes the *relative interior* of a set. We have $\langle \mu, f \rangle < 0$ for all $f \in \operatorname{ri} \mathcal{D}_\gamma(\mathcal{F}_+^{|V|} \cap C)$ if and only if $\langle \mu, f \rangle \leq 0$ for all $f \in \mathcal{D}_\gamma(\mathcal{F}_+^{|V|} \cap C)$ and

$$\exists f \in \mathcal{D}_\gamma(\mathcal{F}_+^{|V|} \cap C) : \langle \mu, f \rangle \neq 0. \tag{5.20}$$

Therefore, if Program 3.2.1 has no solution for $\overline{\gamma} = \gamma$ then there exists a *nonzero* measure $\mu \in (\mathcal{M}_+)^{|E|}$ such that for all $f \in C$ and $(u, v, \sigma) \in E$,

$$\sum_{v \in V} \sum_{(v,u,\sigma) \in E} \bar{\gamma} \mathbb{E}_{vu\sigma}[f_v(x)] \leq \sum_{v \in V} \sum_{(u,v,\sigma) \in E} \mathbb{E}_{uv\sigma}[f_v(A_\sigma x))] \tag{5.21}$$

and (5.20) holds.

Note that if the inequality (5.21) is respected for some $f \in C$, it is also respected for λf for all $\lambda > 0$. So we can impose that the inequality should be respected for all $f \in \mathcal{F}_+^{|V|} \setminus \{0\}$.

The constraint (5.21) must be true for all $f \in \mathcal{F}_+^{|V|} \setminus \{0\}$ so in particular in the case where there is a node $v \in V$ such that $f_u(x) = 0$ for all $u \neq v$. Therefore we must have

$$\gamma \sum_{(v,u,\sigma) \in E} \mathbb{E}_{vu\sigma}[f_v(x)] \leq \sum_{(u,v,\sigma) \in E} \mathbb{E}_{uv\sigma}[f_v(A_\sigma x))], \quad \forall f_v \in \mathcal{F}_+$$

for all $v \in V$. This is (5.24) so the strong duality is proven.

To show the weak duality, we show that if there exists a dual solution μ for $\underline{\gamma} = \gamma$ then (5.24) and (5.20) are satisfied for all $\underline{\gamma} < \gamma$. We know that (5.24) is satisfied for γ so the constraint (5.24) is also satisfied for any $\underline{\gamma} < \gamma$. Using (5.21) and (5.25) with $f_v(x) = \|x\|$ for all $v \in V$, we have $\langle \mu, f \rangle < 0$ for all $\underline{\gamma} < \gamma$.

□

We show in Lemma 5.2.10 that strong duality holds for Program 3.2.2 with a fixed $\overline{\gamma}$. This allows Program 3.2.2 to be solved by binary search on $\overline{\gamma}$: Given a fixed value γ, the problem is solved with $\overline{\gamma} = \gamma$; if a feasible solution is found, it means that $\overline{\gamma}^\star \leq \gamma$, otherwise, an infeasibility certificate is found showing that $\overline{\gamma}^\star \geq \gamma$. By Corollary 5.2.4, an infeasibility certificate for γ provides the following lower bound certificate on the CJSR:

$$\max\left\{ \binom{n+d-1}{d}^{-\frac{1}{2d}}, 2^{-h(E)/2d} \right\} \gamma \leq \rho(G, \mathcal{A}).$$

In Theorem 5.2.5 we show a simple way to improve this lower bound certificate by inspecting the sparsity of the infeasibility certificate.

Definition 5.2.5. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V, E)$. Given an infeasibility certificate $\widetilde{\mu}$ of Program 3.2.2, we denote by $E_{\widetilde{\mu}}$ the set of edges $e \in E$ such that the entry of $\widetilde{\mu}$ corresponding to constraint (3.25) with edge e is nonzero.

Theorem 5.2.5. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V, E)$. For any positive integer d, if there exists an infeasibility certificate $\widetilde{\mu}$ of Program 3.2.2 with $\overline{\gamma} = \gamma$ then

$$2^{-h(E_{\widetilde{\mu}})/2d}\gamma \leq \rho(G, \mathcal{A}). \tag{5.22}$$

Proof. We consider the graph $G_{\widetilde{\mu}}(V, E_{\widetilde{\mu}})$. Since the infeasibility certificate $\widetilde{\mu}$ is zero for constraints (3.25) with edges $e \in E \setminus E_{\widetilde{\mu}}$, $\widetilde{\mu}$ remains a valid infeasibility certificate for Program 3.2.2 with input $(G_{\widetilde{\mu}}, \mathcal{A})$ and $\overline{\gamma} = \gamma$, hence $\gamma \leq \rho_{\text{SOS-}2d}(G_{\widetilde{\mu}}, \mathcal{A})$. By Theorem 5.2.3, $2^{-h(E_{\widetilde{\mu}})/2d}\rho_{\text{SOS-}2d}(G_{\widetilde{\mu}}, \mathcal{A}) \leq \rho(G_{\widetilde{\mu}}, \mathcal{A})$ and since $E_{\widetilde{\mu}} \subseteq E$, $\rho(G_{\widetilde{\mu}}, \mathcal{A}) \leq \rho(G, \mathcal{A})$. We obtain (5.22) by combining these three inequalities. □

Example 5.2.2. Applying the result of this section to the running example gives the result of Figure 5.1. The "Kronecker lift" lower bound is the bound obtained by using the *Kronecker lift* to transform the constrained system with 9 edges into an unconstrained system with 9 matrices, one per edge. The upper bound obtained with both systems is the same [Phi+16, Proposition 3.9] hence we can use the guarantee for unconstrained systems (5.7) with $m' = |E| = 9$ for the constrained system.

The entropy of the switching signal $h(E)$ used in Theorem 5.2.3 is $\log_2(2.61803)$, while the value $\binom{n+d-1}{d}$ used in Theorem 5.2.4 is $d + 1$ since $n = 2$. Therefore, as we can see on the figure, the lower bound guaranteed by Theorem 5.2.4 is more accurate for $d = 1$ only. The entropy $h(E_{\widetilde{\mu}})$ used in Theorem 5.2.5 is $\log_2(1.61803)$ for $d = 1, 2$ and $\log_2(1.83929)$ for $d = 3, 4, 5, 6$, it is more accurate than the three other lower bounds for every d.

The lower bound obtained by computing the $2d$-radius is the most accurate one among all lower bounds for the same d for this example. In practice, better lower bounds can be obtained from the solution of Program 3.2.2 using the techniques of [LJP16; LPJ17].

5.3 Certifying lower bounds

Certifying lower bounds $\underline{\gamma}$ is currently either achieved using the guarantees we have on the accuracy of the upper bound on the JSR or by exhibiting trajectories of asymptotic growth rate $\underline{\gamma}$. In this section, we introduce a new

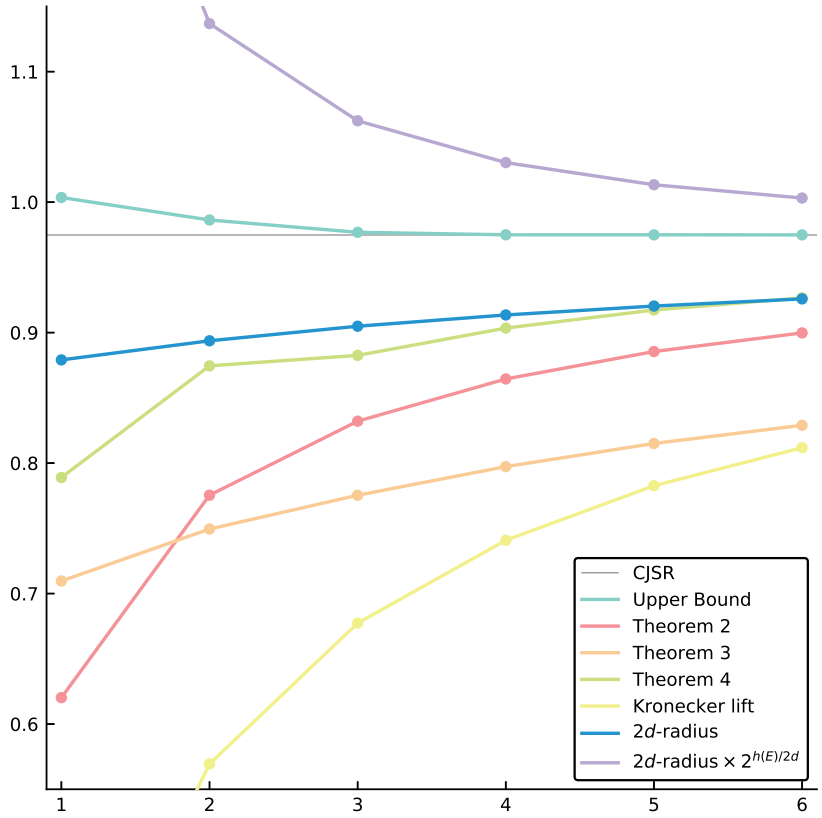


Figure 5.1: Result of Example 5.2.2 for $d = 1, 2, 3, 4, 5, 6$; the value of d is given in the horizontal axis. The exact value of the CJSR found in [LJP16] is represented by the horizontal line. The upper and lower bounds given by Lemma 5.2.8 using the $2d$-radius are denoted "$2d$-radius". The upper bound found by Program 3.2.2 with polynomials of degree $2d$ is denoted "Upper Bound". From this upper bound, three lower bounds can be obtained using Theorem 5.2.3, Theorem 5.2.4 and Theorem 5.2.5. A fourth lower bound can be obtained using the "Kronecker lift" as explained in Example 5.2.2.

way to certify lower bounds by exhibiting nonnegative measures satisfying some invariance condition parametrized by the matrices $A_i/\underline{\gamma}$; see (5.24). This invariance condition is linear on the measure hence the search of measures on the convex cone of nonnegative measures is a *convex* program; see Program 5.3.1. It turns out that this program is the dual of Program 3.2.1.

We revisit the sum-of-squares program proposed by Parrilo and Jadbabaie [PJ08] and show that its dual formulation is the moment relaxation of the search of the measures satisfying the invariance condition.

Thanks to this duality, solving this pair of programs with a given candidate value γ for the JSR either returns Lyapunov functions certifying that

$\rho(\mathcal{A}) \leq \gamma$ or returns moments that form a solution of the moment relaxation; see Section 2.3.2. These moments are not necessarily the moments of measures satisfying the invariance conditions. However, we give a rounding procedure to extract a (infinite) switching sequence from these moments and provide a guarantee on the asymptotic growth rate of this sequence. As a by-product of the rounding procedures, the spectral radius of a finite part of this infinite sequence can be used to give lower bounds on the JSR. In addition, we give a way to sometimes detect when the moments belongs to measures that satisfy the invariance conditions. This happens when the measures are the convex combination of the occupation measures of several periodic trajectories. Since the trajectories are periodic, the measures are atomic and we can recover them from moments of sufficiently high degree. We show on numerical examples that these techniques work well in practice.

A popular method for proving stability of a dynamical system is to find a Lyapunov function. In this section, we introduce measures playing a role dual to Lyapunov function for switched system. These measures provide a certificate for instability. Finding Lyapunov functions and finding these measures are in fact two dual programs, they are respectively provided by Program 3.2.1 and Program 5.3.1. We will be succinct in our definition of measure-theoretic concepts but the interested reader can find a good introduction to writing programs using measures and functions as decision variables in [Las09].

Consider the dual pair $(\mathcal{B}, \mathcal{M})$ where $\mathcal{M}$ is the space of *finite*[4] *signed*[5] Borel measures on $\mathbb{S}^{n-1}$ and the scalar product between a function $f \in \mathcal{B}$ and a measure $\mu \in \mathcal{M}$ is $\langle f, \mu \rangle = \int f \, \mathrm{d}\mu$. We define the scalar product for the space $\mathcal{F}$ defined in (3.22) with $\langle h(f), \mu \rangle = \langle f, \mu \rangle$ for $f \in \mathcal{B}, \mu \in \mathcal{M}$.

Given an application A and a measure $\mu \in \mathcal{M}$, the *pushforward measure* $A\#\mu$ is often defined to be the measure given by $(A\#\mu)(B) = \mu(A^{-1}(B))$ for $B \in \mathbb{S}^{n-1}$. However, since $\mathbb{S}^{n-1}$ may not be invariant under application of the matrices of $\mathcal{A}$, we will use an alternative definition. Given an application A and a measure μ, the pushforward measure $A\#\mu$ is defined to be the measure such that $\langle f, A\#\mu \rangle = \langle f \circ A, \mu \rangle$ for any $f \in \mathcal{F}$. Moreover, given $B \subseteq \mathbb{S}^{n-1}$, we define $\mu(B) = \langle h(\mathbf{1}_B), \mu \rangle$ so that $(A\#\mu)(B)$ is well defined. Using these definitions, one can verify that for any application A and measure $\mu \in \mathcal{M}$,

$$(A\#\mu)(\mathbb{S}^{n-1}) \leq \mu(\mathbb{S}^{n-1}) \max_{x \in \mathrm{supp}(\mu)} \|Ax\|_2 \tag{5.23}$$

where $\mathrm{supp}(\mu)$ is the support of μ.

Let $\mathcal{M}_+$ be the set of (nonnegative) measures of $\mathcal{M}$. Given two measures $\mu, \nu \in \mathcal{M}$, $\mu \geq 0$ denotes $\mu \in \mathcal{M}$ and $\mu \geq \nu$ denotes $\mu - \nu \in \mathcal{M}_+$.

[4] The measure μ is *finite* if $\mu(\mathbb{S}^{n-1})$ is finite.

[5] A *signed* measure is a difference between two measures, i.e. $\mu - \nu$ where μ and ν are measures is a signed measure.

The dual of Program 3.2.1 is:

Program 5.3.1 (Dual of Program 3.2.1)**.** *Input*: A finite set of matrices $\mathcal{A}$ and an automaton G.

Output: Measures $\mu_{uv\sigma}$ and a number $\underline{\gamma}$.

$$\sup_{\mu_{uv\sigma}\in\mathcal{M},\underline{\gamma}\in\mathbb{R}} \underline{\gamma}$$

$$\text{subject to} \sum_{(u,v,\sigma)\in E} A_\sigma \# \mu_{uv\sigma} \geq \underline{\gamma} \sum_{(v,w,\sigma)\in E} \mu_{vw\sigma}, \quad \forall v \in V, \tag{5.24}$$

$$\mu_{uv\sigma} \in \mathcal{M}_+, \quad \forall (u,v,\sigma) \in E,$$

$$\sum_{(u,v,\sigma)\in E} \mu_{uv\sigma}(\mathbb{S}^{n-1}) = 1. \tag{5.25}$$

The constraint (3.23) is the Lyapunov constraint. The constraint (5.24) is similar to the *measure invariance constraint* $A\#\mu = \mu$ of a linear dynamical system $x_{k+1} = Ax_k$ and to the *mass balance constraint* of a *circulation problem* [AMO93]. Without constraint (3.24) (resp. (5.25)), the feasible set of Program 3.2.1 (resp. Program 5.3.1) is a cone. These constraints have no effect on the optimal objective value but they make the feasible set bounded.

The main result of this section is summarized in the following theorem.

Theorem 5.3.1. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton G. Let $\overline{\gamma}^\star$ (resp. $\underline{\gamma}^\star$) be the optimal value of Program 3.2.1 (resp. Program 5.3.1). The following identity holds:

$$\underline{\gamma}^\star = \rho(G,\mathcal{A}) = \overline{\gamma}^\star.$$

As a consequence of Theorem 5.3.1, we have a new criterion for lower bounds on the CJSR using measures.

Corollary 5.3.1. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V,E)$. If there exist non-trivial[6] measures $\mu_{uv\sigma}$ for each $(u,v,\sigma) \in E$ such that

$$\sum_{(u,v,\sigma)\in E} A_\sigma \# \mu_{uv\sigma} \geq \underline{\gamma} \sum_{(v,w,\sigma)\in E} \mu_{vw\sigma}, \quad \forall v \in V$$

then $\underline{\gamma} \leq \rho(G,\mathcal{A})$.

The following lemma shows a recursive way to build an optimal solution of Program 3.2.1.

[6] At least one $\mu_{uv\sigma}$ must be nonzero.

Lemma 5.3.1. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V, E)$. For any natural number k and norm $\|\cdot\|$, we have

$$\overline{\gamma}^\star \leq \hat{\rho}_k(G, \mathcal{A}, \|\cdot\|)$$

where $\hat{\rho}_k(G, \mathcal{A}, \|\cdot\|)$ is defined in (3.21).

Proof. Let $A'_\sigma = A_\sigma / \hat{\rho}_k(G, \mathcal{A}, \|\cdot\|)$, $f_v(x) = \max_{s \in \cup_{i=0}^{k-1} E_i^+(v)} \|A'_s x\|$. For any edge $(u, v, \sigma) \in E$,

$$\begin{aligned} f_v(A'_\sigma x) &= \max\Big(\max_{s \in E_{k-1}^+(v)} \|A'_s A'_\sigma x\|, \max_{s \in \cup_{i=0}^{k-2} E_i^+(v)} \|A'_s A'_\sigma x\| \Big) \\ &\leq \max\Big(\|x\|, \max_{s \in \cup_{i=1}^{k-1} E_i^+(u)} \|A'_s x\| \Big) = f_u(x) \end{aligned}$$

so the Lyapunov functions f_v are solution for $\overline{\gamma} = \hat{\rho}_k(G, \mathcal{A}, \|\cdot\|)$. □

Proof of Theorem 5.3.1. By Lemma 5.2.10, we have $\underline{\gamma}^\star = \overline{\gamma}^\star$, by Theorem 3.2.2, we have $\rho(G, \mathcal{A}) \leq \overline{\gamma}^\star$ and by Lemma 5.3.1 and (3.19), we have $\overline{\gamma}^\star \leq \rho(G, \mathcal{A})$. □

The following lemma illustrates the relation between atomic solutions of Program 5.3.1 and periodic trajectories. Lemma 5.3.1 and Lemma 5.3.2 somehow suggest that Program 3.2.1 is related to the definition (3.21) of the CJSR with norms while Program 5.3.1 is related to the definition (3.20) of the CJSR with the spectral radius.

Lemma 5.3.2. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton G and a cycle $c = (\sigma_1, \ldots, \sigma_k)$ of length k with intermediary nodes $v_0, \ldots, v_{k-1}, v_k = v_0 \in V$ such that $(v_{i-1}, v_i, \sigma_i) \in E$ for $i = 1, \ldots, k$. Let $x_0 \in \mathbb{R}^n$ and $\lambda > 0$ be such that $A_c x_0 = \lambda x_0$ with $\|x_0\|_2 = 1$, consider the following iteration

$$x_i = A_{\sigma_i} x_{i-1} \qquad \hat{x}_i = x_i / \|x_i\|_2 \qquad \alpha_i = \|x_i\|_2 / \lambda^{i/k}$$

The following solution

$$\Big(\mu_{uv\sigma} = \sum_{i=1, v_i = v}^{k} \alpha_i \delta_{\hat{x}_i} \Big)_{(u,v,\sigma) \in E}$$

is feasible for Program 5.3.1 with any $\underline{\gamma} \geq \lambda^{1/k}$ and it satisfies the constraints (5.24) as equality for $\underline{\gamma} = \lambda^{1/k}$.

Proof. By construction, $\alpha_k = 1$ so $\alpha_k \delta_{\hat{x}_k} = \delta_{x_0}$ and for each $i = 0, \dots, k-1$, we have

$$A_{\sigma_i} \# (\alpha_i \delta_{\hat{x}_i}) = \alpha_i \frac{\|x_{i+1}\|_2}{\|x_i\|_2} \delta_{\hat{x}_{i+1}} = \lambda^{1/k} \alpha_{i+1} \delta_{\hat{x}_{i+1}} \leq \underline{\gamma} \alpha_{i+1} \delta_{\hat{x}_{i+1}}$$

which equality if $\lambda^{1/k} = \underline{\gamma}$. □

In some sense, Lemma 5.3.2 is encoding a trajectory in the measures $\mu_{uv\sigma}$. We say that the resulting measures are the *occupation measures* of the trajectory $x_0, x_1, \dots, x_k$ defined in Lemma 5.3.2.

Example 5.3.1. Consider the unconstrained system [AP12b, Example 2.1] with $m = 2$:

$$\mathcal{A} = \{A_1 = e_1 e_2^\top, A_2 = e_2 e_1^\top\}$$

where e_i denotes the ith canonical basis vector.

A solution to Program 3.2.1 is given by $(f(x), \overline{\gamma}) = (\|x\|_2, 1)$. This means that $\|x\|_2$ is a Lyapunov function for the system so as it is well known this certifies that $\rho(\mathcal{A}) \leq 1$.

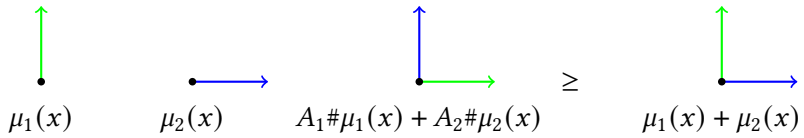


Figure 5.2: A representation of the optimal dual solution of Example 5.3.1 with the constraint (5.24)**.**

A dual solution μ_1 (resp. μ_2)[7] for the first (resp. second) matrix has the measure $\mu_1 = \delta_{(0,1)}/2$ (resp. $\mu_2 = \delta_{(1,0)}/2$). This is the solution obtained by applying Lemma 5.3.2 to the cycle $(1, 2)$. This is shown in Figure 5.2.

Remark 5.3.1. Occupation measures for continuous switched systems are studied in [CDH16]. These measures are supported on the cartesian product of the state space and a finite interval of time $t \in [0, T]$ while in this section, the measures are only supported on the subset $\mathbb{S}^{n-1}$ of the state space. Indeed, since the system (3.11) is homogeneous and time-invariant, we can encode trajectories in a measure on $\mathbb{S}^{n-1}$ (Lemma 5.3.2) and still be able to recover it (Corollary 5.3.1).

The measures studied in [Har+11] are supported on the paths in G. They are related to the measures studied in this section since given a cycle c, we can compute the occupation measures of the trajectory using this switching cycle and starting with a leading eigenvector of A_c as x_0 with Lemma 5.3.2.

[7] In the arbitrary switching case, we write μ_σ instead of $\mu_{uv\sigma}$ for short

One may wonder whether Lemma 5.3.2 also works in the reverse direction to give a *constructive* proof for Corollary 5.3.1 when the measures $\mu_{uv\sigma}$ are atomic. Namely, can we extract a periodic trajectory of period c with $\rho(c) \geq \underline{\gamma}$ from any atomic feasible solution of Program 5.3.1 with $\underline{\gamma}$. As such solution may be the convex hull of solutions obtained by the construction of Lemma 5.3.2, we may recover several periodic trajectory, from which there might be only one that satisfies $\rho(c) \geq \underline{\gamma}$. The following Lemma provides a constructive way to recover a periodic trajectory of period c satisfying $\rho(c) \geq \underline{\gamma}$ in the scalar case[8], i.e. $n = 1$

Lemma 5.3.3. Consider a finite set of matrices $\mathcal{A} \subseteq \mathbb{R}^{1\times 1}$ constrained by an automaton G. If there exists a feasible solution μ of Program 5.3.1 with $\underline{\gamma}$, then there exists a cycle c with $\rho(c) \geq \underline{\gamma}$.

Proof. Let (μ, γ) be the solution. By (5.25) and (5.24), we can find a cycle c for which each edge e has a nonzero measure μ_e.

If $\rho(c) \geq \gamma$, we are done. Otherwise, if $\rho(c) < \gamma$, using Lemma 5.3.2, we can build a feasible solution ν such that (5.24) is satisfied with equality for $\underline{\gamma} = \rho(c)$. This means that $\mu - \lambda\nu$ is feasible with γ for any $\lambda \geq 0$ such that $\mu - \lambda\nu \geq 0$. Let $\lambda^\star$ be the maximum value of λ such that $\mu - \lambda\nu \geq 0$. Since $n = 1$, $\mathbb{S}^{n-1}$ is zero dimensional so for at least one edge e of the cycle c, $\mu_e - \lambda^\star \nu_e$ is zero. Moreover, since μ_e is nonzero for all edge e of the cycle, $\lambda > 0$. Therefore, the number of edges with nonzero measure has decreased and at least one of the constraints (5.24) is now satisfied with strict inequality.

This process can only be repeated finitely many times until μ becomes the trivial solution since the number of edges with nonzero measure decrease each time. Moreover we will have $\rho(c) \geq \gamma$ at least once since the constraints (5.24) cannot be satisfied with strict inequality for the trivial solution. □

Given a feasible solution of Program 5.3.1 and a common partition of the support of the measures, we show in Proposition 5.3.1 how to transform the solution into a solution of a scalar switched system. Using this transformation, we can always recover a cycle c for which $\rho(c) = \gamma$ from a solution of Program 5.3.1 with $\underline{\gamma} = \gamma$ for which the measures are atomic.

Proposition 5.3.1. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V, E)$. Suppose that there exists a feasible solution μ of Program 5.3.1 with $\underline{\gamma} = \gamma$ and a finite family $\mathcal{S}$ of disjoint subsets of $\mathbb{S}^{n-1}$ such that the support of each measure is included in the union of the sets of the family $\mathcal{S}$. Then there exists sets $B_1, \ldots, B_k \in \mathcal{S}$ and a cycle $\sigma_1, \ldots, \sigma_k$ of G

[8]Note that in this case, any measure is atomic since $\mathbb{S}^{n-1}$ is zero-dimensional

such that

$$\prod_{i=1}^{k} \max_{x \in B_i} \|A_{\sigma_i} x\|_2 \geq \gamma^k$$

and $A_{\sigma_i} B_i \cap B_{i+1} \neq \emptyset$ for $i = 1, \ldots, k$ where $B_{k+1} = B_1$.

Proof. Given a set $B \in \mathcal{S}$ and an edge $e \in E$, let μ_e^B denote the measure defined as $\mu_e^B(C) = \mu_e(C \cap B)$. We consider a new constrained switched system with matrices $\mathcal{A}' \subseteq \mathbb{R}^{1\times 1}$ and automaton $G'(V', E')$ where $V' = \{(v, B) \mid v \in V, B \in \mathcal{S}\}$, $e'((u, v, \sigma), B, C) = ((u, B), (v, C), (\sigma, B))$, $E' = \{e'(e, B, C) \mid e \in E, B, C \in \mathcal{S}, A_e B \cap C \neq \emptyset\}$, and $A'_{(\sigma,B)} = \max_{x \in B} \|A_\sigma x\|_2$. From any solution μ of the original system feasible for $\underline{\gamma}$, the following solution of the system with matrices $\mathcal{A}'$ and automaton G'

$$\mu'_{e'(e,B,C)} = \frac{(A_e \# \mu_e^B)(C)}{(A_e \# \mu_e^B)(\mathbb{S}^{n-1})} \mu_e(B)$$

is also feasible with $\underline{\gamma}$. Indeed, by construction, for any $v \in V, C \in \mathcal{S}$, we have

$$\begin{aligned} \sum_{e \in E_1^-(v), B \in \mathcal{S}} (A_e \# \mu_e^B)(C) &= \sum_{e \in E_1^-(v), B \in \mathcal{S}} \mu'_{e'(e,B,C)} \frac{(A_e \# \mu_e^B)(\mathbb{S}^{n-1})}{\mu_e(B)} \\ &\overset{(5.23)}{\leq} \sum_{e \in E'^-_1(v,C)} A'_e \# \mu'_e \end{aligned} \tag{5.26}$$

$$\begin{aligned} \sum_{e \in E_1^+(v)} \mu_e(C) &= \sum_{e \in E_1^+(v), D \in \mathcal{S}} \frac{(A_e \# \mu_e^C)(D)}{(A_e \# \mu_e^C)(\mathbb{S}^{n-1})} \mu_e(C) \\ &= \sum_{e \in E'^+_1(v,C)} \mu'_e. \end{aligned} \tag{5.27}$$

By (5.24) on μ, the left-hand side of (5.27) is smaller than the left-hand side of (5.26). Therefore, the right-hand side of (5.27) is smaller than the right-hand side of (5.26) hence μ' satisfies (5.24) on the new switched system.

Therefore, by Lemma 5.3.3, there is a cycle $(\sigma_1, B_1), \ldots, (\sigma_k, B_k)$ of G' such that the modes σ_i and sets B_i are as required. □

Example 5.3.2. Consider the dual solution obtained in Example 5.3.1.

The supports of μ_1 and μ_2 are respectively $B_1 = \{(0, 1)\}$ and $B_2 = \{(1, 0)\}$. The automaton $G'(V', E')$ obtained by the transformation of Proposition 5.3.1 is defined by $V' = \{(1, B_1), (1, B_2)\}$ and $E' = \{((1, B_1), (1, B_2), (1, B_1)), ((1, B_2), (1, B_1), (2, B_2))\}$. The new 1×1 matrices are $A'_{(1,B_1)} = 1$ and $A'_{(2,B_2)} = 1$.

The computation of the CJSR of this scalar system is a *maximum cycle mean* problem as outlined in [AP12b]. The cycle of maximum geometric mean is $((1, B_1), (2, B_2))$ which geometric mean $\sqrt{1 \cdot 1} = 1$. We recover the cycle $(1, 2)$ found in Example 5.3.1.

5.3.1 Dual SOS program

In Section 3.2.3, we introduced the SOS restriction of Program 3.2.1 with Program 3.2.2. In Section 5.3.1, we introduce Program 5.3.2, the moment relaxation of Program 5.3.1. It turns out that Program 3.2.2 and Program 5.3.2 are dual to each other. Indeed, the proof of Lemma 5.2.10 can be translated verbatim in order to prove that Program 5.3.2 is the dual of Program 3.2.2.

Program 5.3.2 (Dual of Program 3.2.2)**.** *Input*: A finite set of matrices $\mathcal{A}$ and an automaton G.

Output: Pseudo-measures $\widetilde{\mu}_{uv\sigma}$ and a number $\underline{\gamma}$.

$$\sup_{\widetilde{\mu}_{uv\sigma}\in\mathbb{R}^*_{\mathbf{2d}},\underline{\gamma}\in\mathbb{R}} \underline{\gamma}$$

$$\text{subject to} \sum_{(u,v,\sigma)\in E} A_\sigma \# \widetilde{\mu}_{uv\sigma} - \underline{\gamma}^{2d} \sum_{(v,w,\sigma)\in E} \widetilde{\mu}_{vw\sigma} \in \Sigma^*_{\mathbf{2d}}, \quad \forall v \in V, \tag{5.28}$$

$$\widetilde{\mu}_{uv\sigma} \in \Sigma^*_{\mathbf{2d}}, \quad \forall (u,v,\sigma) \in E, \tag{5.29}$$

$$\sum_{(u,v,\sigma)\in E} \widetilde{\mu}_{uv\sigma}(\mathbb{S}^{n-1}) = 1. \tag{5.30}$$

It is important to note that a solution of Program 5.3.2 is not necessarily a solution of Program 5.3.1. First $\widetilde{\mu}_{uv\sigma}$ may not be a measure even if it belongs to $\Sigma^*_{\mathbf{2d}}$ as discussed in Section 2.3.2. Second, the left-hand side of (5.28) may also not be a measure. For this second concern, it helps to be more explicit. Suppose for instance that we are in the quadratic case, i.e. $d = 1$. In that case, if $\widetilde{\mu} \in \Sigma^*_2$, there always exists a measure μ that has the moments of the pseudo-measure $\widetilde{\mu}$. We can take for instance a Gaussian distribution with these second order moments. Hence we can find Gaussian distributions $\mu_{uv\sigma}$ that have the second order moments $\widetilde{\mu}_{uv\sigma}$ and Gaussian distributions ν_v that have the second order moments given by the left-hand side of (5.28). However, we may have

$$\sum_{(u,v,\sigma)\in E} A_\sigma \# \mu_{uv\sigma} - \underline{\gamma}^{2d} \sum_{(v,w,\sigma)\in E} \mu_{vw\sigma} \neq \nu_v$$

as we only know that the left-hand side and right-hand side of the above equation have the same second order moments; see Example 5.3.5.

However, in some cases, we can recover a feasible solution of Program 5.3.1 from a feasible solution of Program 5.3.2. In these cases, by Corollary 5.3.1, this provides a lower bound on the CJSR. Moreover, there exist efficient techniques allowing to detect situations where the solution is moments of an atomic measure; see [HL05; Lau09]. Then, using the transformation of Proposition 5.3.1, we can transform these atomic measures into a feasible solution

of a constrained scalar switched systems. For such system, we could use the algorithm described in Lemma 5.3.3 but as pointed out in [AP12b], computing the CJSR of a scalar system can easily be done by solving a maximum cycle mean problem for which efficient algorithm exists [Kar78].

If we recover a feasible solution of Program 5.3.1 from a feasible solution of Program 5.3.2 with $\gamma = \rho_{\text{SOS-}2d}(G, \mathcal{A})$, we can directly conclude that $\rho_{\text{SOS-}2d}(G, \mathcal{A}) = \rho(G, \mathcal{A})$. This is somewhat similar to the minimization of a multivariate polynomial using SOS where we can detect that we have reached the optimum when the measure is atomic and recover the minimizers of the polynomial from the atoms of the measure.

However, we may also check for atomic feasible solutions of Program 5.3.1 with $\underline{\gamma} < \rho_{\text{SOS-}2d}(G, \mathcal{A})$ to provide lower bounds. Moreover, in practice, $\rho_{\text{SOS-}2d}(G, \mathcal{A})$ is computed by binary search on $\underline{\gamma}$ so we often have several such solutions.

Example 5.3.3. Consider Example 3.2.1. For $i = 1, 2, 3$, let $\widetilde{\mu}_i$ be the solution of Program 5.3.2 corresponding to the matrix A_i. For any d, we can see that the dual solution for $\gamma = 1$ is such that the only monomial x^α such that $\langle \widetilde{\mu}_1, x^\alpha \rangle$ (resp. $\langle \widetilde{\mu}_2, x^\alpha \rangle$, $\langle \widetilde{\mu}_3, x^\alpha \rangle$) is non-zero is x_1^{2d} (resp. x_2^{2d}, x_3^{2d}) and $\langle \widetilde{\mu}_1, x_1^{2d} \rangle = \langle \widetilde{\mu}_2, x_2^{2d} \rangle = \langle \widetilde{\mu}_3, x_3^{2d} \rangle = 1/3$. Note that it means that $\widetilde{\mu}_1 = \delta_{(1,0,0)}/3$, $\widetilde{\mu}_2 = \delta_{(0,1,0)}/3$ and $\widetilde{\mu}_3 = \delta_{(0,0,1)}/3$ where δ_x is the Dirac measure centered on x. Since these measures are solution to Program 5.3.1 with $\underline{\gamma} = 1$, by Corollary 5.3.1, this means that $\rho(\mathcal{A}) \geq 1$.

Example 5.3.4. We consider [PJ08, Example 2.8]:

$$A_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \qquad A_2 = \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix}.$$

to illustrate the fact that this atom extraction pocedure can be used to determine when the upper bound found by Program 3.2.2 is equal to the CJSR. In this unconstrained example, the JSR is one but the upper bound found by Program 3.2.2 for $d = 1$ is $\sqrt{2}$. However, for $d = 2$, the upper bound found is 1 and the solutions of Program 5.3.2 for $\gamma = 1$ is

$$\mu_1 = 0.59698\delta_{(1,1)} + 0.59513\delta_{(1,-1)} \qquad \mu_2 = 0.59513\delta_{(1,1)} + 0.59322\delta_{(1,-1)}.$$

Since $A_1 \# \delta_{(1,1)} = \delta_{(1,1)}$, the cycle extraction method immediately find the cycle $c = 1$ for which $\rho(A_c) = 1$.

Example 5.3.5. We continue the running example; see Example 3.1.1 and Example 3.2.2.

For all d, $\widetilde{\mu}_{212} = \widetilde{\mu}_{323} = \widetilde{\mu}_{344} = \widetilde{\mu}_{431} = 0$ hence the node 4 is "unused" by the dual. For $2d = 2, 4$, $\widetilde{\mu}_{123} = \widetilde{\mu}_{231} = 0$ so the node 2 is "unused" for low degree.

At first, one could think that the dual variables can be used to reduce the systems, e.g. remove nodes or edges. However, it would be a mistake to remove the node 2 since the periodic trajectory with highest growth rate uses this node.

It is also interesting to notice that the matrices corresponding to the dual variables have low rank. For example, for $2d = 2$, $\widetilde{\mu}_{131}$ (resp. $\widetilde{\mu}_{312}$, $\widetilde{\mu}_{331}$) is the Dirac measure $5.873 \cdot \delta_{(0.917,0.399)}$ (resp. $3.966 \cdot \delta_{(0.875,0.485)}$, $6.704 \cdot \delta_{(0.757,-0.653)}$). However, this is not a feasible solution of Program 5.3.1. Indeed, while (5.24) is satisfied for node 1 since $A_2 \# \delta_{(0.875,0.485)}$ gives $\delta_{(0.917,0.399)}$, (5.24) is not satisfied for node 3 as $A_1 \# \delta_{(0.917,0.399)}$ gives $\delta_{(0.999,-0.0271)}$ and $A_1 \# \delta_{(0.757,-0.653)}$ gives $\delta_{(0.422,-0.906)}$.

5.3.2 Constructing high growth sequence

In this section we give an algorithm that generates an infinite sequence of matrices such that the asymptotic growth rate of the product of the matrices is arbitrarily close to the CJSR. Note that by Definition 3.2.1, this asymptotic growth rate must be smaller than the CJSR.

Given an edge $e \in E$, let $\widetilde{\mathbb{E}}_e[p(x)] = \langle \widetilde{\mu}_e, p(x) \rangle$. Given a polynomial $p_0(x) \in \operatorname{int}(\Sigma_{\mathbf{2d}})$ and an initial edge (v_0, v_{-1}, σ_0), the algorithm builds a $G^\top$-admissible sequence $(v_1, v_0, \sigma_1), (v_2, v_1, \sigma_2), \ldots$ such that

$$\theta_k \triangleq \widetilde{\mathbb{E}}_{v_k v_{k-1} \sigma_k}[p_0(A_{\sigma_1} \cdots A_{\sigma_k} x)] \tag{5.31}$$

remains "large" for increasing k. As we will see, using Lemma 5.3.6, this implies that $A_{\sigma_1} \cdots A_{\sigma_k}$ has a "large" norm.

Lemma 5.3.4. For any matrix $A \in \mathbb{R}^{n \times n}$ and symmetric positive semidefinite matrix Q, the following inequality holds

$$\rho(A^\top Q A) \leq \rho(Q)\rho(A^\top A).$$

Proof. Since Q is symmetric positive semidefinite, it has a Cholesky decomposition $Q = R^\top R$. Therefore we have

$$\rho(A^\top Q A) = \rho(A^\top R^\top R A) = \|RA\|^2 \leq \|R\|^2 \|A\|^2 = \rho(Q)\rho(A^\top A).$$

□

Lemma 5.3.5 ([LJP16, Lemma 6])**.** For any polynomial $p(x) \in \operatorname{int}(\Sigma_{\mathbf{2d}})$, there exists a constant $\beta > 0$ such that for any matrix A,

$$\beta \|A\|_2^{2d} p(x) - p(Ax) \quad \text{is SOS}$$

where $\|A\|_2 = \rho(A^\top A)^{1/2}$ is the Euclidean norm. Moreover we can choose $\beta = \kappa(Q)$ where $Q \in \mathcal{S}_+^n$ is a symmetric matrix such that $p(x) = (x^{[d]})^\top Q x^{[d]}$ for all $x \in \mathbb{R}^n$ and $\kappa(Q) = \rho(Q)\rho(Q^{-1})$ is the condition number of Q.

Proof. Consider the matrix Q defined in the statement of the lemma. Note that since $p(x) \in \operatorname{int}(\Sigma_{2\mathbf{d}})$, a positive definite Q such that $p(x) = (x^{[d]})^\top Q x^{[d]}$ exists. We can see that

$$((Ax)^{[d]})^\top Q (Ax)^{[d]} = (x^{[d]})^\top (A^{[d]})^\top Q A^{[d]} x^{[d]}.$$

Moreover, for all $x \in \mathbb{R}^n$,

$$(x^{[d]})^\top Q x^{[d]} \geq \|x^{[d]}\|_2 / \rho(Q^{-1})$$

and by Lemma 5.3.4,

$$(x^{[d]})^\top (A^{[d]})^\top Q A^{[d]} x^{[d]} \leq \rho(Q)\rho((A^{[d]})^\top A^{[d]}) \|x^{[d]}\|_2.$$

Using the fact that $\rho(A^{[d]}) = \rho(A)^d$ and $(AB)^{[d]} = A^{[d]} B^{[d]}$ for any matrix A and B, we have

$$\kappa(Q)\rho(A^\top A)^d Q - (A^{[d]})^\top Q A^{[d]} \geq 0.$$

□

Lemma 5.3.6. Let us consider a solution $(\widetilde{\mu}_e : e \in E)$ of Program 5.3.2. For any polynomial $p(x) \in \operatorname{int}(\Sigma_{2\mathbf{d}})$, there exists a positive constant τ such that for any matrix $A \in \mathbb{R}^{n \times n}$ and edge $e \in E$,

$$\widetilde{\mathbb{E}}_e[p(Ax)] \leq \tau \|A\|_2^{2d}$$

Proof. If all pseudo-expectations are zero, the result is trivially true. Therefore we can suppose that at least one is nonzero. By Lemma 5.3.5, there exists a constant $\beta > 0$ such that

$$\beta \|A\|_2^{2d} p(x) - p(Ax) \text{ is SOS.}$$

Hence for any edge $e \in E$,

$$\widetilde{\mathbb{E}}_e[p(Ax)] \leq \beta \|A\|_2^{2d} \widetilde{\mathbb{E}}_e[p(x)].$$

We obtain the result with the constant $\tau = \beta \max_{e \in E} \widetilde{\mathbb{E}}_e[p(x)]$. Since at least one pseudo-expectation is nonzero and $p(x)$ is in the interior of the SOS cone, $\tau > 0$. □

Lemma 5.3.8 provides a guarantee on the growth rate of θ_k, defined in (5.31), using the dual constraint (5.28).

Algorithm 4 Generates a sequence of large asymptotic growth.

Data: Length of subpaths: $l \in \mathbb{N}$; degree: $d \in \mathbb{N}$; and lower bound to $\rho_{\text{SOS-}2d}(G, \mathcal{A})$: $0 < \gamma < \rho_{\text{SOS-}2d}(G, \mathcal{A})$.
Result: Sequence of arbitrary length $s = (\ldots, v_k, \sigma_k, \ldots, v_0, \sigma_0, v_{-1})$.
Given a feasible solution $(\widetilde{\mu}_e : e \in E)$ of Program 5.3.2 with $\underline{\gamma} = \gamma$ and degree d
Pick an arbitrary polynomial $p_0(x) \in \operatorname{int}(\Sigma_{\mathbf{2d}})$
Pick an edge $(v_0, v_{-1}, \sigma_0) \in E$ such that $\widetilde{\mu}_{v_0 v_{-1} \sigma_0}$ is nonzero
for $k = 0, l, 2l, \ldots$ **do**
  Pick $s \in \arg\max_{s \in E_l^-(v_k)} \widetilde{\mathbb{E}}_{s[1]}[p_k(A_s x)]$
  Set $(v_{k+l}, \sigma_{k+l}, \ldots, \sigma_{k+1}, v_k) \leftarrow s$
  Set $p_{k+l} \leftarrow p_k(A_s x)$
end for

Lemma 5.3.7. Given a finite set of matrices $\mathcal{A}$ constrained by an automaton G, if $\widetilde{\mu}$ is a feasible solution of Program 5.3.2 then, for any edge $(\bar{u}, \bar{v}, \bar{\sigma}) \in E$, the following holds:

$$\sum_{s \in E_k^-(\bar{u})} A_s \# \widetilde{\mu}_{s[1]} \succeq \gamma^{2dk} \widetilde{\mu}_{\bar{u}\bar{v}\bar{\sigma}} \tag{5.32}$$

where $\widetilde{\mu}_1 \succeq \widetilde{\mu}_2$ denotes $\widetilde{\mu}_1 - \widetilde{\mu}_2 \in \Sigma^*_{\mathbf{2d}}$.

Proof. We prove (5.32) by induction, the case of $k = 0$ being trivial. Suppose that

$$\sum_{s' \in E_{k-1}^-(\bar{u})} A_{s'} \# \widetilde{\mu}_{s'[1]} \succeq \gamma^{2d(k-1)} \widetilde{\mu}_{\bar{u}\bar{v}\bar{\sigma}}. \tag{5.33}$$

We can rewrite the left-hand side of (5.32) as

$$\sum_{s \in E_k^-(\bar{u})} A_s \# \widetilde{\mu}_{s[1]} = \sum_{s' \in E_{k-1}^-(\bar{u})} A_{s'} \# \sum_{(u, s'(1), \sigma) \in E} A_\sigma \# \widetilde{\mu}_{u s'(1) \sigma}. \tag{5.34}$$

By (5.28),

$$\sum_{(u, s'(1), \sigma) \in E} A_\sigma \# \widetilde{\mu}_{u s'(1) \sigma} \succeq \gamma^{2d} \sum_{(s'(1), w, \sigma') \in E} \widetilde{\mu}_{s'(1) w \sigma'}.$$

Since the dual variables $\widetilde{\mu}_{s'(1) w \sigma'}$ of the right-hand side are in the dual of the SOS cone, and one of them is $\widetilde{\mu}_{s'[1]}$, we have

$$\sum_{(u, s'(1), \sigma) \in E} A_\sigma \# \widetilde{\mu}_{u s'(1) \sigma} \succeq \gamma^{2d} \widetilde{\mu}_{s'[1]}.$$

Applying $A_{s'}\#$ on both sides and using (5.34) gives

$$\sum_{s\in E_k^-(\bar{u})} A_s\#\widetilde{\mu}_{s[1]} \succeq \gamma^{2d} \sum_{s'\in E_{k-1}^-(\bar{u})} A_{s'}\#\widetilde{\mu}_{s'[1]} \overset{(5.33)}{\succeq} \gamma^{2dk}\widetilde{\mu}_{\bar{u}\bar{v}\bar{\sigma}}.$$

□

Lemma 5.3.8. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V,E)$. For any positive integers d and l, using Program 5.3.2 with any $\gamma < \rho_{\text{SOS-}2d}$
$(G,\mathcal{A})$, Algorithm 4 with paths of length l produces a $G^\top$-admissible sequence $(v_1,v_0,$
$\sigma_0),(v_2,v_1,\sigma_1),\ldots$ for which the sequence of θ_k defined in (5.31) satisfies the following inequality for all $k>0$ multiple of l:

$$\theta_k \geq \frac{\gamma^{2dl}}{d_l^-(v_{k-l+1})}\theta_{k-l}$$

Proof. By Lemma 5.3.7,

$$\sum_{s\in E_l^-(v_{k-l+1})} \widetilde{\mathbb{E}}_{s[1]}[p_{k-l}(A_s x)] \geq \gamma^{2dl}\theta_{k-l}.$$

Since the value of s chosen by Algorithm 4 maximises $\widetilde{\mathbb{E}}_{s[1]}[p_{k-l}(A_s x)]$, the left-hand side of the above inequality is smaller or equal to $d_l^-(v_{k-l+1})\theta_k$. □

Theorem 5.3.2 translates the guarantee on θ_k to a guarantee on $A_{\sigma_1}\cdots A_{\sigma_k}$ using Lemma 5.3.6.

Theorem 5.3.2. Consider a finite set of matrices $\mathcal{A}$ constrained by an automaton $G(V,E)$. For any positive integers d,l and a lower bound $\gamma < \rho_{\text{SOS-}2d}(G,\mathcal{A})$, Algorithm 4 with input l,d and γ produces a $G^\top$-admissible sequence $(v_1,v_0,\sigma_0),(v_2,v_1,\sigma_1),\ldots$ that satisfies the following inequality:

$$\lim_{k\to\infty} \|A_{s_k}\|_2^{\frac{1}{k}} \geq \frac{\gamma}{(\Delta_l^-(G))^{\frac{1}{2dl}}}$$

where $s_k=(\sigma_k,\ldots,\sigma_1)$.

Proof. By Lemma 5.3.8, for any k multiple of l,

$$\widetilde{\mathbb{E}}_{s_k[1]}[p_0(A_{s_k}x)] \geq \frac{\gamma^{2dk}}{(\Delta_l^-(G))^{\frac{k}{l}}}\widetilde{\mathbb{E}}_{v_0v_{-1}\sigma_0}[p_0(x)]$$

By Lemma 5.3.6, there exists a constant $\tau>0$ such that

$$\widetilde{\mathbb{E}}_{s_k[1]}[p_0(A_{s_k}x)] \leq \tau\|A_{s_k}\|^{2d}.$$

Combining these two inequalities, we obtain

$$\tau \|A_{s_k}\|^{2d} \geq \frac{\gamma^{2dk}}{(\Delta_l^-(G))^{\frac{k}{l}}} \widetilde{\mathbb{E}}_{v_0 v_{-1} \sigma_0}[p_0(x)].$$

Since $\widetilde{\mathbb{E}}_{v_0 v_{-1} \sigma_0}$ is nonzero, $\widetilde{\mathbb{E}}_{v_0 v_{-1} \sigma_0}[p_0(x)] > 0$. Therefore taking the $(2dk)$th root and the limit $k \to \infty$ we obtain the result. □

Example 5.3.6. Suppose that we apply Algorithm 4 with $l = 1$ to Example 5.3.3 and let us denote by c_α the coefficient of the monomial x^α in the polynomial $p_0(x)$ chosen arbitrarily by the algorihtm. The start of the sequence produced depends on the order between the coefficients $c_{(2d,0,0)}, c_{(0,2d,0)}, c_{(0,0,2d)}$. If $c_{(2d,0,0)}$ is the largest then the G-admissible left-infinite sequence found is $\ldots, 1, 2, 3, 1, 2, 3, 1, 2, 3$.

The product $A_{\sigma_1} A_{\sigma_2} A_{\sigma_3} \cdots = A_3 A_2 A_1 A_3 A_2 A_1 \cdots$ is periodic and has an asymptotic growth rate $\rho(A_{\sigma_1} A_{\sigma_2} A_{\sigma_3})^{1/3} = 1$. Hence $1 \leq \rho(G, \mathcal{A})$.

5.3.3 Deducing a lower bound certificate

By definition of the CJSR, the asymptotic growth rate of the norm of the product of any G-admissible (or $G^\top$-admissible) sequence of matrices gives a lower bound on the CJSR. In particular the sequence produced by Algorithm 4 provides a lower bound on the CJSR.

If there are two integers $\bar{k}, k$ such that the sequence after $\bar{k}$ is periodic of period k, the asymptotic growth rate of the norm is equal to the kth root of the spectral radius of the product of the matrices of one period. This is due to the Gelfand's formula $\rho(A) = \lim_{k \to \infty} \|A^k\|^{1/k}$. From the same identity, we see that the spectral radius of the product of the matrices of one G-admissible cycle gives a lower bound on the CJSR.

To find lower bounds for the CJSR, one could generate all the cycles of length smaller than some maximum length and compute the spectral radius for all of them. This brute force approach is not scalable because the number of paths considered grows exponentially with the maximum length.[9]

Gripenberg [Gri96] proposes a branch-and-bound algorithm that prunes the search using an a priori fixed absolute error. Other alternative methods exist such as the balanced complex polytope algorithm [GZ08; GP13] and the invariant conitope algorithm [JCG14]. The methods attempts to generate an invariant polytope from the eigenvector of a cycle of high growth rate. A candidate of cycle of higher growth rate can be found while constructing this polytope, the construction is then restarted with its eigenvector as a new

[9] The exponential growth of the brute force approach is the reason why one should choose a small l for Algorithm 4.

stating point. While computing this polytope, convexity arguments allows to prune paths which attenuate the exponential growth of the number of paths. Specialized methods exist for some particular matrix structures such as the "spectral simplex method" [Pro16] in the case of nonnegative matrices with a "product structure".

These algorithms can also be used to produce a G-admissible sequence of matrices of high asymptotic growth rate by reproducing the cycles of high spectral radius infinitely. The advantage of Algorithm 4 is that it provides a guarantee of accuracy given in Theorem 5.3.2. Algorithm 4 provides at the same time a high growth infinite trajectory and lower bounds of guaranteed accuracy.

Algorithm 4 requires to solve a semidefinite program with semidefinite matrices of size $\binom{n+d-1}{d}$. Then, in order to add l new edges to the sequence, it needs to go through $\Delta_l^-(G)$ paths and compare them by computing the scalar product between a polynomial and moments with $\binom{n+2d-1}{2d}$ monomials. The semidefinite program can be solved in a time polynomial in $\binom{n+d-1}{d}$ and $|E|$ as detailed in Section 2.2, and adding l edges to the sequence can be done in a time proportional to $\Delta_l^-(G)\binom{n+2d-1}{2d}$. While polytopes are used in [GZ08; GP13; JCG14] to prune paths, Algorithm 4 uses a solution of Program 5.3.2 to guide the search which enables the discovery of sequences of guaranteed high growth rate even with a small value of l. Moreover, Example 5.3.7 and Section 5.4.3 give examples where Algorithm 4 uncovers rather long cycles of high asymptotic growth rate. This shows the complementarity of Algorithm 4 with existing approaches which performs better when the cycles of high growth rate have a small length as they iterate over possible cycles of increasing length (although some are pruned). Moreover, [Gom+18a] shows that the algorithms can handle constrained switched systems with automaton of large size, as it stabilizes a system with 64 nodes and 512 edges [Gom+18a, Table I].

Example 5.3.7. We consider the switched system introduced in [BTV03] as a counterexample to the finiteness conjecture [LW95]. We use the value $\alpha = 0.7493265463303675$ which is the IEEE double-precision number that is closest to the value given in [Har+11] for which the system does not satisfy the finiteness property. Table 5.1 gives two cycles of high growth rate; as the reader can check, their growth rates are rather close. We verified that there exists no cycle of length up to 32 that provides a larger lower bound.

The polysets corresponding to the optimal solution of Program 3.2.2 for $2d = 2, 4, \ldots, 30$ are given in Figure 5.3. Using Algorithm 4 with $2d = 2$, $l = 4$ and p_0 equal to the solution of Program 3.2.2, the algorithm generates a sequence starting with the cycle of length 21 in Table 5.1.

We consider now the Balanced Polytope algorithm exploiting the nonneg-

length	cycle	growth rate
13	2112112121121	1.4092472220583443
21	211211212112112121121	1.4092472220583487

Table 5.1: Two cycles of high growth rate for the switched system of Example 5.3.7.

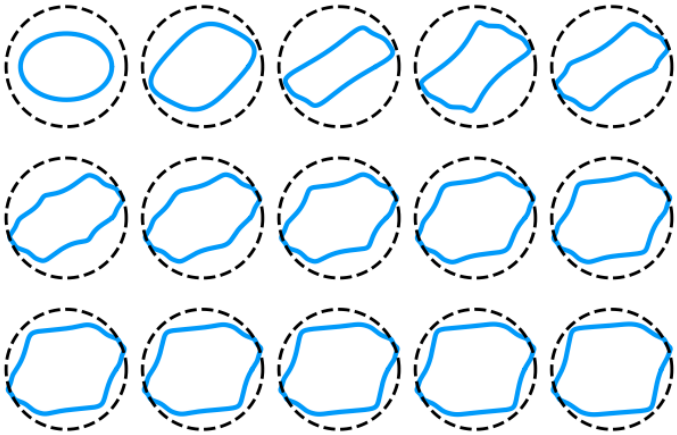


Figure 5.3: Illustration of the polysets corresponding to the optimal solution of Program 3.2.2 for Example 5.3.7 and for different degree $2d$. In the first row, from left to right, it corresponds to $2d = 2, 4, 6, 8, 10$. In the second row, from left to right, it corresponds to $2d = 12, 14, 16, 18, 20$. In the third row, from left to right, it corresponds to $2d = 22, 24, 26, 28, 30$.

ativity of the matrices [GP13, Section 4]. A point p is considered to belong to the interior of a balanced polytope P if MOSEK [ApS19] with its dual simplex algorithm certifies that the maximal t such that $tp \in P$ is larger than $1 + 3 \times 10^{-13}$. The algorithm first finds the cycle of length 13 in Table 5.1 and is then able to find the cycle of length 21 with the tolerance 3×10^{-13}. However, if the 3×10^{-13} tolerance is replaced by -1×10^{-12} or lower, then the algorithm does not find this second cycle. This behavior is not surprising given how close the growth rates are as shown in Table 5.1.

A cycle with growth rate equal to the CJSR is often called *spectral maximizing product* (s.m.p.). The algorithm is able to conclude that the cycle of length 21 is an s.m.p. with the tolerance 3×10^{-13}. If we replace the 3×10^{-13} by 4×10^{-13}, it does not seem to terminate as the number of leaves in the tree increases with the depth considered. We will consider this cycle to be an s.m.p. for the purpose of the benchmark even though it cannot be said for certain.

We can compute "non-constructive" lower bounds (it is not constructive as it does not exhibit a cycle certifying the lower bound) using the guarantee (given in [LJP16, Corollary 1]) on the upper bound (3.28) provided by Program 3.2.2, but in practice the trajectories found by Algorithm 4 are periodic after some time $\bar{k}$ so we are able to compute much better lower bounds than the pessimistic bound provided by the guarantee. This is shown by Example 5.3.8.

Example 5.3.8. We tried the atom extraction procedure introduced in Section 5.3.1 and Algorithm 4 for $l = 1$ and $l = 3$ on our running example; see Example 3.1.1, Example 3.2.2 and Example 5.3.5. The result is shown in Figure 5.4. We showed in [LJP16] that the CJSR of the system is equal to 0.97482. We can see that this lower bound is found for $d = 4$ for $l = 1$ and for $d = 1$ for $l = 3$. The atom extraction finds the lower bound 0.939255.

Example 5.3.9. Consider the unconstrained switched system with the following two matrices[10]:

$$A_1 = \begin{bmatrix} -1 & 1 & -1 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} -1 & 1 & -1 \\ -1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

The s.m.p. has length 41 and growth rate 1.684185:

$$11122112211221122112211221122112211221112.$$

[10]This example was found in a previous collaboration with N. Guglielmi and A. Cicone (unpublished).

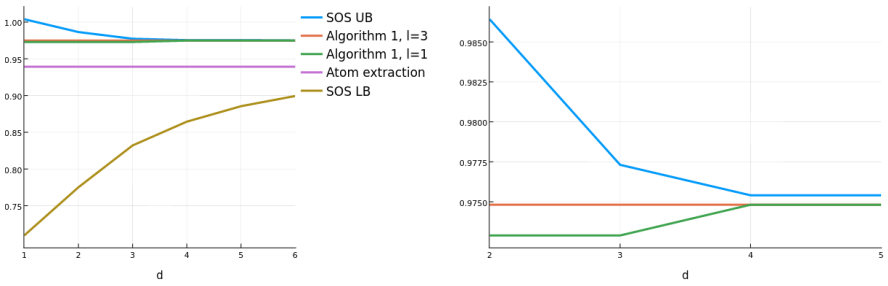


Figure 5.4: Result of Example 5.3.8. The SOS UB is the upper bound found by Program 3.2.2 and the SOS LB is obtained from its guarantee; see [LJP16, Corollary 1]. The value d of horizontal axis corresponds to using polynomials of degree $2d$. The right figure is a zoom of the left figure.

We summarize in Table 5.2 and Table 5.3 the time taken by the different methods on the examples. As we can see in Table 5.2, the time taken by Algorithm 4 to find the s.m.p. is competitive compared to alternative approach once the Sum-of-Squares pair of primal-dual programs Program 3.2.2/Program 5.3.2 has been solved. Moreover, as we can see in Table 5.3, finding upper bounds by solving this pair of programs is competitive with alternative approaches.

5.3.4 Conclusion

We have analyzed the dual of the set program for polysets of degree $2d$ and shown how to leverage it to study the stability of switched system. We gave a rounding algorithm (Algorithm 4) that generates from a solution of the dual program a switching sequence of growth rate guaranteed by Theorem 5.3.2. We also showed in Proposition 5.3.1 that given atomic measures solution of this dual program, we can extract a high growth-rate trajectory of the switched systems by transforming them to the solution a scalar constraint switched systems.

We have introduced two techniques to generate lower bounds from the solution of the SOS dual program. In practice, these techniques provide periodic trajectories of high asymptotic growth rate. Since the SOS program can be solved efficiently, does this give an efficient algorithm to generate lower bounds on the CJSR with *guaranteed accuracy*? This is not clear, as we have no guarantee on length of the period of the trajectory generated and whether it is even periodic.

Example	length	GRIP [s]	BP [s]	d	l	SOS [s]	SEQ [s]
5.3.7	21	0.076	0.07	1	4	0.12	0.0013
				15	2	0.80	0.020
5.3.8	8	0.051	0.54	1	3	0.15	0.0011
				4	1	0.28	0.0011
5.3.9	41	0.040	5.08	1	9	0.14	0.073
				4	2	0.76	0.074

Table 5.2: Comparison of the performance of different algorithms to find a s.m.p. The second column provides the length of the smallest s.m.p. GRIP is the time taken by the Gripenberg algorithm [Gri96] to find a s.m.p. BP is the time taken by the Balanced Polytope algorithm [GP13] to find a s.m.p. The nonnegativity of the matrices is exploited for Example 5.3.7 as suggested in Section 4 of [GP13]. A point p is considered to belong to the interior of a balanced polytope P if MOSEK [ApS19] certifies that the maximal t such that $tp \in P$ is larger than $1 + 3 \times 10^{-13}$. SOS is the time taken by MOSEK [ApS19] to solve the pair of primal-dual programs Program 3.2.2/Program 5.3.2 with degree d using a bisection on γ until $\log(\overline{\gamma}) - \log(\underline{\gamma}) < 1 \times 10^{-2}$ where $\overline{\gamma}$ is the smallest γ such that Program 3.2.2 is feasible and $\underline{\gamma}$ is the largest γ such that Program 5.3.2 is feasible. SEQ is the time taken by Algorithm 4 with input l, d and $\underline{\gamma}$. The timings are taken from `Benchmark.html` of [LPJ19c].

Example	δ	GRIP [s]	BP [s]	d	SOS [s]
5.3.7	6×10^{-4}	1.58	3.038	7	0.91
5.3.8	25×10^{-8}	9.04	0.207	7	1.63
5.3.9	1×10^{-3}	0.37	14.273	6	8.40

Table 5.3: Comparison of the performance of different algorithms to find an upper bound to the CJSR. GRIP is the time taken by the Gripenberg algorithm [Gri96] to prove the upper bound $\rho(G, \mathcal{A}) + \delta$. The timing BP differs from the timing BP in Table 5.2 in the fact that we wait for the algorithm to prove that it is an s.m.p. The timing SOS differs from the timing SOS in Table 5.2 only in the bisection stopping criterion which is $\overline{\gamma} - \rho(G, \mathcal{A}) < \delta$ for this table. The timings are taken from `Benchmark.html` of [LPJ19c].

5.4 Constrained switching stabilitization

In this section, we are motivated by a new application [Gom+17] in the field of co-simulation. It is a numerical technique to couple multiple simulators, each simulating a part of a coupled system, in order to compute the overall behavior more efficiently [Gom+18b]. We focus on the abstract problem in this section and refer the reader to [Gom+18a] for more details on the application to co-simulation.

We tackle the problem of how to best forbid policy sequences that make the constrained switched system unstable. We propose that the best solution is to maximize the *entropy* of the stabilized constrained switched system in order to maximize the adaptability of the resulting (co-) simulation method.

Our goal is to optimally modify a given constrained switched system, by forbidding unstable switching signal cycles from the language it generates. The problem of finding such cycles is considered in Section 5.3. As such, we introduce the following definition, which represents any algorithm available for this purpose.

Definition 5.4.1 (Oracle)**.** Given $\epsilon > 0$, we define a *stability oracle* $O_\epsilon : S \to \{\texttt{Stable}\} \cup \bigcup_{k=1}^{\infty} G_k[k]^\circ$, where S is a CSS. The oracle O_ϵ returns either `Stable` certifying that $\rho(\mathcal{A})(S) < 1$ or a cycle $c \in G_k[k]^\circ$ such that $\rho(A_c)^{1/k} > 1 - \epsilon$.

We emphasize that the oracle has a (slightly) imperfect behavior: in case $1 - \epsilon < \rho(\mathcal{A})(S) < 1$, one cannot guarantee what the outcome of the oracle will be. This imperfection is intentional, as it models the state of the art [PJ08]. The result (3.20) ensures that if $\rho(\mathcal{A})(S) > 1 - \epsilon$, there exists a k and a cycle $c \in G_k[k]^\circ$ such that $\rho(A_c)^{1/k} > 1 - \epsilon$.

We now proceed to define the set of possible different switching signals that are admissible.

Definition 5.4.2 (Admissible Regular Language)**.** We say that $\mathcal{L} = G^*$ is the language *recognized* by the automaton G. A language is *regular* if it is recognized by a finite automaton. A language $\mathcal{L}$ recognized by an automaton G is *admissible* for $\mathcal{A}$ if the constrained switched system $S = \langle \mathcal{A}, G \rangle$ satisfies $\rho(\mathcal{A})(S) < 1$.

Let $\mathcal{L}_0$ denote the language recognized by the automaton G_0 of a given $S = \langle \mathcal{A}, G_0 \rangle$. Informally, our goal is to find the "largest" regular language $\mathcal{L}^\star \subseteq \mathcal{L}_0$ that is admissible. For this optimization problem to be well defined we need to find a metric for the objective. This metric should be in accordance to the fact that given $\mathcal{L} \subseteq \mathcal{L}'$, the objective should favor $\mathcal{L}'$. A widely used notion to describe the size of a regular language is that of Entropy, defined in Definition 5.2.1.

We denote the entropy of the language G^* recognized by an automaton G as $h^*(G)$.

If $\mathcal{L} \subseteq \mathcal{L}'$, then $G_k \subseteq G'_k$ for any k, and so $h(\mathcal{L}) \leq h(\mathcal{L}')$. Our problem can now be formulated.

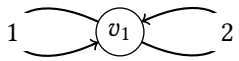
Problem 5.4.1. Given a CSS $\langle \mathcal{A}, G_0 \rangle$, find the language $\mathcal{L}^\star$ solution of the following optimization problem:

$$\begin{aligned} \mathcal{L}^\star = \sup_{\mathcal{L} \text{ regular}} \; & h(\mathcal{L}) \text{ s.t.} \\ & \mathcal{L} \subseteq \mathcal{L}_0, \\ & \mathcal{L} \text{ is admissible for } \mathcal{A}. \end{aligned} \tag{5.35}$$

where $\mathcal{L}_0$ is the language recognized by G_0.

Remark 5.4.1. In (5.4.1), we restrict our attention to regular languages. While there are examples that highlight the benefit of using non-regular languages (see Example 5.4.1), in practice, one needs an *efficient* way of generating accepted switching signals. For instance, during a co-simulation, at any step, the simulators need to compute as quickly as possible the set of policies that can be taken (see [Gom+17, Section 4.4] for how this can be done). Automata allow the decision procedure to be fast, with little memory. In addition, as hinted in Example 5.4.2, regular languages may be constructed to approximate an admissible language with entropy arbitrarily close to the entropy of the optimal solution, even if that optimal solution is a non-regular language.

Example 5.4.1. Consider $\mathcal{A} = \{A_1, A_2\}$, with $A_1 = 2$ and $A_2 = \frac{1}{2}$, and $G = (V, E)$, where $V = \{v_1\}$ and $E = \{(v_1, v_1, 1), (v_1, v_1, 2)\}$. That is, G has the form



The optimal solution $\mathcal{L}^\star$ of (relaxed) (5.4.1) should include every word that has more 1s than 2s. As shown in [Sip13, Example 1.73], no automaton can be built that accepts this language.

Example 5.4.2. Consider $\mathcal{A} = \{A_1, A_2\}$, with $A_1 = 1$ and $A_2 = \frac{1}{2}$. A language is admissible if it does not contain the infinite repetition of the symbol 1. Let $\mathcal{L}_k$ be language of all words that do not contain k consecutive 1's. Figure 5.5 suggests that that $h(\mathcal{L}_k)$ tends to $\log_2(2)$ when k tends to infinity. The quantity $\log_2(2)$ denotes the entropy of the optimal solution.

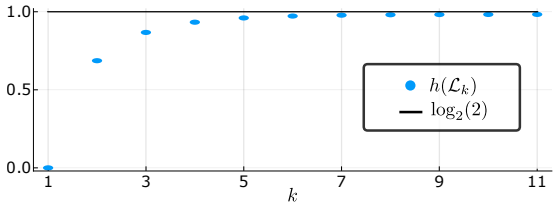


Figure 5.5: Evolution of $h(\mathcal{L}_k)$ of Example 5.4.2 in terms of k.

5.4.1 Lift-and-Constrain Stabilization

Constraining for more stability

Algorithm 5 details an iterative procedure that stabilizes a given CSS $S = \langle \mathcal{A}, G \rangle$, using the oracle in Definition 5.4.1. At each iteration, if the oracle returns a cycle $c = \sigma_k \dots \sigma_k$, then c is eliminated from G. The removal of a cycle can be accomplished by removing an edge of G, thus potentially decreasing $\rho(\mathcal{A})(S)$. After removing the cycle c, any infinite sequence in G^* for which c is a subsequence will be eliminated too. This is illustrated in Example 5.4.3. The algorithm can produce an empty CSS, which does not imply that the original CSS is impossible to stabilize. An empty CSS is trivially stable.

Example 5.4.3. Consider the automaton in Figure 5.6, and suppose the oracle has returned the cycle 234. This cycle is highlighted in red, in the figure. Any of the edges in red can be removed to forbid the unstable sequence. If edge $v_1 \xrightarrow{2} v_2$ is removed, the infinite sequences accepted by the resulting automaton end with either an infinite sequence of 2's, or an infinite sequence of 3's. If edge $v_2 \xrightarrow{3} v_3$ is removed instead, the resulting automaton accepts infinite sequences comprised of repeating subsequences which include 2, or 3, or 12.

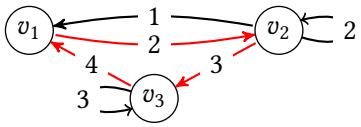


Figure 5.6: Automaton of Example 5.4.3.

As Example 5.4.3 shows, the choice of different edges to be removed has a different impact in the entropy of the resulting automaton. Informally, removing the edge $v_2 \xrightarrow{3} v_3$ seems to be the best choice because the resulting automata allows for *more* sequences. This is corroborated by computing the entropy of the resulting automaton alternatives. See Section 5.2.2 for how to compute the entropy in this example.

Algorithm 5 Stabilization algorithm for a constrained switched system. $h^*(G)$ denotes the entropy of the language recognized by G. The difference $G - e$ denotes the automaton obtained by removing the edge e from G.

Input A CSS $S = \langle \mathcal{A}, G \rangle$.
Output A stable CSS $S = \langle \mathcal{A}, G \rangle$.
while $O_\epsilon(S) \neq$ `Stable` **do**
1. Find $e \in \arg\max\{ h^*(G - e) \mid e \in E, e$ is an edge of the cycle $O_\epsilon(S) \}$
2. Set $G := G - e$

end while

The following result demonstrates that Algorithm 5 always terminates.

Theorem 5.4.1. Given a CSS $S = \langle \mathcal{A}, G \rangle$ and an oracle satisfying Definition 5.4.1, Algorithm 5 terminates after finitely many iterations and the resulting CSS is stable.

Proof. At each iteration of the algorithm, the number of edges of the automaton $G = (V, E)$ decreases by one. Since at the beginning of the algorithm $|E|$ is finite, the algorithm must terminate after a finite number of iterations. The condition for termination of Algorithm 5 implies that the resulting system is stable. □

Remark 5.4.2. In Theorem 5.4.1, the assumption that the oracle in Definition 5.4.1 always terminates is crucial, as the problem solved by the oracle is undecidable in general.

Lifting for less conservativeness

Algorithm 5 takes a constrained switched system $S = \langle \mathcal{A}, G \rangle$, and outputs a constrained switched system $S' = \langle \mathcal{A}', G' \rangle$ that is stable, while attempting to maximize the entropy of the language recognized by G'. If we let $\mathcal{L}'$ denote this language, then, relating this to (5.4.1), $\mathcal{L}'$ is admissible and regular, and thereby a potential solution. However, it may not be the optimal solution. Similarly, if the algorithm returns an empty CSS, this does not mean that the original CSS is impossible to stabilize. To maximize the entropy of the stabilized CSS's, we propose to take an *M-Path-Dependent lift* of the automaton representing the input language $\mathcal{L}_0$.

Definition 5.4.3 ([Phi+16, Definition 3])**.** Given an automaton G, we define the *lifted* automaton $G^{[k]}$ of degree k as follows. For each path $v_0, \sigma_0, v_1, \sigma_1, \ldots, \sigma_k, v_{k+1}$ with length $k + 1$ of G, $G^{[k]}$ has a node $u^- = v_0\sigma_0 v_1\sigma_1 \ldots \sigma_{k-1}v_k$, a node $u^+ = v_1\sigma_1 v_2\sigma_2 \ldots \sigma_k v_{k+1}$ and an edge (u^-, u^+, σ_k).

Figure 5.7 shows the second degree ($k = 2$) lift of the automaton in Figure 5.6.

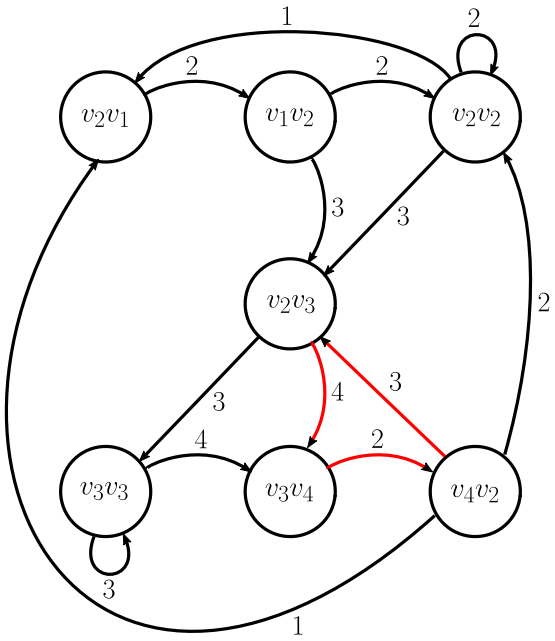


Figure 5.7: Second degree lifted automaton of Example 5.4.3.

The lifted automaton represents the same language, as shown by Proposition 5.4.1, but, as suggested by Theorem 5.4.2 and illustrated by Example 5.4.2, lifting the automaton before applying Algorithm 5 allows one to obtain an admissible language with higher entropy.

Proposition 5.4.1. Let $\mathcal{L}$ be the language recognized by G and $\mathcal{L}^{[k]}$ the language recognized by $G^{[k]}$, where $G^{[k]}$ is the lift of degree k of G. Then $\mathcal{L} = \mathcal{L}^{[k]}$.

Proof. Consider a sequence $\sigma_0 \dots \sigma_{i-1}$.

If $\sigma_0 \dots \sigma_{i-1} \in G_k[i]$, there exists nodes $v_0, v_1, \dots, v_i$ of G such that

$$v_0, \sigma_0, v_1, \sigma_1, \dots, \sigma_{i-1}, v_i$$

is a path of G. As no node has zero ingoing degree, there exists a path of length k that ends in node v_0, denoted as $v_{-k}, \sigma_{-k}, \dots, \sigma_{-1}, v_0$ in G. By Definition 5.4.3, for any $j = 0, \dots, i$, $u_j = v_{j-k}\sigma_{j-k} \dots \sigma_{j-1}v_j$ is a node of $G^{[k]}$ and for any $j = 0, \dots, i-1$, there is an edge (u_j, u_{j+1}, σ_j) in $G^{[k]}$. Therefore $\sigma_0 \dots \sigma_{i-1} \in G_i^{[k]}$.

If $\sigma_0 \dots \sigma_{i-1} \in G_i^{[k]}$, there exists nodes $u_0, u_1, \dots, u_i$ of $G^{[k]}$ such that $u_0, \sigma_0, u_1, \sigma_1, \dots, \sigma_{i-1}, u_i$ is a path of $G^{[k]}$. Let $v_{-k}, \dots, v_i$ be the nodes of G and

$\sigma_{-k}, \ldots, \sigma_{-1}$ be the symbols such that for any $j = 0, \ldots, i$, $u_j = v_{j-k}\sigma_{j-k}\ldots\sigma_{j-1}v_j$. By Definition 5.4.3, $v_0, \sigma_0, v_1, \sigma_1, \ldots, \sigma_{i-1}, v_i$ is a path of G hence $\sigma_0\ldots\sigma_{i-1} \in G_k[i]$.

□

Theorem 5.4.2. Consider Algorithm 5 with input $\mathcal{A}, G_0^{[k]}$ (resp. $\mathcal{A}, G_0^{[k+1]}$) where $G_0^{[k]}$ (resp. $G_0^{[k+1]}$) is the lift of degree k (resp. $k+1$) of a given automaton G. If $O_\epsilon(\mathcal{A}, G_0^{[k]})$ and $O_\epsilon(\mathcal{A}, G_0^{[k+1]})$ are cycles corresponding to the same word, then $h^*(G_1^{[k]}) \leq h^*(G_1^{[k+1]})$.

Proof. Let e be the edge such that $G_1^{[k]} = G_0^{[k]} - e$, that is, the edge removed by the algorithm for $G_0^{[k]}$. Let $\sigma_1\sigma_2\ldots\sigma_k\sigma_{k+1}\sigma_{k+2}$ be a sub-word of the repetition of the cycle c and $v_1, v_2, \ldots, v_{k+3}$ be such that

$$e = (v_1\sigma_1v_2\sigma_2\ldots\sigma_kv_{k+1}, v_2\sigma_2\ldots\sigma_kv_{k+1}\sigma_{k+1}v_{k+2}, \sigma_{k+1})$$

and $(v_{k+2}, v_{k+3}, \sigma_{k+2})$ is an edge of G. Let ${G_0^{[k+1]}}'$ be the graph obtained by removing the node $v_1\sigma_1v_2\sigma_2\ldots\sigma_{k+1}v_{k+2}$ in $G_0^{[k+1]}$. The two automata $G_1^{[k]}$ and ${G_0^{[k+1]}}'$ recognize the same language. Let ${G_0^{[k+1]}}''$ be the graph obtained by removing the edge $e' = (v_1\sigma_1v_2\sigma_2\ldots\sigma_{k+1}v_{k+2}, v_2\sigma_2v_3\sigma_3\ldots\sigma_{k+2}v_{k+3}, \sigma_{k+2})$ in $G_0^{[k+1]}$. The language recognized by ${G_0^{[k+1]}}'$ is a subset of the language recognized by ${G_0^{[k+1]}}''$.

Moreover, as e' is an edge of the cycle $O_\epsilon(\mathcal{A}, G_0^{[k+1]})$, $h^*({G_0^{[k+1]}}'') \leq h^*(G_1^{[k+1]})$. Therefore

$$h^*({G^{[k]}}_0) = h^*({G_0^{[k+1]}}') \leq h^*({G_0^{[k+1]}}'') \leq h^*(G_1^{[k+1]}).$$

□

In Section 5.4.3 we show results corroborating Theorem 5.4.2.

5.4.2 Implementation Details & Optimality

Implementation

The implementation of the stabilization of a CSS is summarized as follows:

1. find all unstable cycles of length up to 3 using brute force enumeration;
2. since several cycles can be disallowed by removing a single edge, select the edge that disallows the largest number of unstable cycles, and use the entropy of the resulting graph to break ties;
3. repeat steps 1–2 until all allowed cycles have a spectral radius below 1;

4. Use the method of [LPJ17] to determine whether the resulting CSS is stable or whether there is an unstable cycle.

5. if there is an unstable cycle, select the edge that maximizes the entropy of the resulting system (steps 1–2 of Algorithm 5);

6. repeat steps 4–5 until the resulting system is stable.

It is easy to see that this implementation is a realization of Algorithm 5. Steps 1–4 are an optimization since they execute relatively quickly, and make the execution of the method in [LPJ17] take less time.

In Step 6, instead of computing the entropy, we compute the spectral radius of the adjacency matrix of the resulting system. This is equivalent to maximizing the entropy (see Section 5.2.2).

Optimality

The solution attained by Algorithm 5 is not necessarily the optimal solution. For once, applying different lift degrees will yield different optimal solutions. Second, Algorithm 5 removes an edge before finding the next unstable cycle, which means that it misses the chance of optimizing which edge to remove, when more cycles are available (recall steps 1–3 of the above implementation).

Unfortunately, we found no way of guessing which lift degree yields the optimal solution. However, with small enough constrained switched systems (in the number of matrices and states), it is possible to find the optimal solution, for a given lift degree k.

To find the optimal solution, suppose that, for a CSS with a lift degree k, we know what the unstable cycles are. Now we can iterate over all possible procedures to disallow these cycles in the CSS (each procedure is a sequence of edges to be removed), and compute the entropy of the resulting CSS. The optimal solution is the one that has the maximal entropy.

In order to collect all the unstable cycles, the following procedure can be used:

1. given a CSS with a lift of degree k, apply Algorithm 5 to find an admissible language, and record all the cycles that were removed throughout the procedure;

2. iterate over all possible ways of disallowing the cycles on the original CSS with a lift of degree k, and apply the one that results in a language with maximal entropy;

3. the resulting language is not necessarily admissible, because the best procedure is not necessarily that same as the one picked by Algorithm 5

in Step 1, so apply Algorithm 5 to identify and disallow the remaining cycles, adding these to the set of unstable cycles.

4. now repeat Steps 2–3, collecting more and more unstable cycles, until the resulting language is admissible.

The resulting set of unstable cycles represents all possible unstable cycles, and the admissible language found is the optimal solution.

An example application is described in Section 5.4.3.

5.4.3 Application

Consider the unconstrained switched system described in [Gom+18a, Section 4]. Applying Algorithm 5 directly to the unconstrained switched system (which corresponds to a lift of degree 0), leads to removal of the edges with labels 2, 3 and 4. This completely disallows the use of the matrices A_2, A_3 and A_4. The resulting language turns out to be admissible, its entropy is $\log_2(5)$.

Applying Algorithm 5 to a lift of degree 1, we get a constrained switched system with the automaton shown in Figure 5.8, where the edges in red were removed by the algorithm. We can see that the matrices A_2, A_3 and A_4 are now allowed by the algorithm (only the cyclic application of each one of these matrices is still disallowed). This solution is less conservative than the one with degree 0. Its entropy is $\log_2(7.26)$. One allowed cycle is 32645, where the symbols 5 and 6 seem to play the role of stabilizing the cycle.

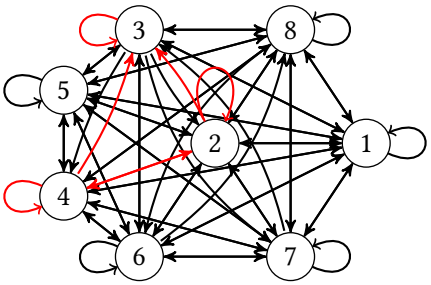


Figure 5.8: Solution with entropy $\log_2(7.2568898)$**.**

We applied Algorithm 5 to the lifts of degree 0, 1 and 2. At each application of the algorithm, a stable constrained switched system was produced, with an entropy that increased with the degree of the lift. These results, summarized in Table 5.4, corroborate Theorem 5.4.2.

This application to co-simulation illustrates an important advantage of the method presented in Section 5.3: it is capable of finding large unstable cycles. This method does not find the unstable cycles by iterating through the cycles of some length K but instead extracts them from an infinite switching

Table 5.4: Entropy achieved per lift degree.

k	Entropy [bit]	CPU time [s]
0	$\log_2(5)$	0.13
1	$\log_2(7.2568898)$	1.8
2	$\log_2(7.7083039)$	280

signal, hence it is not harder for the method to find large unstable cycles. For the lift of degree 2 for example, it found the unstable cycle 542245332 of length 9, and in a subsequent iteration found the cycle 22453354224533542245335422452325 4 of length 33. A brute force method would have to enumerate all 8^{33} cycles to achieve the stabilization of the adaptive solver.

Regarding the optimality of the solution found for the lift with degree 1, we have applied the procedure detailed in Section 5.4.2 to confirm that $\log_2(7.2568898)$ is indeed the maximal entropy for that degree.

5.4.4 Conclusion

As the stability of the system is a set program, we showed in this section how the rounding algorithm presented in Section 5.3 can be used to restrict the switching possibilities of a constrained switching systems so as to leave as many switching policies as possible (provided that the system becomes stable).

Our algorithm takes the form of a hierarchy of sufficient conditions, where increasingly better solutions are found by lifting the automaton (see Figure 5.5 and Table 5.4). Essentially, this allows one to control the optimality of the solution, at the cost of processing power and memory.

5.5 Low rank reduction

In [AP12b], Ahmadi and Parrilo show how to reduce the computation of the JSR of matrices that are all of rank one to a combinatorial problem, which coincides with the CJSR of 1×1 matrices (i.e. scalars). As a final contribution, we generalize this approach and give a reduction of the computation of the JSR (or CJSR) of matrices that are all of rank at most r to the computation of the CJSR of $r \times r$ matrices.

Suppose we want to compute the CJSR of a finite set of matrices $\mathcal{A} \triangleq \{A_1, \dots, A_m\} \subset \mathbb{R}^{n \times n}$ of rank at most r constrained by an automaton $G(V, E)$. For $\sigma = 1, \dots, m$, since the matrix A_σ has rank at most r, there exists $X_\sigma, Y_\sigma \in \mathbb{R}^{n \times r}$ such that $A_\sigma = X_\sigma Y_\sigma^\top$. This can be used to build a new system with matrices of $\mathbb{R}^{r \times r}$ with the same CJSR. This new system can therefore be used to reduce the computation of the CJSR of a system of low rank matrices to a

system of matrices of small size. Note that in the case $r = 1$, it is known that the CJSR is computable in polynomial time [AP12b].

Theorem 5.5.1 (Low Rank Reduction)**.** Consider a finite set of matrices $\mathcal{A} \triangleq \{A_1, \dots, A_m\} \subset \mathbb{R}^{n\times n}$ of rank at most r constrained by an automaton $G(V, E)$.

For a fixed decomposition $A_\sigma = X_\sigma Y_\sigma^T$ for $\sigma = 1, \dots, m$ where $X_\sigma, Y_\sigma \in \mathbb{R}^{n\times r}$, denote the set of matrices $\mathcal{A}' \triangleq \{A'_{\sigma_1\sigma_2} \mid \sigma_1, \sigma_2 = 1, \dots, m\} \subset \mathbb{R}^{r\times r}$ where $A'_{\sigma_1\sigma_2} = Y_{\sigma_1}^T X_{\sigma_2}$. Define the graph $G'(V', E')$ with $V' \triangleq E$ and

$$E' \triangleq \{\, ((u, v, \sigma_1), (v, w, \sigma_2), \sigma_2\sigma_1) \mid (u, v, \sigma_1), (v, w, \sigma_2) \in E \,\}.$$

Then the two CJSR are the same: $\rho(G, \mathcal{A}) = \rho(G', \mathcal{A}')$.

Proof. As the CJSR does not depend on the norm used, we choose a norm $\|\cdot\|$ that is *submultiplicative*, that is $\|AB\| \le \|A\|\|B\|$ for all matrices A, B.

Let $\beta = \max_{\sigma=1}^m \max\{\|X_\sigma\|, \|Y_\sigma^T\|\}$. If $\beta = 0$, then $\rho(G, \mathcal{A}) = 0 = \rho(G', \mathcal{A}')$. Therefore we may assume that $\beta > 0$. Consider a positive integer k. We first show that $[\hat\rho_k(G, \mathcal{A}, \|\cdot\|)]^k \le \beta^2[\hat\rho_{k-1}(G', \mathcal{A}', \|\cdot\|)]^{k-1}$ where $\hat\rho_k(G, \mathcal{A}, \|\cdot\|)$ is defined in (3.21). For any G-admissible $(\sigma_1, \sigma_2, \dots, \sigma_k)$, we have

$$A_{\sigma_k} \cdots A_{\sigma_2} A_{\sigma_1} = X_{\sigma_k} A'_{\sigma_k\sigma_{k-1}} \cdots A'_{\sigma_3\sigma_2} A'_{\sigma_2\sigma_1} Y_{\sigma_1}^T.$$

using the submultiplicativity of the norm chosen, we have

$$\begin{aligned}\|A_{\sigma_k} \cdots A_{\sigma_1}\| &\le \|X_{\sigma_k}\| \cdot \|A'_{\sigma_k\sigma_{k-1}} \cdots A'_{\sigma_3\sigma_2} A'_{\sigma_2\sigma_1}\| \cdot \|Y_{\sigma_1}^T\| \\ &\le \beta^2 \|A'_{\sigma_k\sigma_{k-1}} \cdots A'_{\sigma_3\sigma_2} A'_{\sigma_2\sigma_1}\| \\ &\le \beta^2 [\hat\rho_{k-1}(G', \mathcal{A}', \|\cdot\|)]^{k-1}.\end{aligned}$$

The same way, we now show that $[\hat\rho_{k-1}(G', \mathcal{A}', \|\cdot\|)]^{k-1} \le \beta^2[\hat\rho_{k-2}(G, \mathcal{A}, \|\cdot\|)]^{k-2}$. For any G'-admissible $(\sigma_2\sigma_1, \dots, \sigma_k\sigma_{k-1})$, we have

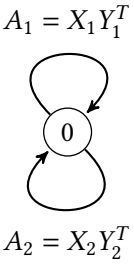
$$\begin{aligned}\|A'_{\sigma_k\sigma_{k-1}} \cdots A'_{\sigma_3\sigma_2} A'_{\sigma_2\sigma_1}\| &\le \|Y_k^T\| \cdot \|A_{\sigma_{k-1}} \cdots A_{\sigma_2}\| \cdot \|X_1\| \\ &\le \beta^2 [\hat\rho_{k-2}(G, \mathcal{A}, \|\cdot\|)]^{k-2}.\end{aligned}$$

In summary, we have

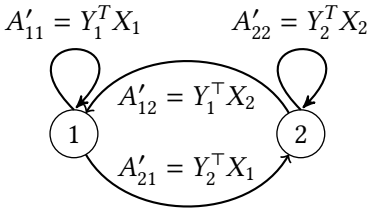
$$\begin{aligned}\hat\rho_k(G, \mathcal{A}, \|\cdot\|) &\le \beta^{\frac{2}{k}} [\hat\rho_{k-1}(G', \mathcal{A}', \|\cdot\|)]^{\frac{k-1}{k}} \\ &\le \beta^{\frac{4}{k}} [\hat\rho_{k-2}(G, \mathcal{A}, \|\cdot\|)]^{\frac{k-2}{k}}.\end{aligned}$$

Taking the limit $k \to \infty$ we get $\rho(G, \mathcal{A}) \le \rho(G', \mathcal{A}') \le \rho(G, \mathcal{A})$. □

Example 5.5.1. Consider an unconstrained switched system with 2 rank r matrices A_1, A_2. This system is equivalent to the constrained switched system with automaton represented in Figure 5.9a. Its low rank reduction is represented in Figure 5.9b.



(a) Automaton G. We have $V = \{0\}$ and $E = \{(0,0,1),(0,0,2)\}$.



(b) Automaton G'. We have $V' = \{1,2\}$ and $E' = \{(1,1,11),(1,2,21),(2,1,12),(2,2,22)\}$.

Figure 5.9: Simple example of the low rank reduction.

Remark 5.5.1. The matrices X_σ, Y_σ of the factorization $A_\sigma = X_\sigma Y_\sigma^T$ are not unique. For any invertible matrix $S \in \mathbb{R}^{r\times r}$, $A_\sigma = (X_\sigma S)(S^{-1}Y_\sigma^T)$ also gives a factorization. However, if $\rho(G', \mathcal{A}')$ is approximated using Program 3.2.2, any two factorizations will give the same approximation. The effect of using $X_\sigma S$ and $Y_\sigma S^{-T}$ instead of X_σ and Y_σ will simply be a linear change of variable of the polynomial p_σ.

What is the impact of this reduction on the computational complexity and accuracy of the approximation ? The entropy of the language of allowed switching signals is the same for the initial system and the reduced system hence the guarantee in Theorem 5.2.3 is the same for both systems. However, the dimension of the matrices goes from the dimension of the matrices n to their rank r hence for low rank matrices the guarantee in Theorem 5.2.4 is improved.

In terms of computational complexity, there can be up to m nodes and m^2 edges in the automaton of the reduced system. Therefore, even if the size of the matrices decreases from n to r, the number of variables and constraints increases. This shows that the reduction only decreases the computational complexity if the rank of the matrices is *sufficiently* low.

5.6 Conclusion

We have analyzed in details the set program with constraints of the form (4.4) or (4.5) for polysets of degree $2d$. We showed that one guarantee is tight in Section 5.1 and provided a new guarantee in Section 5.2 but in practice, the performance exceeds the worst-case expectation based on these guarantees. One promising direction might be to provide a guarantee based on the matrices describing the switched system, similarly to the guarantee of Conjecture 4.6.1 based on the joint condition number of the input matrices of the set program.

In Section 5.3, we showed how to exploit the infeasibility certificate of a set program for polysets in order to certify the infeasibility of the generic set program and in Section 5.4, we showed how to minimally relax the set program in order to render it feasible. It is however unclear why this approach would be specific to the class of set programs studied in this section and it would be compelling to extend this technique to other classes of set programs.

Stabilizability 6

As mentioned in the introduction of Section 3.3, the computation of controlled invariant set is a key problem in System and Control theory. The design of an efficient algorithm for computing controlled invariant sets of a class of systems renders many important problems efficiently solvable; see [Bla99] for more details. We introduced in Section 3.3 an algebraic approach for the computation of controlled invariant ellipsoids using the LMI (3.32).

An alternative approach to (3.32) for computing ellipsoidal controlled invariant sets was introduced in [LTJ18; LTJ20; LRJ20b]. A key ingredient in this technique is that the problem is formulated in the dual space of the geometric problem. It leverages duality like [Boy+94; AG15] but contrary to them it combines it with a projection of the state space to its subspace that is not *directly* controllable. Compared to the method described above, with this approach the *controlled* invariance LMI constraint has size $n \times n$ and not $(2n) \times (2n)$. Both methods solve the problem in the dual space as the ellipsoid $\mathcal{E}_{P^{-1}}$ is the polar of the ellipsoid $\mathcal{E}_P$. However, the method of [LTJ18; LTJ20; LRJ20b] is developed at the abstract level of sets instead of relying on algebraic manipulation at the level of matrices. As shown in Chapter 4, this allows the method to be extended to other templates such as piecewise semi-ellipsoids, polysets and piecewise polysets.

In Section 6.1, we show how to use this geometric approach to compute controlled invariant set for a continuout-time control systems (3.5).

In Section 6.2, we show how to use the results of Section 4.4 to compute controlled invariant sets for discrete-time control systems. We detail the application of the method to two classes of hybrid systems: Discrete-Time Affine Hybrid Control System (HCS for short) and Discrete-Time Affine Hybrid Algebraic System (HAS for short). HAS are not control systems but the computation of invariant sets for such systems presents the same features than for HCS.

As the computation of controlled invariant sets of a HCS has been reduced to the computation of invariant sets of a HAS, we describe in Section 6.2 our method to compute invariant sets of HAS as ellipsoids or polysets. In Section 6.2.1, we show that using the results of Section 4.4, the invariance of ellipsoids for a *homogeneous* HAS, see Definition 3.1.5, can be formulated as a semidefinite program. In Section 6.2.2, we show how to use homogenization

technique developed in Section 4.7 to generalize the semidefinite program of Section 6.2.1 to non-homogeneous HAS. In Section 6.2.3, we generalize the semidefinite program of Section 6.2.1 to compute invariant polysets of arbitrary degree.

We end the chapter with an application of the controlled invariant sets to safety critical model predictive control in Section 6.3 and stochastic programming in Section 6.4. We show that precomputing such sets allows to guarantee safety of the model predictive controller thereby removing the need for long horizon in model predictive control and infeasibility cuts in stochastic programming.

6.1 Continuous-time

Proposition 6.1.1. A set S is controlled invariant for control system (3.5) with $\mathcal{U} = \mathbb{R}^{n_u}$ if and only if it is controlled invariant for the control algebraic system (3.9)

$$\pi_{\mathrm{Im}(B)^\perp}\dot{x} = \pi_{\mathrm{Im}(B)^\perp}Ax$$

where $\pi_{\mathrm{Im}(B)^\perp}$ is a projection into the orthogonal complement of the linear subspace $\mathrm{Im}(B)$.

Proof. By Proposition 1.1.5, there exists $u \in \mathbb{R}^{n_u}$ such that $\dot{x} = Ax + Bu$ if and only if $\pi_{\mathrm{Im}(B)^\perp}\dot{x} = \pi_{\mathrm{Im}(B)^\perp}Ax$. As the input u is unconstrained, the result follows. □

Proposition 6.1.2 (Nagumo condition [BM15, Theorem 4.7])**.** A set S is invariant for system (3.9) if and only if

$$\forall x \in \partial S, \exists y \in T_S(x), Ey = Ax. \tag{6.1}$$

The Nagumo condition can then be expressed in terms of the normal cone instead of the tangent cone, as we will see in (6.3).

Theorem 6.1.1 (Controlled invariance of convex set)**.** A convex set C is invariant for system (3.9) with matrices $A, E \in \mathbb{R}^{r\times n}$ if and only if

$$\forall z \in \mathbb{R}^r, \forall x \in F_C(E^\top z), \langle z, Ax\rangle \leq 0. \tag{6.2}$$

Proof. As C is convex, $T_S(x)$ is a convex cone. By definition of the polar of a cone, $x \in ET_S(x)$ if and only if $\langle y, x\rangle \leq 0$ for all $y \in [ET_S(x)]^\circ$. By Proposition 1.2.21, $[ET_S(x)]^\circ = E^{-\top}N_S(x)$. Therefore, the set C is invariant if and only if

$$\forall x \in \partial C, \forall z \in E^{-\top}N_C(x), \langle y, x\rangle \leq 0. \tag{6.3}$$

By Proposition 1.2.23, we have

$$\{\,(x,z) \in \partial\mathcal{C} \times \mathbb{R}^r \mid E^\top z \in N_C(x)\,\} = \{\,(x,z) \in \partial\mathcal{C} \times \mathbb{R}^r \mid x \in F_C(E^\top z)\,\}.$$

As $\operatorname{Im}(F_C) \subseteq \partial\mathcal{C}$ (with equality if every face is exposed), we obtain (6.2) □

Theorem 6.1.2. Consider a nonempty closed convex set C such that $\delta^*(\cdot|C)$ is differentiable. Then C is invariant for system (3.9) with matrices $A, E \in \mathbb{R}^{r\times n}$ if and only if

$$\forall z \in \mathbb{R}^r, \langle z, A\nabla\delta^*(E^\top z|C)\rangle \leq 0. \tag{6.4}$$

Proof. By Proposition 1.2.24, $F_C(E^\top z) = \{\nabla\delta^*(E^\top z|C)\}$ hence (6.2) is equivalent to (6.4). □

6.1.1 Ellipsoid template

Note that while the quadratically stabilizable systems of the form (3.5) is equivalent to their quadratically stabilizable via linear control, it is no longer the case for *uncertain* or *switched* systems.

Since the support function of an ellipsoid $\mathcal{E}_P$ is $\delta^*(y|\mathcal{E}_P) = \sqrt{y^\top P y}$, we have the following corollary of Theorem 6.1.2.

Corollary 6.1.1. Given a positive definite matrix P, the ellipsoid $\mathcal{E}_P$ is controlled invariant for system (3.9) if and only if

$$AP^{-1}E^\top + EP^{-1}A^\top \preceq 0. \tag{6.5}$$

Note that for the trivial case $\operatorname{Im}(B) = \mathbb{R}^n$ for system (3.5), the Proposition 6.1.1, will produce a system (3.9) with $r = 0$ hence the LMI (6.5) will be trivially satisfied for any P^{-1} which is expected.

In comparison to (3.30), for a system (3.9) with matrices $A, E \in \mathbb{R}^{r\times n}$, the LMI (3.30) has size $n\times n$ while the LMI (6.5) has only size $r\times r$. The characterization of controlled invariance of ellipsoids using (6.5) can also be obtained by applying an elimination procedure to reduce (3.30); see [Boy+94, Equation (7.11)]. However, uncertain or switched system may need a nonlinear state feedback to be quadratically stabilizabilized [Pet85]. For such systems, (3.30) is conservative since it assumes a linear feedback while (6.5) does not assume anything about the feedback. It was shown in [Bar85] that if (6.5) is satisfied then a stabilizing nonlinear continous state feedback can be deduced from the solution P. There is even a closed form for the feedback in case of single input [Bar85, Eq. (15)].

6.1.2 Polyset template

Moreover, to find invariant sets more sophisticated than ellipsoids, we can simply search for sets such that the support function is a polynomial.

Corollary 6.1.2. Given an homogeneous nonnegative polynomial $p(x)$ of degree $2d$, the set C of support function $\delta^*(y|C) = p(y)^{\frac{1}{2d}}$ is invariant for system (3.9) with matrices $A, E \in \mathbb{R}^{r\times n}$ if and only if the polynomial

$$z^\top A \nabla p(E^\top z) \tag{6.6}$$

is nonpositive for all $z \in \mathbb{R}^r$.

Proof. We have

$$\nabla \delta^*(y|C) = \frac{1}{p(y)^{1-\frac{1}{2d}}} \nabla p(y)$$

and since $p(y)^{1-\frac{1}{2d}}$ is positive, (6.4) is equivalent to (6.6). □

While verifying the nonnegativity of polynomials is co-NP-hard, we can find invariant sets by restricting them to be SOS as described in Section 2.3.

6.2 Discrete-time

6.2.1 Ellipsoid template for homogeneous systems

In this section, we show how to compute ellipsoidal controlled invariant sets using Theorem 4.4.1 and (4.18) in the particular case of homogeneous HCS and HAS. We show how to handle non-homogeneity in Section 6.2.2 and how to use more general sets in Section 6.2.3; we start by describing the ellipsoidal homogeneous case for clarity. This section details the semidefinite program needed to find these ellipsoidal invariant sets and shows its exactness in Theorem 6.2.1.

The optimization problem to solve is given in Program 6.2.1. The complexity for solving the semidefinite program is given by (2.15) where n and m depends on whether the program is written in the standard form (2.9) or the conic form (2.10) but in either cases, it depends affinely on the state dimension, the number of nodes and the number of transitions of the system. We use the notation $p_q(x, z)$ to denote the evaluation of p_q at the vector $y = (x, z)$.

Program 6.2.1.

$$\max_{D_q \succ 0} \quad \sum_{q\in V} \log\det D_q$$

$$A_\sigma D_q A_\sigma^\top \preceq E_\sigma D_{q'} E_\sigma^\top, \qquad \forall q \to_\sigma q' \tag{6.7}$$

$$a^\top D_q a \le \beta^2, \qquad \forall q \in V, (a,\beta) \in \mathcal{H}_{\text{rep}}(\mathcal{P}_q) \tag{6.8}$$

where $\mathcal{H}_{\text{rep}}(\mathcal{P}_Q)$ denotes the set of all (a, β) such that the half-space $a^\top x \leq \beta$ supports $\mathcal{P}_q$, which is commonly referred to as its H-representation [Zie95].

The constraint (6.7) is (4.18), it ensures invariance of the set. The constraint (6.8) ensures that C_q is contained in $\mathcal{P}_q$.

Remark 6.2.1. As we show in Theorem 6.2.1, (6.7) and (6.8) ensures that only invariant ellipsoids are feasible for Program 6.2.1. The most *relevant* solution among all feasible solutions depends on the application. Hence the objective function may involve some or all the ellipsoids and use one metric or another depending on the purpose of the optimization. We consider the sum of the $\log\det D_q$ in Program 6.2.1 but it can be replaced by any of the variations listed in Section 4.2.1.

Theorem 6.2.1. Consider a homogeneous HAS S as in Definition 3.1.4. The symmetric matrix D_q is feasible for Program 6.2.2 if and only if there exist invariant convex sets $C = (C_q)_{q\in V}$, as defined in Definition 3.3.2, such that $C_q^\circ = \mathcal{E}_{D_q}$ for all $q \in V$. Moreover, the optimal solution of Program 6.2.2 is the solution that maximizes the sum of the logarithms of the volume of the ellipsoids.

Proof. We first show that (6.7) is equivalent to the invariance of the sets C_q. By Theorem 4.4.1 and (4.18), the invariance of the sets C_q is equivalent to (6.7).

We now show that (6.8) is equivalent to $C_q \subseteq \mathcal{P}_q$. Since $\mathcal{P}_q$ is symmetric, it is the intersection of pairs of half-spaces $-\beta \leq \langle a, x\rangle \leq \beta$ and $C_q \subseteq \mathcal{P}_q$ if and only if $a/\beta, -a/\beta \in C_q^\circ$ for each pair of half-spaces. Since $C_q^\circ = \mathcal{E}_{D_q}$, this is equivalent to (6.8). □

6.2.2 Ellipsoid template for non-homogeneous systems

In this section, we show how to adapt Program 6.2.1 in the case of non-homogeneous systems using the homogenization technique detailed in Section 4.7. The non-homogeneous version of Theorem 4.4.1 is given as follows.

Theorem 6.2.2. Consider a HAS S as in Definition 3.1.4. The closed convex sets $C = (C_q)_{q\in V}$ are *invariant* for S, as defined in Definition 3.3.2, if and only if $C_q \subseteq \mathcal{P}_q$ for each $q \in V$ and for all $q \to_\sigma q'$,

$$r(A_\sigma, c_\sigma)^{-\top}\tau(C_q)^* \supseteq r(E_\sigma, 0)^{-\top}\tau(C_{q'})^*. \tag{6.9}$$

Proof. The invariance constraint (3.33) of Definition 3.3.2

$$A_\sigma C_q + c_\sigma \subseteq E_\sigma C_{q'}$$

can be rewritten, using (4.22), into

$$r(A_\sigma, c_\sigma)\tau(C_q) \subseteq r(E_\sigma, 0)\tau(C_{q'}). \tag{6.10}$$

As the sets C_q are closed and convex, so are the cones $\tau(C_q)$ hence $\tau(C_q)^{**} = \tau(C_q)$. Therefore, by Proposition 1.2.21, (6.10) is equivalent to (6.9). □

The optimization problem to solve is represented in Program 6.2.2. To transform this program into a semidefinite program, the equality (6.12) between quadratic forms is replaced by an equality corresponding to the coefficient of each quadratic monomial in y and the inequality (6.13) between quadratic forms is replaced by an LMI constraint. The complexity for solving the semidefinite program is given by (2.15) where n and m depends on whether the program is written in the standard form (2.9) or the conic form (2.10) but in either cases, it depends affinely on the state dimension, the number of nodes and the number of transitions of the system. We use the notation $p_q(x, z)$ to denote the evaluation of p_q at the vector $y = (x, z)$.

Program 6.2.2.

$$\max_{\substack{D_q \in \mathcal{S}^n, d_q \in \mathbb{R}^n, \\ \delta_q \in \mathbb{R}, \lambda_{q \to_\sigma q'} \geq 0}} \quad \sum_{q \in V} \log\det D_q$$

$$\begin{bmatrix} D_q & d_q \\ d_q^\top & \delta_q + 1 \end{bmatrix} \succ 0 \tag{6.11}$$

$$p_q(y) = y^\top H_{h_q} \begin{bmatrix} D_q & d_q \\ d_q^\top & \delta_q \end{bmatrix} H_{h_q} y \tag{6.12}$$

$$p_q(r(A_\sigma, c_\sigma)^\top y) \leq \lambda_{q \to_\sigma q'} p_{q'}(r(E_\sigma, 0)^\top y), \quad \forall q \to_\sigma q', y \in \mathbb{R}^{n_{q,x}+1} \tag{6.13}$$

$$p_q(-a, \beta) \leq 0, \quad \forall q \in V, (a, \beta) \in \mathcal{H}_{\mathrm{rep}}(\mathcal{P}_q) \tag{6.14}$$

$$p_q(0, 1) < 0, \quad \forall q \in V. \tag{6.15}$$

The constraint (6.11) ensures both convexity of $\tau(C_q)^*$ and the fact that $\det D_q$ does not overestimate the volume of the ellipsoid transformed by the Householder reflection. The constraint (6.13) is the S-procedure applied to the condition (6.9). The constraint (6.14) uses (4.23) to ensure that C_q is contained in $\mathcal{P}_q$. The constraint (6.15) ensures that $\tau(C_q)^*$ has non-empty interior. Note that if $\mathcal{P}_q$ has no unbounded subspace, (6.15) is not necessary since the non-empty interior condition will already be ensured by (6.14).

Theorem 6.2.3. Consider a HAS S as in Definition 3.1.4 and points $(h_q \in \mathcal{P}_q)_{q \in V}$. The polynomial $p_q(x, z)$ is feasible for Program 6.2.2 if and only if there exist invariant convex sets $C = (C_q)_{q \in V}$, as defined in Definition 3.3.2, such that $h_q \in C_q$ for each $q \in V$ and $\tau(C_q)^*$ is the 0-sublevel set of $p_q(x, z)$. Moreover, the optimal solution of Program 6.2.2 is the solution that minimizes the sum of the logarithms of the volume of the intersection of the each cone $\tau(C_q)^*$ with the hyperplane $\{ x \mid \langle h_q, x \rangle = 1 \}$.

Proof. Consider a solution $p = (p_q(x,z))_{q\in V}$ of Program 6.2.2. By Corollary 4.7.1, constraints (6.11) and (6.12) are satisfied if and only if there exist ellipsoids C_q such that $\tau(C_q)^*$ is the 0-sublevel set of $p_q(x,z)$. By (4.23), constraint (6.14) is satisfied if and only if $C_q \subseteq \mathcal{P}_q$. By Proposition 1.5.8, constraint (6.13) is satisfied if and only if (6.9) hold for all $q \to_\sigma q'$. Therefore, by Theorem 6.2.2, the solution p is a feasible solution of Program 6.2.2 if and only if the sets C_q are invariant for S.

We now prove the optimality of the solution. By Proposition 4.7.1, there exist Q_q, c_q such that $\mathcal{E}_{Q_q,c_q} = \mathcal{E}_{D_q,d_q,\delta_q}$ and $\lambda_q > 0$ such that $D_q = \lambda_q Q_q$. The volume of the intersection of $\tau(C_q)^*$ with the hyperplane $\{\, x \mid \langle h_q, x\rangle = 1 \,\}$ is $-\det(Q_q)$. Therefore, it remains to show that $\lambda_q = 1$ for an optimal solution. We observe that without the constraint (6.11), for any feasible solution, D_q, d_q, δ_q can be scaled by any positive constant while remaining feasible but affecting the objective function. The Schur complement (see Proposition 1.1.4) of the block D_q of the matrix in the left-hand side of constraint (6.11) is $\delta_q + 1 - d_q^\top D_q^{-1} d_q$ hence constraint (6.11) ensures that

$$d_q^\top D_q^{-1} d_q - \delta_q \le 1.$$

Combining this inequality with equation (4.24) implies that $\lambda_q \le 1$. Since the objective is to maximize $\det(D_q) = \lambda_q \det(Q_q)$, we know that if (D_q, d_q, δ_q) is optimal, then $\lambda_q = d_q^\top D_q^{-1} d_q - \delta_q = 1$. □

Example 6.2.1. We apply Program 6.2.2 to Example 3.3.2 with the same values for the parameters as the ones used in [RMT13], that is, $m_0 = 500\,\mathrm{kg}$, $m = 1000\,\mathrm{kg}$, $k_d = 4600\,\mathrm{N\,s\,m^{-1}}$ and $k_s = 4500\,\mathrm{N\,kg^{-1}}$. The values used for h_q are the same for each node $q \in V$: $u = d_i = 0$ and $v_0 = v_i = (5 + v_a)/2$ for $i = 1, \dots, M$.

We vary the number of trailers M from 1 to 10. Figure 6.1 represents the controlled invariant set at node q_{a0}. As we can see, the constraints on the trailers are propagated to the truck and, as the number M increases, the truck speed and acceleration become more constrained.

The time taken by MOSEK 8.1.0.34 ([ApS17]) to solve the problem is given by Figure 6.2[1].

6.2.3 Polyset template for non-homogeneous systems

In this section, we generalize the results of Section 6.2.2 to polysets. Note that the last variable of the polynomial is the *perspective* variable of the cone defined in (4.21). Therefore, it is not conservative to consider homogeneous

[1] We set $\lambda_{q\to_\sigma q'}$ to 1 for each transition $q \to_\sigma q'$ to make the problem convex.

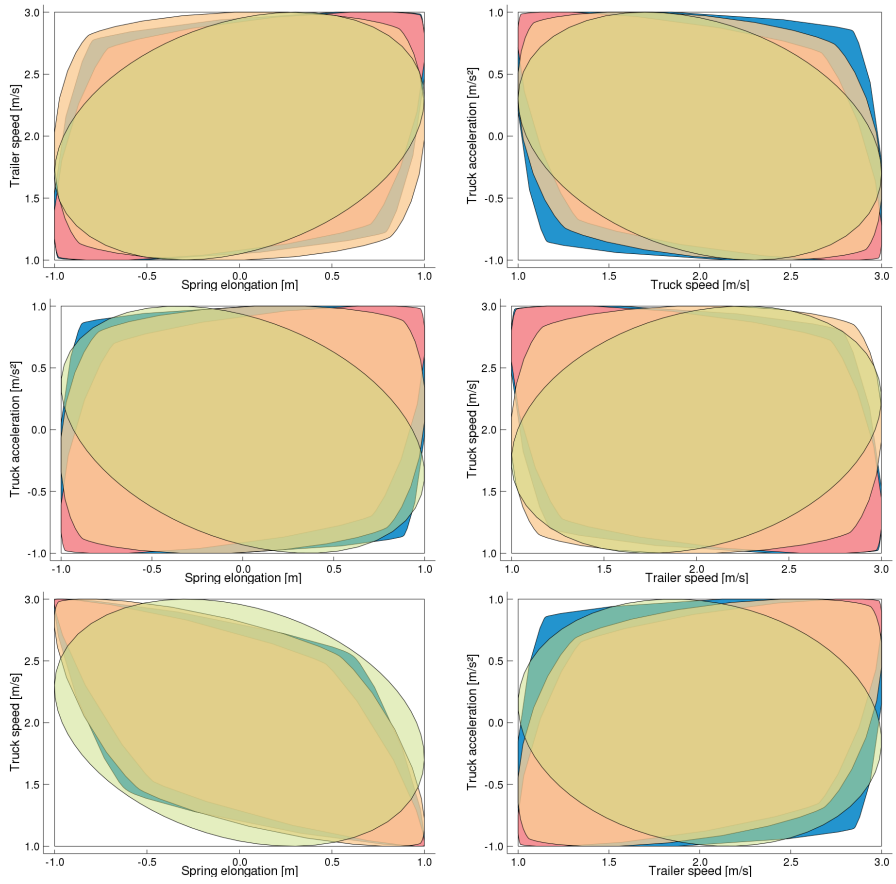


Figure 6.1: Several projections of the optimal solution of Program 6.2.2 (in green) and Program 6.2.3 with the volume heuristic developed in [DHL17a] (in orange for quartic, red for sextic and blue for octic) for Example 6.2.1 at node q_{a0} for various numbers of trailers.

polynomials as cones cannot be the sublevel set of non-homogeneous polynomials.

The main challenge of this generalization resides in constraint (6.11) ensuring both convexity of $\tau(C_q)^*$ and the fact that $\det D_q$ does not overestimate the volume. Indeed, while checking the convexity and computing the volume of ellipsoids can be done easily, it is more involved when using polynomials of higher degree as described in Section 2.3.1. The volume objective for polyset is discussed in Section 4.2.1.

The following program searches invariant polysets of degree $2d$ parametrized by homogeneous polynomials $p_q(x)$. For increasing degree d, the approximation capability of the polysets is improved at the expense of increased computational time; see Section 2.3 for a discussion on the computational com-

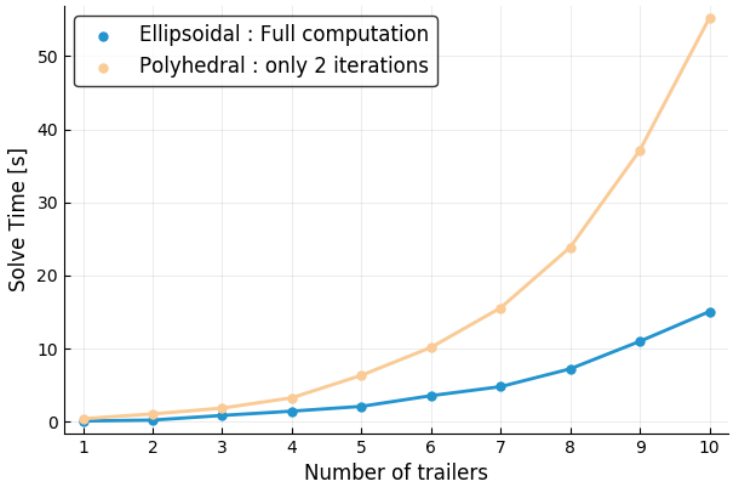


Figure 6.2: Computation time with MOSEK 8.1.0.34 for Program 6.2.2 with Example 6.2.1 with various numbers of trailers compared to two iterations of the polyhedral approach (see e.g., the procedure p. 201 in [BM15]) implemented with the CDD library [Fuk99]. Note that after two iterations, the polyhedral sets obtained are not controlled invariant. One needs to wait for the convergence of the algorithm to obtain a controlled invariant set. Moreover, iterations are usually increasingly slower as the number of facets of the polyhedral sets increases with the iterations.

plexity of Sum-of-Squares programs. We use the notation $p_q(x, z)$ to denote the evaluation of p_q at the vector $y = (x, z)$.

Program 6.2.3.

$$s_q(x,z) + z^{2d} \text{ is SOS} \tag{6.16}$$

$$s_q(x,1) \text{ is SOS-convex} \tag{6.17}$$

$$s_q(x,z) = p_q(H_{h_q}(x,z)), \qquad \forall x \in \mathbb{R}^{n_{q,x}}, z \in \mathbb{R} \tag{6.18}$$

$$p_q(r(A_\sigma, c_\sigma)^\top y) \leq \lambda_{q\to_\sigma q'} p_{q'}(r(E_\sigma, 0)^\top y), \qquad \forall q \to_\sigma q', y \in \mathbb{R}^{n_{q,x}+1} \tag{6.19}$$

$$p_q(-a,\beta) \leq 0, \qquad \forall q \in V, (a,\beta) \in \mathcal{H}_{\text{rep}}(\mathcal{P}_q) \tag{6.20}$$

$$p_q(0,1) < 0, \qquad \forall q \in V. \tag{6.21}$$

The constraint (6.17) ensure the convexity of $\tau(C_q)^*$ and constraints (6.19), (6.20) and (6.21) are identical to the corresponding constraints of Program 6.2.2. The constraint (6.16) is the generalization of (6.11) for polynomials of arbitrary degree. It certifies that $s_q(x, 1) + 1$ is nonnegative which is required for heuristics such as [MLB05; DHL17a] to estimate the volume of its 1-sublevel set.

Theorem 6.2.4. Consider a HAS S as in Definition 3.1.4 and points $(h_q \in \mathcal{P}_q)_{q\in V}$. The polynomial $p_q(x,z)$ is feasible for Program 6.2.3 if and only if there exist invariant convex sets $C = (C_q)_{q\in V}$, as defined in Definition 3.3.2, such that $h_q \in C_q$ for each $q \in V$ and $\tau(C_q)^*$ is the 0-sublevel set of $p_q(x,z)$.

Proof. Consider a solution $p = (p_q(x,z), s_q(x,z))_{q\in V}$ of Program 6.2.3. By (6.17), the 0-sublevel set of $s_q(x,z)$ is convex. As the 0-sublevel set of $p_q(x,z)$ is its image under the Householder transformation, it is also convex. Therefore there exist sets C_q such that $\tau(C_q)^*$ is the 0-sublevel set of $p_q(x,z)$. By (4.23), constraint (6.14) is satisfied if and only if $C_q \subseteq \mathcal{P}_q$. By Proposition 1.5.8, constraint (6.13) is satisfied if and only if (6.9) hold for all $q \to_\sigma q'$. Therefore, by Theorem 6.2.2, the solution p is a feasible solution of Program 6.2.2 if and only if the sets C_q are invariant for S. □

6.3 Model Predictive Control

As mentioned in the introduction, the controlled invariant sets can be used to derive a feedback control law. We illustrate this with a Model Predictive Control (MPC) numerical experiment. We consider a truck with one trailer ($M = 1$) as in Example 6.2.1. The truck starts with speeds $v_0 = v_1 = 2\,\mathrm{m\,s^{-1}}$ and spring elongation $d = 0\,\mathrm{m}$ and has as objective to maximize the distance covered in 20 s. The maximal speed is initially $4\,\mathrm{m\,s^{-1}}$ but after 10 s, it drops to $v_a = 3\,\mathrm{m\,s^{-1}}$.

In a classical MPC controller, the truck acceleration u is controlled by solving a constrained optimal control problem up to horizon H. We observe that if $H \leq 2.5\,\mathrm{s}$, the controller is at some point unable to find values of u satisfying input constraints such that the state remains in the safe set.

For safety-critical applications, this lack of guarantee is not acceptable as it is necessary to be certain that the system can remain in the safe set. Moreover, in a real-time context, the need to pick a large horizon is problematic as it increases the cost of online computations. In our setting, we constrain the state to remain in the controlled invariant sets computed in Example 6.2.1[2] and thereby solve both issues; safety is guaranteed for arbitrarily long simulations and the length of the horizon does not influence safety so smaller length can be used. Note that the controlled invariant sets can be computed offline so if it allows to reduce the horizon length, it enables online computational cost to be moved offline. Besides, constraining the state variables to belong to the controlled invariant sets obtained as solution of Program 6.2.2 or Program 6.2.3 reduces to a convex program as shown by Proposition 2.1.1

[2]Example 6.2.1 corresponds to an MPC controller of horizon 0.8 s. An MPC controller of different horizons computes different controlled invariant sets by updating the hybrid system accordingly.

and Theorem 6.3.1. The results of the experiment can be found in Figure 6.3 and Figure 6.4.

Theorem 6.3.1. Given an SOS-convex polynomial $s(y)$, a point $c \in \mathbb{R}^{n+1}$ such that $s(c) < 0$ and a matrix $H \in \mathbb{R}^{(n+1)\times(n+1)}$, the membership of a vector $x \in \mathbb{R}^n$ to the set C satisfying

$$\tau(C)^* = \{ Hy \mid s(y) \leq 0 \} \tag{6.22}$$

is equivalent to the existence of $\lambda \geq 0$ such that

$$\lambda s(y) - \langle (x, 1), Hy \rangle \text{ is SOS.} \tag{6.23}$$

Proof. By (4.21), the constraint $x \in C$ is equivalent to $(x, 1) \in \tau(C)$ which is equivalent to $\langle (x, 1), u \rangle \ \forall u \in \tau(C)^*$ by definition of duality. Therefore, by (6.22), the constraint $x \in C$ is equivalent to $0 \leq \inf_{y \in \tau(C)^*} \langle (x, 1), Hy \rangle$. By Proposition 2.3.1, this minimization is semidefinite representable. Since c is strictly feasible for this program, strong duality holds and its optimal objective value is equal to the optimal objective value of its dual which gives (6.23). □

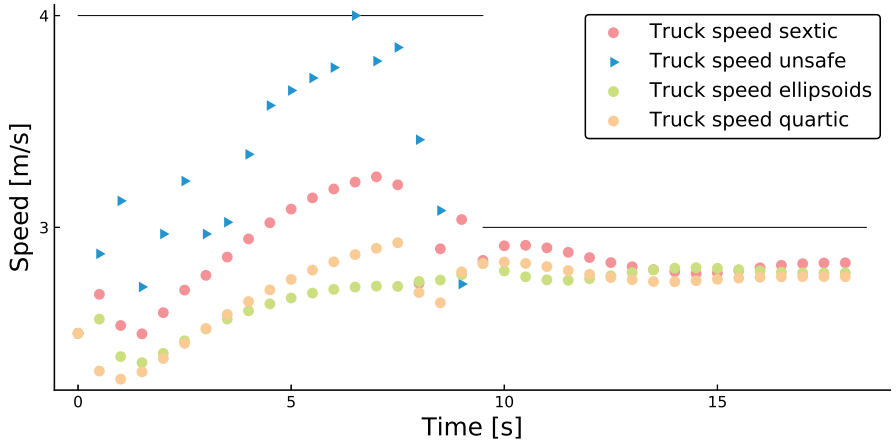


Figure 6.3: Evolution with time of the speed of the truck for various MPC strategies. In the legend, *ellipsoids* (resp. *quartic*, *sextic*) designates our MPC strategy using our computed invariant sets using Program 6.2.2 (resp. Program 6.2.3 with degree 4 and 6 and the volume heuristic developed in [DHL17a]), while *unsafe* designates a classical MPC approach. The piecewise horizontal line represents the speed limitation at time t. One can see that the MPC approach with invariant sets allows to remain in the safe set with a conservativeness that decreases with increasing degree. Moreover, the unsafe controller can fail to find feasible values, as shown in Figure 6.4.

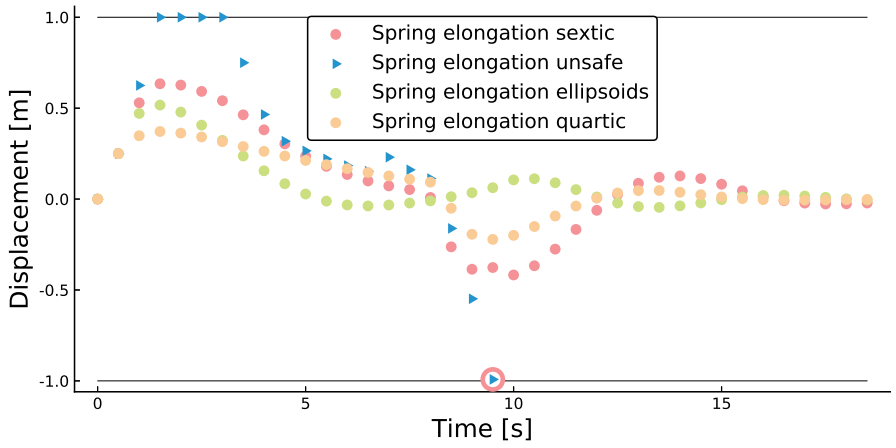


Figure 6.4: Spring elongation in safe and unsafe modes. See Figure 6.3 for the legend syntax. We see (just before $t = 10\,\mathrm{s}$) that the unsafe controller makes the trailer go too close to the truck.

6.4 Stochastic Programming

In stochastic programming, large scale convex programs are decomposed into smaller convex programs linked together by a markov chain [BL11; Pap18]. The convex program at each state u of the markov chain is:

Program 6.4.1.

$$\begin{aligned} Q(x,u) = \text{minimize } & c^T y + \mathcal{Q}_u(y) \\ \text{s.t. } & W_u y = h_u - T_u x, \\ & x \in \mathbb{R}^{n_u}_+ \end{aligned}$$

where the convex function $\mathcal{Q}_u(y)$ is the sum of $Q(x,v)$ for each state v accessible from u weighted by the probability to go from state u to state v. When the program is infeasible for some x, $Q(x,u) = \infty$. At the initial state of the markov chain, there is no term $-T_u x$ and the solution at this stage is the solution of the original large scale convex program. Note that if v is accessible from different states u, u', the number of variables at u and u' must match for $Q(\cdot, v)$ to be well-defined.

A polyhedral lower-approximation of the function $Q(\cdot, u)$ as well as a polyhedron outer-approximation of its domain can obtained Algorithm 1 as described in Section 2.4. The polyhedral approximation of $\mathcal{Q}_u(y)$ is easily obtained from the polyhedral description of $Q(y,v)$ for each state v accessible from u by weighting the generated cuts according to the probabilities of the corresponding transition in the markov chain. This method is called the *Stochastic Dual Dynamic Programming (SDDP)* algorithm.

When Algorithm 1 is guaranteed to never generate any infeasibility cut, the stochastic program is said have *relatively complete recourse* [BL11, p. 92].

Definition 6.4.1. A stochastic program has *relatively complete recourse* if for each state u, vector x, any feasible solution y of Program 6.4.1, and any transition $u \to_\sigma v$ of the markov chain, there exists a feasible solution z of Program 6.4.1 corresponding to $Q(y, v)$.

When a stochastic program does not have relatively complete recourse, the common approach is to transform the constraints causing the infeasibility from hard constraints to soft constraints. That is, they are moved into the objective with a penalization cost when the constraint is violated. As discussed in [SN05], this penalization increases the Lipschitz constant of $Q(x, u)$ which increases the number of iterations needed for SDDP to reach the same accuracy:

> We argue that two stages (linear) stochastic programming problems with recourse can be solved with a reasonable accuracy by using Monte Carlo sampling techniques, while multistage stochastic programs, in general, are intractable.
>
> (...)
>
> In order to avoid such infinite penalizations and to restore the applicability of Theorem 2 one can introduce a finite penalty for infeasibility. In some cases this can reasonably solve the problem. However, in some situations the infeasibility may result in a catastrophic event. In that case the penalty could be huge. Translated into the sample size bounds considered in the previous section, this means huge variances in the estimate (2.19) or huge Lipschitz constant in (2.22), which makes these estimates useless. In a sense, in such situation "nothing works".

In view of this issue, it might be desirable to precompute additional constraints that ensures relatively complete recourse. The relatively complete recourse property is independent of the probabilities on the transitions of the markov chain. For this reason, we consider the hybrid system obtained from the markov chain by replacing stochastic transitions into autonomous switching between the transitions. More precisely, this hybrid system is given in Definition 6.4.2. Note that, as the stochastic uncertainty is replaced by an autonomous switching, the model of uncertainty does not affect the resulting HAS hence the approach admits any uncertainty model for the stochastic program.

Definition 6.4.2. The HAS *representing the feasibility part of* the stochastic program Program 6.4.1 is the HAS such that T is the same automaton than the

markov chain, and for each transition $q \to_\sigma q'$, we have $A_\sigma = -T_{q'}$, $E_\sigma = W_{q'}$, $c_\sigma = h_{q'}$ and $\mathcal{P}_{q'} = \mathbb{R}_+^{n_{q'}}$.

Remark 6.4.1. The markov chain of stochastic program usually have a specific structure made of a root node and D stages of K scenario each. There is a transition from the root to each scenario of each stage and for $d < D$, there is a transition from each scenario of stage d to each scenario of stage $d + 1$. Moreover, it may happen that the programs Program 6.4.1 for corresponding nodes of different stages have the same feasible set. In this case, a smaller HAS can be considered instead. The automaton is the same as in Definition 6.4.2 except that only te root node and the nodes of the first stage are kept and the transitions from nodes a stage d to a stage $d + 1$ are replaced by transitions to nodes of the stage 1 to nodes of the same stage. This automaton is considerably smaller has it only has $K + 1$ nodes instead of $KD + 1$. The invariance condition of the sets of these automaton is strenghtened as they require invariance for infinitely many stages.

Given controlled invariant sets $\mathcal{S}_u$ of HAS, we adapt Program 6.4.1 using the sets as follows.

Program 6.4.2.

$$\begin{aligned} Q(x,u) = \text{minimize } & c^T y + Q_u(y) \\ \text{s.t. } & W_u y = h_u - T_u x, \\ & x \in \mathcal{S}_u \end{aligned}$$

Theorem 6.4.1. Consider a HAS representing the feasibility part of a stochastic program Program 6.4.1. If the sets $\mathcal{S}_q$ are invariant for the HAS, as defined in Definition 3.3.2, then the stochastic program Program 6.4.2 has relatively complete recourse.

Proof. For each state u, vector x, any feasible solution y of Program 6.4.2, and any transition $u \to_\sigma v$, by Definition 3.0.3, there exists a solution $z \in \mathcal{S}_v$ such that $W_v z = -T_v y + h_v$. We observe that z is feasible for the Program 6.4.2 corresponding to $Q(y, v)$ hence the Program 6.4.2 has relatively complete recourse. □

Remark 6.4.2. With piecewise ellipsoidal or piecewise polyset templates for the sets $\mathcal{S}_q$, the conic patition can be designed using infeasibility cuts. More precisely, if Program 6.4.1 does not have relatively completely recourse then the SDDP algorithm may encouner infeasibilities and can then generate infeasibility cuts. These infeasibilities or infeasibility cuts provide a hint on where the feasible set should be refined to be invariant or rather where it matters to refine it for the purpose of solving Program 6.4.1, i.e. taking into account the

directional objective given by c in Program 6.4.1. It seems therefore appropriate to refine the partition in these directions similarly to how the polytope resulting from the first fixed point iteration was used in Example 4.0.1.

6.5 Conclusion

We have developed a methodology for computing controlled invariant sets in Section 4.4 and this section develops the method for continuous-time control systems, Discrete-Time Affine Hybrid Control System (HCS) and Discrete-Time Affine Hybrid Algebraic System (HAS) with *autonomous switching* (see Remark 3.3.2). This method can be combined with semidefinite programming in order to compute ellipsoidal controlled invariant sets. We have shown that our technique can be used as a building block in a model predictive control scheme in Section 6.3. This allows, among other things, to reduce the online computational cost by precomputing controlled invariant sets. In Section 6.4, we have also introduced a potential application to stochastic programming which is an essential tool used in the energy market nowadays. While this seems promising, it remains to test the approach on an actual application coming from the energy market and show the advantages of the method suggested.

We feel that we have only scratched the surface of the potential of the duality correspondence of Section 4.4. Many extensions of this work are possible such as hybrid systems with controlled switching.

The reformulation of the computation of controlled invariant sets of hybrid control system to the computation of invariant sets of hybrid algebraic system with Proposition 3.3.1 and Proposition 3.3.2 allows to have a more behavioral invariance relation. In the future, we would like to put our results in the framework of behavioral theory in order to investigate how to further generalize them; see [WP13].

Entropic cone

7

In 1948, Shannon published "A Mathematical Theory of Communication" [Sha48]. In this paper, Shannon introduces the entropy of a random variable. Suppose we have a random variable X of alphabet $\mathcal{X}$, he defines the entropy of X as

$$H_b(X) = \sum_{x \in \mathcal{X}} \Pr[X = x] \log_b \frac{1}{\Pr[X = x]}$$

where the basis b is positive. If b is 2 (resp. e), the unit is the bits (resp. nats). Note that $H_b(X) = H_a(X) \log_b(a)$ so the entropies using different bases are equivalent up to a positive constant factor.

This quantity represents the "amount of information contained in X". More precisely, suppose Alice and Bob know the probability distribution of X. They can meet and set up a communication protocol. Then Alice will see realizations x of X and will need to tell Bob the value x using their transmission protocol. The entropy of X is the minimum, over all possible protocols, of the *expected* number of characters of a language of b symbols needed to transfer one value.

An optimal protocol is given the entropic coding: for this protocol, each value x is encoded in $\log_2 \frac{1}{\Pr[X=x]}$ bits.

The entropy of several random variables in simply the entropy of their cartesian product:

$$\begin{aligned} H_b(\{X_1, \dots, X_n\}) &= H_b((X_1, \dots, X_n)) \\ &= \sum_{(x_1,\dots,x_n) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n} p(x_1, \dots, x_n) \log_b \frac{1}{p(x_1, \dots, x_n)} \end{aligned}$$

where $p(x_1, \dots, x_n) = \Pr[(X_1, \dots, X_n) = (x_1, \dots, x_n)]$. By convention, we say that the entropy of an empty set of random variables is 0.

Given a n random variables, we can compute the entropy of any of the 2^n subset of those n variables. The entropic vector of a set of n random variables is a vector h, indexed by the subsets of $[n] = \{1, \dots, n\}$, such that $h_S = H_b(\{ X_i \mid i \in S \})$.

We denote the set of vectors of $\mathbb{R}^{2^n-1}$ that are entropic as:

$$\mathcal{H}_n = \{ h \in \mathbb{R}^{2^n-1} \mid \exists X_1, \dots, X_n, \forall \emptyset \neq S \subseteq [n], h_S = H_b(\{ X_i \mid i \in S \}) \}.$$

This set was shown to be a convex cone in [ZY97, Theorem 1]. We do not include the dimension corresponding to the entropy of the empty set as it is zero. We will see that thanks to this choice, the cone $\mathcal{H}_n$ so that it is solid, i.e. full-dimensional.

The entropic cone has many applications including Network coding [Bas+13; CG07; DFZ05], secret sharing [Bei11; MP07], guessing games [Bab+13], quantum information [Pip03], conditional independence [MS95], Additive combinatorics [MMT10] and Group Theory [CY02].

The set $\mathcal{E}_n \triangleq \mathbb{R}^{2^n-1}$ of vectors indexed by the nonempty subsets of $[n]$ is called the *entropy space* and its elements are called *entropy vectors*. Entropy vectors can also be viewed as function of $[n] \setminus \emptyset \to \mathbb{R}$. A special class of entropy vectors will be encountered quite often so we will give them a special name.

Definition 7.0.1. For a given set $I \subseteq [n]$ and a number m, we define the *entropy vector*

$$r_I^m = J \mapsto \min(m, |J \cap I|).$$

That is, in the index notation, if $h = r_I^m$ then $h_J = \min(m, |J \cap I|)$.

Note that we do not specify the number of variables n for r as most of the time, it will be clear from the context.

As it is often done in the litterature [Cha11; MC13], we will use shortened notations for set. For instance, the singleton $\{i\}$ will be denoted as i and the set $\{i, j, k\}$ will be denoted as ijk.

7.1 The entropic cone of 1 variable

Consider first one binary random variable X with alphabet $\mathcal{X} = \{0, 1\}$ and $\Pr[X = 1] = \alpha$ for some $0 \leq \alpha \leq 1$. We have $H_b(X) = -\alpha \log_b(\alpha) - (1 - \alpha) \log_b(1 - \alpha) \triangleq H_b(\alpha)$. The function $H_b(\alpha)$ is shown in Figure 7.1. We see that depending on α, $H_b(X)$ can take any value between 0 and $H_b(\frac{1}{2})$ or, more formally, between r_1^0 and $r_1^{H_b(\frac{1}{2})}$. This observation leads to the following two questions.

1. Can the entropy take negative values ?
2. Can the entropy take values higher than $H_b(\frac{1}{2})$?

To answer the first question, we generalize the entropy function H_b.

Definition 7.1.1. Consider a fixed basis b and two probability mass functions p, q over the *same* alphabet $\mathcal{X}$. The *entropy function* $H_p(q)$ is defined as

$$H_p(q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{q(x)}.$$

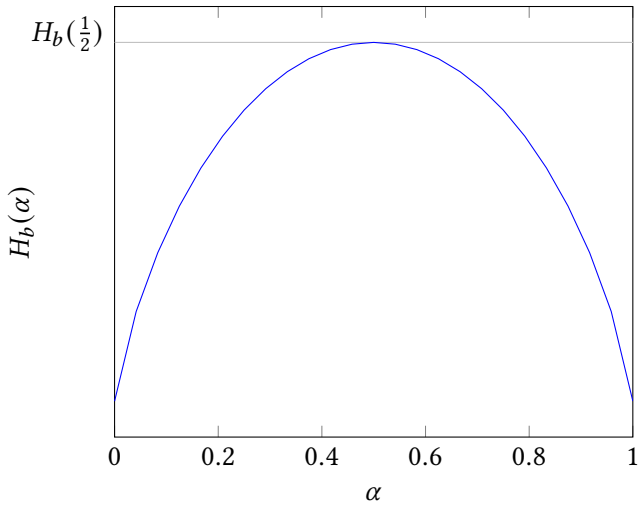


Figure 7.1: The value of $H_b(\alpha)$ for $0 \leq \alpha \leq 1$.

The notation $H_p(p)$ will sometimes be shortened as $H(p)$ or even $H_b(p)$ so specify the base.

Denote the support of p, or a random variable X with probability mass function p, as $\operatorname{supp}(X) = \operatorname{supp}(p) = \{ x \in \mathcal{X} \mid p(x) > 0 \}$. The nonnegativity of $x \mapsto -\log(x)$ for $x \leq 1$ implies the following property of $H_q(p)$.

Proposition 7.1.1. For any probability mass functions p, q over the same alphabet,

$$H_p(q) \geq 0$$

with equality if and only if $\operatorname{supp}(p) \cap \operatorname{supp}(q) = \emptyset$ or $|\operatorname{supp}(p)| = 1$.

In particular, if we take p as the joint probability mass function of a set of random variables I, Proposition 7.1.1 on $H_p(p)$ gives the following corollary.

Corollary 7.1.1. For any set of random variables X_I,

$$H(X_I) \geq 0$$

with equality if and only if all random variables X_i, $i \in I$, are deterministic.

Corollary 7.1.1 answers the first questions, we now turn our attention to the second one. As one can anticipate, using a larger alphabet, we can get a higher entropy. For instance, if $\mathcal{X} = \{00, 01, 10, 11\}$ where the first and second bits of the characters are independent and have the value 1 with probability p_1 and p_2, then the entropy is $H(p_1) + H(p_2)$.

This observation can be generalized as follows.

Lemma 7.1.1 (The entropic cone is close to addition)**.** If $g, h \in \mathcal{H}_n$, then $g + h \in \mathcal{H}_n$.

Proof. Let X (resp. Y) be a random variables with entropy vector g (resp. h). We define the variables $Z_i = (X_i, Y_i)$ with X_i and Y_i independent, that is, $\Pr[Z_i = (x, y)] = \Pr[X_i = x] \Pr[Y_i = y]$. The entropy vector of Z is $g + h$. □

In particular, we have the following corollary.

Corollary 7.1.2. If $h \in \mathcal{H}_n$, then for any nonnegative integer m, $mh \in \mathcal{H}_n$.

We can now conclude. We have seen that $[0, H_b(\frac{1}{2})] \subseteq \mathcal{H}_1$ which, using Corollary 7.1.2, implies that $\mathbb{R}^+ \subseteq \mathcal{H}_1$. From Corollary 7.1.1, we have $\mathcal{H}_1 \subseteq \mathbb{R}^+$ hence $\mathcal{H}_1 = \mathbb{R}^+$.

7.2 Kullback-Leibler divergence and convexity

Kullback and Leibler introduces the so-called Kullback-Leibler divergence [KL51] between two probability mass functions p, q over the *same* alphabet $\mathcal{X}$:

$$\begin{aligned} D_{\mathrm{KL}}(p\|q) &= H_p(q) - H_p(p) \\ &= \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}. \end{aligned}$$

We recall here the Jensen's inequality.

Theorem 7.2.1 (Jensen's inequality)**.** If $f : \mathbb{R}^n \to \mathbb{R}^n$ is convex on its domain then for any $x_1, \ldots, x_n \in \operatorname{dom} f$ and $\lambda_1, \ldots, \lambda_n \geq 0$ such that $\lambda_1 + \cdots + \lambda_n = 1$,

$$f(\lambda_1 x_1 + \cdots + \lambda_n x_n) \leq \lambda_1 f(x_1) + \cdots \lambda_n f(x_n). \tag{7.1}$$

Moreover if f is strictly convex, the equality occurs if and only if $x_1 = \cdots = x_n$ or one of the $\lambda_i = 1$.

Since $x \mapsto -\log x$ is strictly convex on its domain $\mathbb{R}^+$, the Kullback-Leibler divergence has the following property.

Proposition 7.2.1. For any probability mass functions p, q over the same alphabet,

$$D_{\mathrm{KL}}(p\|q) \geq 0$$

with equality if and only if $p = q$.

Proof. By Jensen's inequality on $-\log x$,

$$\begin{aligned} D_{\mathrm{KL}}(p\|q) &= -\sum_{x\in\mathcal{X}} p(x)\log\frac{q(x)}{p(x)} \\ &\geq -\log\sum_{x\in\mathcal{X}} p(x)\frac{q(x)}{p(x)} \\ &= -\log q(x) \\ &= 0. \end{aligned}$$

If $D_{\mathrm{KL}}(p\|q) = 0$, we have either $p(x) = 1$ for some $x \in \mathcal{X}$ or $q(x)/p(x) = q(x')/p(x')$ for any $x, x' \in \mathcal{X}$. In the first case, $-\log q(x) = 0$ so $q(x) = 1$ and $p = q$. In the second case, $q(x)/q(x') = p(x)/p(x')$ for any $x, x' \in \mathcal{X}$ so p and q are proportional. Since $\sum_{x\in\mathcal{X}} p(x) = 1 = \sum_{x\in\mathcal{X}} q(x)$, they are equal. □

7.3 The Shannon inequalities

We will see further than $\operatorname{cl}\mathcal{H}_n$ is a convex cone. We can therefore define its dual $(\operatorname{cl}\mathcal{H}_n)^*$, which is[1]

$$\{\, h^* \in \mathcal{E}_n^* \mid \langle h^*, h\rangle \geq 0, \forall h \in \mathcal{H}_n \,\}$$

where the dual entropy space $\mathcal{E}_n^*$ is simply $\mathbb{R}^{2^n-1}$. The dual $(\operatorname{cl}\mathcal{H}_n)^*$ is the set of all linear inequalities satisfied for all entropy vector of $\mathcal{H}_n$.

The set of Shannon inequalities is a subset of the linear inequalities satisfied by all entropic vector. That is, it is a cone $\mathcal{P}_n^* \subseteq (\operatorname{cl}\mathcal{H}_n)^*$. The cone $\mathcal{P}_n^*$ is polyhedral as it is given by all the linear inequalities that are consequence of a finite number of linear inequalities.

These inequalities are the following ones:

normalized

$$h_\emptyset = 0, \tag{7.2}$$

nonnegative For any I,

$$h_I \geq 0. \tag{7.3}$$

nondecreasing If $I \subseteq J$, then

$$h_I \leq h_J. \tag{7.4}$$

submodular For all J, K,

$$h_J + h_K \geq h_{J\cup K} + h_{J\cap K}. \tag{7.5}$$

Or equivalently: If $I \subseteq J$, then for any K,

$$h_{K\cup I} - h_I \geq h_{K\cup J} - h_J. \tag{7.6}$$

[1]Note that we do not need the closure in the definition of $(\operatorname{cl}\mathcal{H}_n)^*$

We can see that (7.3) is a consequence of (7.2) and (7.4). The inequality (7.3) is a direct consequence of Proposition 7.1.1 and the inequality (7.4) is a consequence of Proposition 7.1.1 and Proposition 7.3.1.

Proposition 7.3.1. For any $I \subseteq J$,

$$H_b(X_J) - H_b(X_I) = \sum_{x \in \mathcal{X}_I} \Pr[X_I = x] H_b(X_J | X_I = x)$$

where $X_J|X_I = x$ is the random variable of alphabet $\mathcal{X}_{J\setminus I}$ with probability mass function $y \mapsto \Pr[X_J = y|X_I = x]$.

Proof. We have

$$\begin{aligned} H(X_J) - H(X_I) &= \sum_{x \in \mathcal{X}_J} \Pr[X_J = x] \log \frac{\Pr[X_I = x_I]}{\Pr[X_J = x]} \\ &= \sum_{x \in \mathcal{X}_J} \Pr[X_J = x] \log \frac{1}{\Pr[X_J = x | X_I = x_I]} \\ &= \sum_{x \in \mathcal{X}_I} \Pr[X_I = x] \sum_{y \in \mathcal{X}_J} \Pr[X_J = y | X_I = x] \log \frac{1}{\Pr[X_J = y | X_I = x]}. \end{aligned}$$

□

Theorem 7.3.1. For any $I \subseteq J$,

$$\langle \Delta_{J|I}, h \rangle \triangleq h_J - h_I \geq 0$$

with equality if and only if $J \setminus I$ is deterministic when I is known.

The submodularity is the consequence of Proposition 7.2.1 and the following relation:

$$H_b(J) + H_b(K) - H_b(I) = H_b(p_{J|I} p_{K|I} p_I) \tag{7.7}$$

where $I = J \cap K$.

Theorem 7.3.2. If an entropy vector h is entropic and $J \cap K = I$ then

$$\langle \Delta_{J,K|I}, h \rangle \triangleq h_J + h_K - h_{J\cup K} - h_I \geq 0$$

with equality if and only if $p = p_{J|I} p_{K|I} p_I$, that is, $X_{J|I}$ and $X_{K|I}$ are independent.

7.4 The entropic cone of 2 variables

The entropic cone of 2 variables lives in $\mathbb{R}^3$. Its superset $\mathcal{P}_3$ has 3 extreme rays (or 2 up to symmetry): r_1^1, r_2^1 and r_{12}^1. The first one, r_1^1 is entropic, it corresponds to a deterministic random variable X_2 and a coin flip random variable X_1. The third one is also entropic, it corresponds to a X_1 and X_2 being equal to the same coin flip.

Corollary 7.1.2 shows that mr is also entropic for each of our 3 rays r and any nonnegative integer m. In Section 7.1 we could also prove that αr_1^1 was entropic for any nonnegative number α.

We can use that fact and symmetry to prove that both the rays generated by r_1^1 and r_2^1 are entropic. To prove that the ray generated by r_{12}^1 is entropic, we can do the same to the variable $X_1 = X_2$.

This technique can be generalized. The idea is to decompose an entropy vector into independent components with positive entropy and then use the fact that each independent component is entropic using the fact that $\mathcal{H}_1 = \mathbb{R}_+$. This may not be entirely clear because we applied this technique to examples that were too trivial. For this reason, we now present a more insightful example.

Example 7.4.1. Suppose we are given the entropy vector $h = 2r_1^1 + 2.5r_2^1 + 4.2r_{12}^1$. From the fact that $\mathcal{H}_1 = \mathbb{R}_+$, we can define 3 random variables Y_1, Y_2, Y_3 with $H_b(Y_1) = 2$, $H_b(Y_2) = 2.5$ and $H_b(Y_3) = 4.2$. Now, taking $X_1 = (Y_1, Y_3)$ and $X_2 = (Y_2, Y_3)$, the entropy vector of X is h.

The numbers $2, 2.5$ and 4.2 of the previous example are sometimes called the atoms of an entropy vector. If those atoms are nonnegative, we can define independent variables at each atom using the fact that $\mathcal{H}_1 = \mathbb{R}_+$, we can show that the vector is entropic. Those atoms are commonly represented in a Venn diagram; see Figure 7.2.

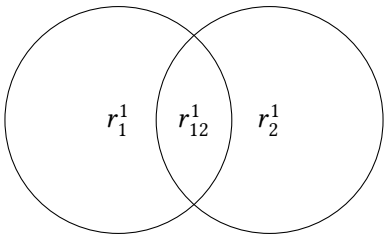


Figure 7.2: The decomposition of an entropy vector of 2 variables in its atoms.

In the general case of entropy vectors of n variables, the $2^n - 1$ vectors r_I^1 provides a base for the entropy space. The matrix of the change of basis

is unimodular and can be obtained as follows. Suppose $h = \sum_{I \in [n]\setminus\emptyset} \alpha_I r^1_I$, we have

$$h_J = \sum_{I \subseteq J} \alpha_I$$
$$\alpha_I = \sum_{J \subseteq I} (-1)^{|I\setminus J|} (h_{[n]} - h_{[n]\setminus J}).$$

The first identity is easily obtained using the Venn diagram while the second identity can be obtained using the inclusion exclusion principle.

Definition 7.4.1. The *cone of positive atoms* $\mathcal{A}_n$ is defined as

$$\mathcal{A}_n = \{ \sum_{I \subseteq [n]\setminus\emptyset} \alpha_I r^1_I \mid \alpha \geq 0 \}.$$

As we have seen, using the fact that $\mathcal{H}_1 = \mathbb{R}_+$, we have the following theorem.

Theorem 7.4.1. For any natural number n,

$$\mathcal{A}_n \subseteq \mathcal{H}_n.$$

Since $\mathcal{A}_n \subseteq \mathcal{H}_n \subseteq \mathcal{P}_n$ and $\mathcal{A}_2 = \mathcal{P}_2$, we have $\mathcal{A}_2 = \mathcal{H}_2 = \mathcal{P}_2$.

7.5 The entropic cone of 3 variables

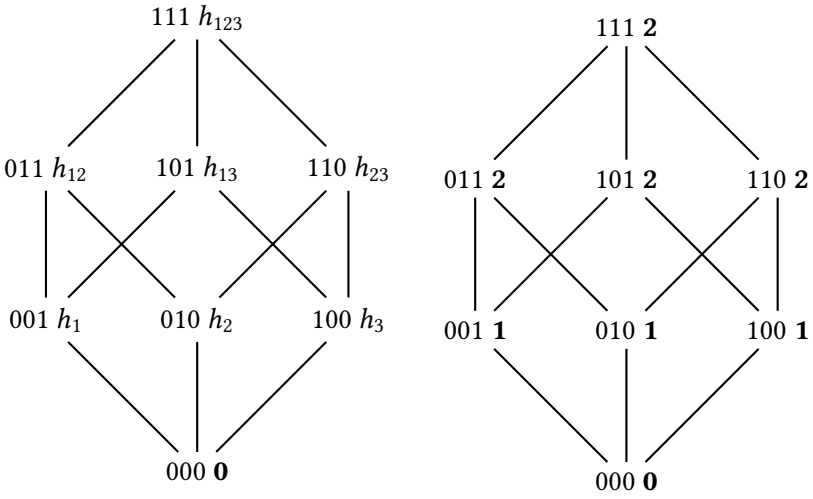
For 3 variables, the relation $\mathcal{A}_3 = \mathcal{P}_3$ does not hold anymore. The extreme rays of $\mathcal{P}_3$ are the extreme rays of $\mathcal{A}_3$ except for $r^2_{[3]} \in \mathcal{P}_3$ that is not $\mathcal{A}_3$; see Figure 7.3b for an illustration of $r^2_{[3]}$. This vector is entropic for $b = 2$ as shown by Example 7.5.1.

Example 7.5.1. Consider the three independent coin tosses X_1, X_2 and $X_3 = X_1 \oplus X_2$ where $\oplus$ is the XOR operator. One can verify that the entropy vector of those random variables is $H_2(X) = r^2_{[3]}$.

Now, to conclude that "$\mathcal{H}_3 = \mathcal{P}_3$", we would like to show that $\mathcal{H}_n$ is a cone. That is, we would like to generalize Corollary 7.1.2 and say that $\mathcal{H}_n$ is closed to multiplication with any nonnegative number. However, this cannot be shown and $\mathcal{H}_n$ is not a cone for $n \geq 3$.

This is shown by Example 7.5.2 that is taken from [ZY97].

Example 7.5.2 ([ZY97])**.** Consider again $r^2_{[3]}$. We can see that $r^2_{[3]}$ is supported by the following dual entropy vector: $\Delta_{1,2}, \Delta_{2,3}, \Delta_{3,1}, \Delta_{1|23}, \Delta_{2|13}, \Delta_{3|12}$.



(a) Lattice representation of an entropy vector h.

(b) $r^2_{[3]}$.

Figure 7.3: Lattice representation of the entropy vector of $\mathcal{E}_3$.

Consider a positive multiple $h = \lambda r^2_{[3]}$ for some $\lambda > 0$. It is still supported by the same dual vectors so by Theorem 7.3.2, X_1, X_2, X_3 are pairwise independent. Therefore,

$$p_{12} = p_1 p_2$$
$$p_{13} = p_1 p_3$$
$$p_{23} = p_2 p_3.$$

Moreover, by Theorem 7.3.1, there exists functions $x_1(x_2, x_3), x_2(x_1, x_3), x_3(x_1, x_2)$ such that $p_{123} = p_{12}\delta_{x_3(x_1,x_2)} = p_{13}\delta_{x_2(x_1,x_3)} = p_{23}\delta_{x_1(x_2,x_3)}$.

That is, for any x_1, x_2, x_3,

$$p_1(x_1)p_2(x_2)\delta_{x_3(x_1,x_2)}(x_3) = p_1(x_1)p_3(x_3)\delta_{x_2(x_1,x_3)}(x_2) = p_2(x_2)p_3(x_3)\delta_{x_1(x_2,x_3)}.$$

We see with the first inequality that for any x_2, x_3, choosing $x_1 = x_1(x_2, x_3)$, we have

$$p_1(x_1)p_2(x_2) = p_1(x_1)p_3(x_3)$$

hence $p_2(x_2) = p_3(x_3)$. That is, X_1, X_2 and X_3 are uniform and have alphabets of the same length. Therefore, the only entropy vectors of the ray of $r^2_{[3]}$ that are entropic are the vectors

$$h = \log_b(m) r^2_{[3]}.$$

for a positive integer m.

We saw that the full dimensional polyhedron $\mathcal{A}_n$ is included in $\mathcal{H}_n$ but we have just seen that outside this polyhedron, $\mathcal{H}_n$ might not be a cone. At this point one might be wondering how much "porous" $\mathcal{H}_n$ is. It turns out that it is not that porous, there are just barely some "scratch" on the boundary. More precisely, the closure of $\mathcal{H}_n$ is a convex cone (Theorem 7.5.1) and the relative interior of its closure is entropic (Theorem 7.5.2).

Lemma 7.5.1. Consider two alphabets $\mathcal{X}_1, \mathcal{X}_2$ with $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$ and two probability mass functions p_1 and p_2 respectively defined on $\mathcal{X}_1$ and $\mathcal{X}_2$. If $0 \leq \alpha \leq 1$, the probability mass function

$$p(x) = \begin{cases} \alpha p_1(x), & x \in \mathcal{X}_1 \\ (1-\alpha)p_2(x), & x \in \mathcal{X}_2 \end{cases}$$

has an entropy equal to

$$H_b(p) = \alpha H_b(p_1) + (1-\alpha)H_b(p_2) + H_b(\alpha).$$

Corollary 7.5.1. If $h, g \in \mathcal{H}_n$ then for any $0 \leq \alpha \leq 1$, $\alpha h + (1-\alpha)g + H_b(\alpha)r^n_{[n]} \in \mathcal{H}_n$.

In particular, taking $g = 0$, and using Corollary 7.1.2, we have the following corollary.

Corollary 7.5.2 ([Mat07, Lemma 4])**.** If $h \in \mathcal{H}_n$ then for any $\alpha, \beta > 0$, $\alpha h + \beta r^n_{[n]} \in \mathcal{H}_n$.

And finally, since we can take β arbitrarily close to zero in the previous corollary.

Corollary 7.5.3 ([Mat07, Corollary 2])**.** If $h \in \mathcal{H}_n$ then for any nonnegative number α, $\alpha h \in \mathrm{cl}(\mathcal{H}_n)$.

Corollary 7.5.3 and Lemma 7.1.1 implies the following theorem.

Theorem 7.5.1. For integer n, $\mathrm{cl}(\mathcal{H}_n)$ is a convex cone.

As mentioned earlier the only difference between $\mathrm{cl}(\mathcal{H}_n)$ and $\mathcal{H}_n$ is on the boundary of $\mathcal{H}_n$.

Theorem 7.5.2. For any integer n,

$$\mathrm{ri}(\mathrm{cl}(\mathcal{H}_n)) \subseteq \mathcal{H}_n.$$

This theorem is the consequence of the fact that $\mathcal{H}_n$ contains the full dimensional cone $\mathcal{A}_n$ as shown by the following lemma that is taken from the proof of [Mat07, Theorem 1].

Lemma 7.5.2. Consider a set $\mathcal{S}$. If $\mathrm{cl}(\mathcal{S})$ is a convex cone, $\mathcal{S}$ contains a full dimensional cone K and $\mathcal{S}$ is close to addition then

$$\mathrm{ri}(\mathrm{cl}(\mathcal{S})) \subseteq \mathcal{S}.$$

Proof. Suppose $x \in \mathrm{ri}(\mathrm{cl}(\mathcal{S}))$. Since x is in the relative interior, $(x-K) \cap \mathrm{cl}(\mathcal{S})$ should not be empty. Since K is full dimensional, that means that $(x-K) \cap \mathcal{S}$ is also nonempty. Let $y \in (x-K) \cap \mathcal{S}$. Since $\mathcal{S}$ contains K and is closed to addition, $y + K \subseteq \mathcal{S}$ and in particular $x \in \mathcal{S}$. □

7.6 The entropic cone of 4 variables

The entropic cone of 4 variables has, in additions to $\mathcal{A}_3$, the extreme rays $r^2_{[3]}, r^2_{[4]}, r^3_{[4]}$ and the three rays given by Figure 7.4 (up to symmetry). The vector $r^2_{[3]}$ is entropic, as seen in Section 7.6. The vector $r^3_{[4]}$ is also entropic, it corresponds to X_1, X_2, X_3 being random coin tosses and $X_4 = X_1 \oplus X_2 \oplus X_3$. The vector $r^2_{[4]}$ is entropic too but it is less easy to see.

7.7 Sculpting the Entropic Cone

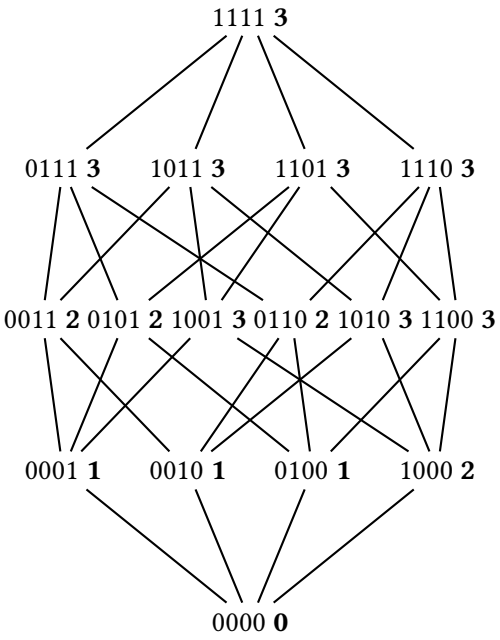
In general, $\mathcal{H}_n \subseteq \mathcal{P}_n$. We have seen that $\mathcal{H}_1 = \mathcal{P}_1, \mathcal{H}_2 = \mathcal{P}_2, \mathrm{cl}\,\mathcal{H}_3 = \mathcal{P}_3$ but as we will see, $\mathrm{cl}\,\mathcal{H}_n \subset \mathcal{P}_n$ for $n \geq 4$.

The outer cone $\mathcal{P}_n$ is obtained using Theorem 7.3.1 and Theorem 7.3.2. As $\mathcal{P}_4$ is a strict superset of $\mathrm{cl}\,\mathcal{H}_4$, that means that those theorems are not enough. What have we missed ? Do we need a new method to generate new entropic inequalities ? It turns out that we can generate new inequalities using exactly the same method that we used to produce the outer bound $\mathcal{P}_n$.

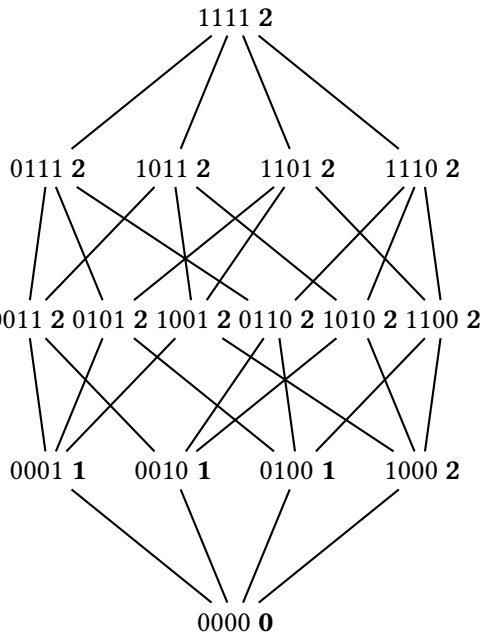
7.7.1 Generating non-shannon inequalities

The method used to generate $\mathcal{P}^*_n$ is as follows:

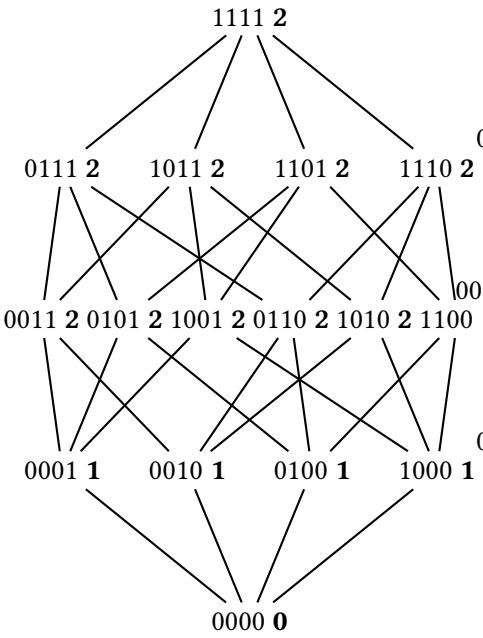
1. We consider an entropic vector h and its probability mass function.
2. We apply transformations on the probability mass function.
3. Then use the fact that $x \mapsto -\log(x)$ is nonnegative for $0 \leq x \leq 1$ using Proposition 7.1.1 and the fact that $x \mapsto -\log(x)$ is convex using Proposition 7.2.1 to derive inequalities between the generated probability mass functions. That is, we enforce $H_p(q) \geq 0$ and $D(p\|q) \geq 0$ between all pairs p, q.
4. Then we look at the implications of those inequalities on the coordinates of the entropy vector h.



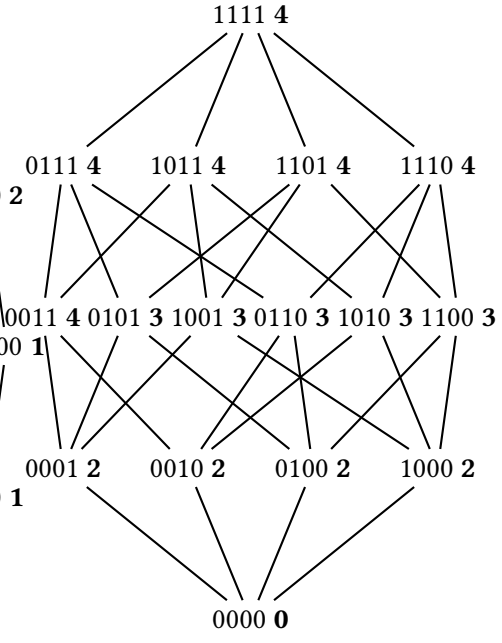
(a) X_1, X_2, X_3 **are independent coin tosses and** $X_4 = (X_1 \oplus X_3, X_2 \oplus X_3)$.



(b) X_1 **and** X_2 **are independent coin tosses,** $X_3 = X_1 \oplus X_2$ **and** $X_4 = (X_1, X_2)$.



(c) X_1 **and** X_2 **are independent coin tosses and** $X_4 = X_3 = X_1 \oplus X_2$.



(d) Non-entropic extreme ray of $\mathcal{P}_4$**. Its free expansion is the Vamos matroid [Vam68].**

Figure 7.4: Extreme rays of $\mathcal{H}_4$ **not expressible using the** r_S^m **notation.**

We can consider different transformations:

Marginalization For $I \subseteq [n]$, we can transform the probability mass function p to

$$\mathcal{X}_I \to \mathbb{R} : x \mapsto p_I(x_I).$$

Contraction For $I \subseteq J$ and some x_I, we can transform the probability mass function p to

$$\mathcal{X}_{J\setminus I} \to \mathbb{R} : y \mapsto p_{J|I}(y|x_I).$$

Inner-adhesivity For J, K with $I = J \cap K$, we can transform the probability mass function p to

$$\mathsf{ia}_{J,K|I}(p) : \mathcal{X}_{J\cup K} \to \mathbb{R} : x \mapsto \frac{p_J(x_J)p_K(x_K)}{p_I(x_I)} = p_{J|I}(x_J|x_I)p_{K|I}(x_K|x_I)p_I(x_I). \tag{7.8}$$

We denote $\mathsf{ia}_{J,K|J\cap K}$ as $\mathsf{ia}_{J,K}$.

Self-adhesivity For $I \subseteq J$, we can transform the probability mass function p to

$$\mathsf{sa}_{J|I}(p) : \mathcal{X}\times\mathcal{X}_{J\setminus I} \to \mathbb{R} : x \mapsto \frac{p_{J'}(x_{J'})p_J(x_K)}{p_I(x_I)} = p_{J'|I}(x_{J'}|x_I)p_{J|I}(x_K|x_I)p_I(x_I). \tag{7.9}$$

where $n' = n + |J \setminus I|$, $J' = [n]$ and $K = ([n'] \setminus [n]) \cup I$

One could consider additional types of transformations. For instance, for J, K, L with $I = J \cap K \cap L$, we can transform the probability mass function p to

$$\mathcal{X}_{J\cup K\cup L} \to \mathbb{R} : x \mapsto p_{J|I}(x_J|x_I)p_{K|I}(x_K|x_I)p_{L|I}(x_L|x_I)p_I(x_I).$$

However this transformation can be achieved by doing two inner-adhesivity transformations: one with $(J, K \cup L)$ and then one with $(J \cup K, L)$.

We can see that Theorem 7.3.1 was obtained using contraction and then Proposition 7.1.1 and Theorem 7.3.2 was obtained using inner-adhesivity and then Proposition 7.2.1.

If we apply a transformation on a probability mass function that is already a transformation of p, we might not be able to express the resulting probability mass function using p without using summation. For instance, suppose we do an inner-adhesivity transformation using the sets 12 and 13 with a probability mass function p of 3 variables: $q = \mathsf{ia}_{12,13}(p) = p_{2|1}p_{3|1}p_1$. If we marginalize q on the second variable, it is $q_{13} = p_{13}$ but if we marginalize on the first variable, it is

$$\sum_{x_1\in\mathcal{X}_1} q(x_1, x_2, x_3) = \sum_{x_1\in\mathcal{X}_1} p_{2|1}(x_2|x_1)p_{3|1}(x_3|x_1)p_1(x_1)$$

but we cannot simplify it further. This happens when we marginalize on a variable of I after an inner-adhesivity. Suppose q is obtained after applying an inner-adhesivity transformation on p with (J, K) of intersection $I = J \cap K$. Let h (resp. g) be the entropic vector of p (resp. q). Suppose we want to compute g_L in terms of h.

- If $L \cap J = I$ or $L \cap K = I$, then $g_L = h_L$.

- Otherwise, if $I \subseteq L$,

$$g_L = h_{L\cap J} + h_{L\cap K} - h_I.$$

  Note that the above equation gives $g_L = h_L$ if $L \cap J = I$ or $L \cap K = I$, it was only split in two cases for clarity.

- Otherwise, q_L is obtained from q using marginalization on one of the variables in I and g_L cannot be expressed in terms of h.

Even if some coordinates of g cannot be expressed in terms of g, those coordinates can still be used to derive inequalities on h.

To illustrate this technique, we show in the following example how to prove the Zhang-Yeung inequality.

Example 7.7.1. One form of the Zhang-Yeung inequality is as follows:

$$\langle \Delta_{3,4|1} + \Delta_{3,4|2} + \Delta_{1,2} - \Delta_{3,4} + \Delta_{3,4|5} + \Delta_{5,3|4} + \Delta_{4,5|3}, h\rangle \geq 0$$

Let $q = \mathrm{ia}_{5,12|34}p$. Let us abbreviate $H(p)$ with p. The Zhang-Yeung in-

equality is:

$$
\begin{aligned}
&-p_{134} + \mathrm{ia}_{3,4|1}p - p_{234} + \mathrm{ia}_{3,4|2}p - p_{12} + \mathrm{ia}_{1,2}p - p_{32} \\
&+\mathrm{ia}_{3,4}p - p_{345} + \mathrm{ia}_{3,4|5}p - p_{345} + \mathrm{ia}_{5,3|4}p - p_{345} + \mathrm{ia}_{4,5|3}p \geq 0 \\
&-p_{134} + p_{13} + p_{14} - p_1 - p_{234} + p_{23} + p_{24} - p_2 - p_{12} + p_1 + p_2 - p_{32} + p_3 + p_4 \\
&-p_{345} + p_{35} + p_{45} - p_5 - p_{345} + p_{45} + p_{34} - p_4 - p_{345} + p_{34} + p_{35} - p_3 \geq 0 \\
&+p_{13} + p_{35} - p_3 + p_{14} + p_{45} - p_4 \\
&+p_{23} + p_{35} - p_3 + p_{24} + p_{45} - p_4 \\
&-p_5 - p_{134} - p_{345} + p_{34} - p_{234} - p_{345} + p_{34} - p_{1234} - p_{345} + p_{34} + p_{1234} - p_{12} \geq 0 \\
&+p_{13} + p_{35} - p_3 + p_{14} + p_{45} - p_4 \\
&+p_{23} + p_{35} - p_3 + p_{24} + p_{45} - p_4 \\
&-p_5 - q_{1345} - q_{2345} - q + p_{1234} - p_{12} \geq 0 \\
&-q_{135} + p_{13} + p_{35} - p_3 - q_{145} + p_{14} + p_{45} - p_4 \\
&-q_{235} + p_{23} + p_{35} - p_3 - q_{245} + p_{24} + p_{45} - p_4 \\
&-q_{125} + p^1_{15} + q_{25} - p_5 - q_{1345} + q_{135} + q_{145} - q_{15} \\
&-q_{2345} + q_{235} + q_{245} - q_{25} - q + q_{125} + p_{1234} - p_{12} \geq 0 \\
&-q_{135} + \mathrm{ia}_{1,5|3}q - q_{145} + \mathrm{ia}_{1,5|4}q - q_{235} + \mathrm{ia}_{2,5|3}q - q_{245} + \mathrm{ia}_{2,5|4}q \\
&-q_{125} + \mathrm{ia}_{1,2|5}q - q_{1345} + \mathrm{ia}_{3,4|15}q - q_{2345} + \mathrm{ia}_{3,4|25}q - q + \mathrm{ia}_{34,5|12}q \geq 0
\end{aligned}
\tag{7.10}
$$

$$
\begin{aligned}
&D(q_{135} \| \mathrm{ia}_{1,5|3}q) + D(q_{145} \| \mathrm{ia}_{1,5|4}q) + D(q_{235} \| \mathrm{ia}_{2,5|3}q) + D(q_{245} \| \mathrm{ia}_{2,5|4}q) \\
&+D(q_{125} \| \mathrm{ia}_{1,2|5}q) + D(q_{1345} \| \mathrm{ia}_{3,4|15}q) + D(q_{2345} \| \mathrm{ia}_{3,4|25}q) + D(q \| \mathrm{ia}_{34,5|12}q) \geq 0.
\end{aligned}
\tag{7.11}
$$

The last inequality is true by Proposition 7.2.1 so the Zhang-Yeung inequality is proven.

We used the following probability mass function which are not expressible in terms of p without summation.

$$
\begin{aligned}
q_{135} &= \sum_{4} p_{5|34} p_{1|34} p_{34} \\
q_{145} &= \sum_{3} p_{5|34} p_{1|34} p_{34} \\
q_{235} &= \sum_{4} p_{5|34} p_{2|34} p_{34} \\
q_{245} &= \sum_{3} p_{5|34} p_{2|34} p_{34} \\
q_{125} &= \sum_{34} p_{5|34} p_{12|34} p_{34} \\
q_{15} &= \sum_{34} p_{5|34} p_{1|34} p_{34} \\
q_{25} &= \sum_{34} p_{5|34} p_{2|34} p_{34}.
\end{aligned}
$$

The Zhang-Yeung inequality is a *non-Shannon inequality* as it is not a consequence of the inequalities introduced in Section 7.3. It was discovered by Zhang and Yeung in 1998 [ZY98]. The existence of this non-Shannon inequality shows that the inclusion $\operatorname{cl} \mathcal{H}_n \subset \mathcal{P}_n$ is strict for $n \geq 4$.

Matúš and Csirmaz [MC13] recently introduced a way to visualize the projection of a cut of approximations of the Entropic Cone of 4 random variables in 3 dimensions. In this visualization, the polymatroid cone $\mathcal{P}_4$ is the tetrahedron shown in Figure 7.5b.

(a) Polymatroid cone $\mathcal{P}_4$.

(b) The polymatroid cone $\mathcal{P}_4$ cut by the Zhang-Yeung inequality.

Figure 7.5: 3D visualization of approximations of the Entropic Cone of 4 random variables.

7.7.2 Formalizing the generation of non-shannon inequalities

In the previous section, we have seen how to generate new inequalities in terms of transformation on the probability mass function. In practice, instead of looking at all those transformations together and all the possible succession of those transformations, a specific transformation is chosen and we only keep its implication on the entropy space. This may look conservative to only look at it from the entropy space but it makes it easier to reason about and to compute non-shannon inequalities.

The following theorems translate the invariance over probability mass functions to the entropy space.

The contraction of a polymatroid is defined similarly to the contraction on matroid initially studied by Tutte [Tut58a; Tut58b]; see also [Wel10, Section 4.3] or [Oxl06, Section 3].

Definition 7.7.1 (Contraction)**.** The *contraction* of a polymatroid h onto $J \subset [n]$, denoted by $h \cdot J$ is defined as

$$(h \cdot J)_K = h_{K \cap I} - h_I, \qquad K \subseteq J$$

where $I = [n] \setminus J$. It is also called the contraction of I from h and denoted h/I.

The following theorem is a consequence of Proposition 7.3.1 and Theorem 7.5.1.

Theorem 7.7.1 ([MC13, Lemma 1])**.** The family of cones $\operatorname{cl} \mathcal{H}_n$ is closed under contraction.

We define the following operators

$$\begin{aligned} \mathsf{ia}_{J,K|I}(h) &= \{\, g \in \mathcal{E}_{n'} \mid \forall I \subseteq L \subseteq J \cap K, g_L = h_{L\cap J} + h_{L\cap K} - h_I \,\}, \quad I = J \cap K, \\ \mathsf{sa}_{J|I}(h) &= \{\, g \in \mathcal{E}_{n'} \mid \forall I \subseteq L \subseteq J' \cap K, g_L = h_{L\cap J'} + h_{L\cap K} - h_I \,\}, \quad I \subseteq J, \end{aligned}$$

where for inner-adhesivity, $n' = |J \cup K|$ and for self-adhesivity $n' = n + |J \setminus I|$, $J' = [n]$ and $K = ([n'] \setminus [n]) \cup I$.

Again, we denote $\mathsf{ia}_{J,K|J\cap K}$ as $\mathsf{ia}_{J,K}$.

Definition 7.7.2. We say that a family of sets $\mathcal{S}_n \subseteq \mathcal{E}_n$ is *inner-adhesive* if for any n, $x \in \mathcal{S}_n$ and $J, K \subseteq [n]$, there exists $y \in \mathcal{S}_{|J\cup K|}$ such that $y \in \mathsf{ia}_{J,K}(x)$ and we say that it is *self-adhesive* if for any n, $x \in \mathcal{S}_n$ and $I \subseteq J \subseteq [n]$, there exists $y \in \mathcal{S}_{n+|J\setminus I|}$ such that $y \in \mathsf{sa}_{J|I}(x)$.

The following theorem is a consequence of equation (7.7), and the transformation (7.8) and (7.9).

Theorem 7.7.2. The cone $\mathcal{H}_n$ is inner-adhesive and self-adhesive.

An important numerical quantity, related with the geometric properties of $\mathcal{H}_4$ is the *Ingleton score* defined as

$$\mathbb{I}^* \triangleq \inf_{0 \neq h \in \mathcal{H}_4} \mathbb{I}_{ij}(h) \tag{7.12}$$

where $\mathbb{I}_{ij}(h) = \langle \square_{ij}, h \rangle / h_{[n]}$ [DFZ11, Definition 3] and $\square_{ij}$ is the Ingleton dual entropy vector [Ing71]. The Ingleton score of an approximation of the entropic cone of four variables is proportional to the "height" of the 3D visualization we introduced. The Ingleton score of $\mathcal{P}_4$ represented in Figure 7.5a is $-\frac{1}{4}$ and the Ingleton score of the approximation represented in Figure 7.5b is $-\frac{1}{6}$.

The current best lower bound on $\mathbb{I}^*$ is equal to -0.15789 [DFZ11]. Upper bounds on $\mathbb{I}^*$ can be obtained from exhibiting four jointly distributed variables for which the entropy vector has low Ingleton score. The current best upper bound on $\mathbb{I}^*$ is equal to -0.09243 [MC13].

The current best lower and upper bound are given respectively by the outer and inner approximation of $\mathcal{P}_4$ illustrated by Figure 7.6.

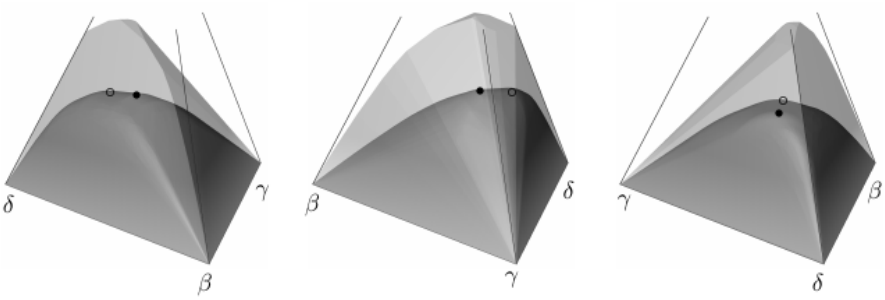


Figure 7.6: Outer [DFZ11] and inner [MC13] approximations of $\mathcal{P}_4$. The figures comes from [MC13].

7.8 Hierarchy of set programs

We saw in Section 7.7 with Theorem 7.7.2 that the entropic cone is both inner-adhesive and self-adhesive. Those adhesive invariance are inclusion constraints of the form (4.7) between entropic cones of different dimensions. Consider the maximal cones included in the polymatroid cone $\mathcal{P}_i$ for each number i of random variables, that are invariant under inner-adhesivity and self-adhesivity. It is unknown whether these cones correspond to the entropic cones. However, as the entropic cone is invariant under both inner-adhesivity and self-adhesivity, these cones provide outer approximations of the entropic cones. Moreover, these cones are the solution of the generic set program with the countably infinite set variables $\mathcal{S}_1, \mathcal{S}_2, \ldots$ where $\mathcal{S}_i$ corresponds to

the outer approximation of the entropic cone of i random variables with the inclusion constraints $\mathcal{S}_i \subseteq \mathcal{P}_i$ and the inclusion constraints of the form (4.7) corresponding the inner-adhesivity and self-adhesivity.

Consider the set program obtained by removing the variables $\mathcal{S}_i$ for $i > N$ and the constraints involving the sets that were removed. The set programs for each N form a hierarchy of set programs such that the optimal solution for $\mathcal{S}_i$ corresponding to the set program for N_1 is included in the optimal solution for $\mathcal{S}_i$ corresponding to the set program for N_2 if $i < N_2 < N_1$. That is, with increasing N, the outer approximation is closer to the entropic cone.

Solving the set program for polyhedral templates may need a prohibitive number of halfspaces or rays in the representation as $\mathcal{S}_i$ has dimension $2^i - 1$. However, for the purpose of optimization along a fixed direction in the entropic cone, as discussed in Section 4.2.2, we can circumvent this issue by only refining the initial outer approximation $\mathcal{P}_i$ where it matters for the optimization. To verify this, we used Algorithm 3 to find lower bounds for the Ingleton score and obtained the best known lower bound -0.15789 in under a minute. The feasibility cuts generated to find this lower bound are represented in Figure 7.7. Comparing Figure 7.7 with the outer approximation of Figure 7.6, we see that, as expected, our approach only generates cuts close to the direction corresponding to the Ingleton score.

Figure 7.7: The polymatroid cone $\mathcal{P}_4$ cut by the feasibility cuts uncovered to find the lower bound -0.15789 of the Ingleton score.

7.9 Conclusion

This chapter gives a glimpse of the potential of set programming for a given problem. Until now, only polyhedral approximation of the entropic cone was considered. However, when formulating the problem as a set program, we notice that a much wider range of methods can be utilized. In view of the specificity of the problem, certain templates or methods seem more suitable: as the entropic cone of 4 variables is piecewise linear in a significant part of the space, piecewise semi-ellipsoidal sets or piecewise polysets seem appropriate. If the entropic cone only needs to be closely approximated in specific directions then polyhedral templates can be considered as well with Algorithm 3 to refine the approximation where it is relevant.

Using Algorithm 3, we found the same lower bound on the Ingleton score than the the one obtained in [DFZ11]. Both methods use the polyhedral template so finding the same lower bound is not unexpected but our methods finds it under a minute while hours of computations were needed in [DFZ11]. Running Algorithm 3 longer should provide an improved lower bound but the current implementation did not yet provide any. Several implementation decisions have a significant impact on the time taken to obtain the lower bounds:

- when an infeasibility cut is found, should it be shared to other nodes for which it is valid (even if it may not be in a direction that matters for this node);
- should infeasibility cuts be kept forever or should old ones be dropped when their corresponding dual variables seem to repeatedly be zero (too many cuts may render the optimization too slow);
- when an infeasibility is found, should the "early abortion" described in Section 4.2.2 be applied ?
- as discussed in Section 4.2.2, in which order should the elements in the set I of Algorithm 3.

We believe that if the impact of these choices should be analysed carefully, a better lower bound should be obtainable in a reasonable time.

Conclusion 8

In this thesis, we have explored the insights provided by analyzing set programs at the level of generic sets or generic convex sets. We analyzed how these observations particularize to set programs for specific templates and how it affects their computability.

In Chapter 1, we introduced the different operations on sets that can be combined in the inclusion constraints of a set program. We showed the properties related to these operations and the different representations of convex sets. These results were then particularized for polyhedras, zonotopes, ellipsoids and specific subclasses of semialgebraic sets.

In Chapter 2, we detailed the optimization tools needed for computing the sets of the different templates including for instance the verification of membership of a point in a set and inclusion of one sets in another set.

In Chapter 3, we introduced the different dynamical systems and the condition for the invariance of sets for each one. The classical algebraic methods to compute invariant sets were described.

In Chapter 4, we describe set programs using the operations defined in Chapter 1 and discuss the computation of solutions of such programs at the generic level depending on the operations used in the constraints. We then translate this to each particular template.

In Chapter 5, we focus on the accuracy guarantees of the polyset template for feasibility set programs with linear inclusion constraints. The guarantee is based on the entropy of the language generated by the automaton underlying the constraint structure. We also provide a rounding technique for generating infeasibility certificates of the generic set program given infeasibility certificates of the set program on polyset templates of a fixed degree. We then show a way to approximate the set of constraints to remove in order to render the set program feasible while maximizing the entropy of the resulting language.

In Chapter 6, the geometric approach to set programs developed in Chapter 4 is used to provide a geometric interpretation for the semidefinite programs used to compute ellipsoidal controlled invariant sets. The crucial advantage of this geometric approach over the algebraic approach presented in Chapter 3 is the fact that it is carried out at the abstract level of generic convex sets. Therefore, it directly provides semidefinite and sum-of-squares program formulations for controlled invariant sets of any template that satisfy certain

properties. For instance, it provides a semidefinite program for computing controlled invariant piecewise semi-ellipsoidal sets and sum-of-squares program for sets with polynomial or even piecewise polynomial support function. As highlighted in the introduction, the representation of a set highly depends on the intended end-use. To show this, we detail how the controlled invariant set can be used in two different applications: model predictive control and stochastic programming.

While the computation of invariant sets in Systems and Control is naturally formulated as a single set program, we showed in Chapter 7 that the challenging problem of the approximation of the entropic cone can be formulated as a hierarchy of set programs. The results of Chapter 4 applied for this hierarchy of set programs provide a novel method for optimizing on the entropic cone.

We hope this thesis has motivated the advantages of studying the generic set program at an abstract level before focusing on a specific template. Using Table 8.1, it allows to select the appropriate representation depending on the operations used in the set program in a template-independent manner. Then, using Table 8.2, the appropriate templates can be selected and the performance of the several applicable templates can be compared in order to select the most suitable one depending on the performance and accuracy of the computation as well as the usability and quality for the intended end-use of the set.

	$\cap$	$\operatorname{conv}\cup$	$+$	$\sharp$	A	A^{-1}
Gauge function	max	$\square$	$\sharp$	$+$		$\circ A$
Support function	$\square$	max	$+$	$\sharp$	$\circ A^\top$	
Polar set	$\operatorname{conv}\cup$	$\cap$	$\sharp$	$+$	$A^{-\top}$	$A^\top$

Table 8.1: Given an operation on convex sets specified by the column header, the corresponding operation on the gauge function, support function or polar set is provided in the table.

Many open questions remain to be investigated. We only scratched the surface of the possible inclusion constraints with (4.4), (4.5), (4.6) and (4.7) and many other possible inclusion constraints can be considered. For instance, we could consider inclusions involving the other operations of Table 8.2 but also inclusion of complements. Note that the inclusion of $\mathcal{S}$ in the complement of $\mathcal{T}$ is equivalent to the inclusion of $\mathcal{T}$ in the complement of $\mathcal{S}$ and to $\mathcal{S} \cap \mathcal{T} = \emptyset$. Moreover, the linear functions involved in (4.4), (4.5), (4.6) and (4.7) could be generalized to homogeneous functions or even *generalized homogeneous functions*, see [Pol20] for a definition of generalized homogeneity.

Template	$\cap$	conv $\cup$	$+$	$\sharp$	A	A^{-1}	$\circ$
Polyhedra	✓	✓	✓	✓	✓	✓	✓
Zonotopes	✗	✗	✓	✗	✓	✗	✗
Ellipsoids	✗	✗	✗	✗	✓	✓	✓
Polysets	✗	✗	✗	✗	✗	✓	✗
Piecewise semi-ellipsoids	✗	✗	✗	✗	✓	✓	✓
Piecewise polysets	✗	✗	✗	✗	✗	✓	✗

Table 8.2: This table highlights the invariance of the different templates studied in this thesis under the operations preserving convexity that we consider. Each row represents a template and each column represents an operation. A ✓ symbol indicates that the template is invariant under the operation, a ✗ symbol indicates that it is not. A similar table can be found in [KA20, Table 1].

Bibliography

[ABS95] D. Avis, D. Bremner, and R. Seidel. "How good are convex hull algorithms?" In: *Proceedings of the eleventh annual symposium on Computational geometry*. ACM. 1995, pp. 20–28.

[AG15] A. A. Ahmadi and O. Günlük. "Robust-to-dynamics linear programming". In: *IEEE 54th Annual Conference on Decision and Control (CDC)*. IEEE. 2015, pp. 5915–5919.

[AG18] A. A. Ahmadi and O. Gunluk. "Robust-to-Dynamics Optimization". In: *arXiv e-prints*, arXiv:1805.03682 (May 2018), arXiv:1805.03682. arXiv: 1805.03682 [math.OC].

[Ahm+13] A. A. Ahmadi, A. Olshevsky, P. A. Parrilo, and J. N. Tsitsiklis. "NP-hardness of deciding convexity of quartic polynomials and related problems". In: *Mathematical Programming* 137.1-2 (2013), pp. 453–476.

[Ahm+14] A. A. Ahmadi, R. M. Jungers, P. A. Parrilo, and M. Roozbehani. "Joint spectral radius and path-complete graph Lyapunov functions". In: *SIAM Journal on Control and Optimization* 52.1 (2014), pp. 687–717.

[AJ+13] A. A. Ahmadi, R. Jungers, et al. "SOS-convex Lyapunov functions with applications to nonlinear switched systems". In: *Proceedings of the IEEE Conference on Decision and Control*. 2013.

[AJ13] A. A. Ahmadi and R. M. Jungers. "Switched stability of nonlinear systems via sos-convex lyapunov functions and semidefinite programming". In: *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE. 2013, pp. 727–732.

[AJ16] N. Athanasopoulos and R. M. Jungers. "Computing the Domain of Attraction of Switching Systems Subject to Non-Convex Constraints". In: *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. HSCC '16. Vienna, Austria: ACM, 2016, pp. 41–50. ISBN: 978-1-4503-3955-1.

[AKK65] R. L. Adler, A. G. Konheim, and M. H. Konheim. "Topological entropy". In: *Transactions of the American Mathematical Society* 114.2 (1965), pp. 309–319.

[Alu+95] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. "The algorithmic analysis of hybrid systems". In: *Theoretical computer science* 138.1 (1995), pp. 3–34.

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993. ISBN: 0-13-617549-X.

[And+99] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' guide*. Vol. 9. Siam, 1999.

[AP12a] A. A. Ahmadi and P. A. Parrilo. "A convex polynomial that is not sos-convex". In: *Mathematical Programming* 135.1-2 (2012), pp. 275–292.

[AP12b] A. A. Ahmadi and P. A. Parrilo. "Joint spectral radius of rank one matrices and the maximum cycle mean problem." In: *CDC*. 2012, pp. 731–733.

[AP13] A. A. Ahmadi and P. A. Parrilo. "A complete characterization of the gap between convexity and SOS-convexity". In: *SIAM Journal on Optimization* 23.2 (2013), pp. 811–833.

[ApS17] M. ApS. *MOSEK Optimization Suite release 8.1.0.43*. URL: `http://docs.mosek.com/8.1/intro.pdf`. 2017.

[ApS19] M. ApS. *MOSEK Optimization Suite release 8.1.0.82*. URL: `http://docs.mosek.com/8.1/intro.pdf`. 2019.

[AS98] T. Ando and M.-h. Shih. "Simultaneous Contractibility." In: *SIAM Journal on Matrix Analysis & Applications* 19.2 (1998), p. 487. ISSN: 08954798.

[Avi00] D. Avis. "A revised implementation of the reverse search vertex enumeration algorithm". In: *Polytopes — combinatorics and computation*. Springer. 2000, pp. 177–198.

[Bab+13] R. Baber, D. Christofides, A. N. Dang, S. Riis, and E. R. Vaughan. "Multiple unicasts, graph guessing games, and non-Shannon inequalities". In: *Network Coding (NetCod), 2013 International Symposium on*. IEEE. 2013, pp. 1–6.

[Bac19] F. Bach. "Submodular functions: from discrete to continuous domains". In: *Mathematical Programming* 175.1-2 (2019), pp. 419–459.

[Bal92] K. Ball. "Ellipsoids of maximal volume in convex bodies". In: *Geometriae Dedicata* 41.2 (1992), pp. 241–250.

[Bal97] K. Ball. "An elementary introduction to modern convex geometry". In: *Flavors of geometry* 31 (1997), pp. 1–58.

[Bar+12] 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*. ACM. 2012, pp. 307–326.

[Bar85] B. R. Barmish. "Necessary and sufficient conditions for quadratic stabilizability of an uncertain system". In: *Journal of Optimization theory and applications* 46.4 (1985), pp. 399–408.

[Bas+13] R. Bassoli, H. Marques, J. Rodriguez, K. W. Shum, and R. Tafazolli. "Network coding theory: A survey". In: *Communications Surveys & Tutorials, IEEE* 15.4 (2013), pp. 1950–1978.

[BCR13] J. Bochnak, M. Coste, and M.-F. Roy. *Real algebraic geometry*. Vol. 36. Springer Science & Business Media, 2013.

[BDH96] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. "The quickhull algorithm for convex hulls". In: *ACM Transactions on Mathematical Software (TOMS)* 22.4 (1996), pp. 469–483.

[Bei11] A. Beimel. "Secret-sharing schemes: a survey". In: *Coding and cryptology*. Springer, 2011, pp. 11–46.

[Bez+17] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. "Julia: A fresh approach to numerical computing". In: *SIAM review* 59.1 (2017), pp. 65–98.

[BGL01] C. I. Byrnes, T. T. Georgiou, and A. Lindquist. "A generalized entropy criterion for Nevanlinna-Pick interpolation with degree constraint". In: *IEEE Transactions on Automatic Control* 46.6 (2001), pp. 822–839.

[Bia+16] A. A. Bian, B. Mirzasoleiman, J. M. Buhmann, and A. Krause. "Guaranteed non-convex optimization: Submodular maximization over continuous domains". In: *arXiv preprint arXiv:1606.05615* (2016).

[BKM88] A. Brook, D. Kendrick, and A. Meeraus. "GAMS, a user's guide". In: *ACM Signum Newsletter* 23.3-4 (1988), pp. 10–11.

[BL00] R. W. Brockett and D. Liberzon. "Quantized feedback stabilization of linear systems". In: *IEEE transactions on Automatic Control* 45.7 (2000), pp. 1279–1289.

[BL11] J. R. Birge and F. Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[Bla99] F. Blanchini. "Set invariance in control". In: *Automatica* 35.11 (1999), pp. 1747–1767.

[BM15] F. Blanchini and S. Miani. *Set-theoretic methods in control*. Second. Springer, 2015.

[BN01] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, 2001.

[BN05] V. D. Blondel and Y. Nesterov. "Computationally efficient approximations of the joint spectral radius". In: *SIAM Journal on Matrix Analysis and Applications* 27.1 (2005), pp. 256–272.

[Bor99] B. Borchers. "CSDP, AC library for semidefinite programming". In: *Optimization methods and Software* 11.1-4 (1999), pp. 613–623.

[Bow71] R. Bowen. "Entropy for group endomorphisms and homogeneous spaces". In: *Transactions of the American Mathematical Society* 153 (1971), pp. 401–414.

[Boy+94] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*. Vol. 15. SIAM, 1994.

[BPT12] G. Blekherman, P. A. Parrilo, and R. R. Thomas. *Semidefinite Optimization and Convex Algebraic Geometry*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2012. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9781611972290.

[BT18] P. Breiding and S. Timme. "HomotopyContinuation. jl: A package for homotopy continuation in Julia". In: *International Congress on Mathematical Software*. Springer. 2018, pp. 458–465.

[BTV03] V. D. Blondel, J. Theys, and A. A. Vladimirov. "An elementary counterexample to the finiteness conjecture". In: *SIAM Journal on Matrix Analysis and Applications* 24.4 (2003), pp. 963–970.

[BV04] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BW92] M. A. Berger and Y. Wang. "Bounded semigroups of matrices". In: *Linear Algebra and its Applications* 166 (1992), pp. 21–27.

[BYZ00] S. J. Benson, Y. Ye, and X. Zhang. "Solving Large-Scale Sparse Semidefinite Programs for Combinatorial Optimization". In: *SIAM Journal on Optimization* 10.2 (2000), pp. 443–461.

[BZ09] L. T. Biegler and V. M. Zavala. "Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization". In: *Computers & Chemical Engineering* 33.3 (2009), pp. 575–582.

[CDH16] M. Claeys, J. Daafouz, and D. Henrion. "Modal occupation measures and LMI relaxations for nonlinear switched systems control". In: *Automatica* 64 (2016), pp. 143–154.

[CG06] S. Campi and P. Gronchi. "On volume product inequalities for convex sets". In: *Proceedings of the American Mathematical Society* 134.8 (2006), pp. 2393–2402.

[CG07] T. Chan and A. Grant. "Entropy vectors and network codes". In: *arXiv preprint cs/0702063* (2007).

[Cha11] T. Chan. "Recent progresses in characterising information inequalities". In: *Entropy* 13.2 (2011), pp. 379–401.

[CLO15] D. A. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms*. Springer, 2015.

[CLR95] M.-D. Choi, T. Y. Lam, and B. Reznick. "Sums of squares of real polynomials". In: *Proceedings of Symposia in Pure mathematics*. Vol. 58. American Mathematical Society. 1995, pp. 103–126.

[CP11] A. Chambolle and T. Pock. "A first-order primal-dual algorithm for convex problems with applications to imaging". In: *Journal of mathematical imaging and vision* 40.1 (2011), pp. 120–145.

[CY02] T. H. Chan and R. W. Yeung. "On a relation between information inequalities and group theory". In: *Information Theory, IEEE Transactions on* 48.7 (2002), pp. 1992–1995.

[Dai12] X. Dai. "A Gel'fand-type spectral radius formula and stability of linear constrained switching systems". In: *Linear Algebra and its Applications* 436.5 (2012), pp. 1099–1113.

[DB16] S. Diamond and S. Boyd. "CVXPY: A Python-embedded modeling language for convex optimization". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 2909–2913.

[DD09] P. Dreesen and B. De Moor. "Polynomial Optimization Problems are Eigenvalue Problems". In: *Model-Based Control: Bridging Rigorous Theory and Advanced Technology*. Ed. by P. M. Hof, C. Scherer, and P. S. Heuberger. Boston, MA: Springer US, 2009, pp. 49–68. ISBN: 978-1-4419-0895-7. DOI: 10.1007/978-1-4419-0895-7_4.

[DFZ05] R. Dougherty, C. Freiling, and K. Zeger. "Insufficiency of linear coding in network information flow". In: *Information Theory, IEEE Transactions on* 51.8 (2005), pp. 2745–2759.

[DFZ11] R. Dougherty, C. Freiling, and K. Zeger. "Non-Shannon information inequalities in four random variables". In: *arXiv preprint arXiv:1104.3602* (2011).

[DHL17a] F. Dabbene, D. Henrion, and C. M. Lagoa. "Simple approximations of semialgebraic sets and their applications to control". In: *Automatica* 78 (2017), pp. 110–118.

[DHL17b] I. Dunning, J. Huchette, and M. Lubin. "JuMP: A modeling language for mathematical optimization". In: *SIAM Review* 59.2 (2017), pp. 295–320.

[DPW96] C. Durieu, B. T. Polyak, and E. Walter. "Trace versus determinant in ellipsoidal outer-bounding, with application to state estimation". In: *IFAC Proceedings Volumes* 29.1 (1996), pp. 3975–3980.

[Els95] L. Elsner. "The generalized spectral-radius theorem: an analytic-geometric proof". In: *Linear Algebra and its Applications* 220 (1995), pp. 151–159.

[Faw18] H. Fawzi. "On representing the positive semidefinite cone using the second-order cone". In: *Mathematical Programming* (2018), pp. 1–10.

[Fek23] M. Fekete. "Über die Verteilung der Wurzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten". In: *Mathematische Zeitschrift* 17.1 (1923), pp. 228–249.

[FGK90] R. Fourer, D. M. Gay, and B. W. Kernighan. "A modeling language for mathematical programming". In: *Management Science* 36.5 (1990), pp. 519–554.

[FPR08] A. Ferrante, M. Pavon, and F. Ramponi. "Hellinger versus Kullback–Leibler multivariable spectrum approximation". In: *IEEE Transactions on Automatic Control* 53.4 (2008), pp. 954–967.

[Fuk03] K. Fukuda. "CDD–A C-implementation of the double description method". In: *Institut fur Operations Research, ETH Zurich, available via anonymous ftp: ifor13. ethz. ch, directory pub/fukuda/cdd* (2003).

[Fuk96] K. Fukuda. "Note on New Complexity Classes ENP, EP and CEP—an Extension of the Classes NP Co-NP and P". In: *ETH Zürich, Institute for Operations Research, Zürich June* 12 (1996), p. 1996.

[Fuk99] K. Fukuda. "cdd/cdd+ Reference Manual". In: *Institute for Operations Research, ETH-Zentrum* (1999).

[Gal+93] G. Gallo, G. Longo, S. Pallottino, and S. Nguyen. "Directed hypergraphs and applications". In: *Discrete applied mathematics* 42.2-3 (1993), pp. 177–201.

[GB14] M. Grant and S. Boyd. *CVX: Matlab software for disciplined convex programming, version 2.1*. 2014.

[GCG19] M. Garstka, M. Cannon, and P. Goulart. "COSMO: A conic operator splitting method for large convex problems". In: *European Control Conference*. Naples, Italy, 2019. DOI: 10.23919/ECC.2019.8796161. arXiv: 1901.10887 [math.OC].

[Gom+17] C. Gomes, B. Legat, R. M. Jungers, and H. Vangheluwe. "Stable Adaptive Co-Simulation : A Switched Systems Approach". In: *IUTAM Symposium on Co-Simulation and Solver Coupling*. Darmstadt, Germany, 2017, to appear.

[Gom+18a] C. Gomes, R. M. Jungers, B. Legat, and H. Vangheluwe. "Minimally Constrained Stable Switched Systems and Application to Co-simulation". In: *57th IEEE Conference on Decision and Control*. IEEE. 2018.

[Gom+18b] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe. "Co-simulation: a survey". In: *ACM Computing Surveys (CSUR)* 51.3 (2018), pp. 1–33.

[Gom+19] C. Gomes, B. Legat, R. M. Jungers, and H. Vangheluwe. "Stable Adaptive Co-simulation: A Switched Systems Approach". In: *IUTAM Symposium on Co-Simulation and Solver Coupling*. Springer. 2019, pp. 81–97.

[GP13] N. Guglielmi and V. Protasov. "Exact computation of joint spectral characteristics of linear operators". In: *Foundations of Computational Mathematics* 13.1 (2013), pp. 37–97.

[Gri96] G. Gripenberg. "Computing the joint spectral radius". In: *Linear Algebra and its Applications* 234 (1996), pp. 43–60.

[GV12] G. H. Golub and C. F. Van Loan. *Matrix computations*. Vol. 3. JHU Press, 2012.

[GW95] M. X. Goemans and D. P. Williamson. "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming". In: *Journal of the ACM (JACM)* 42.6 (1995), pp. 1115–1145.

[GZ08] N. Guglielmi and M. Zennaro. "An algorithm for finding extremal polytope norms of matrix families". In: *Linear Algebra and its Applications* 428.10 (2008), pp. 2265–2282.

[Har+11] K. G. Hare, I. D. Morris, N. Sidorov, and J. Theys. "An explicit counterexample to the Lagarias-Wang finiteness conjecture". In: *Advances in Mathematics* 226.6 (2011), pp. 4667–4701. ISSN: 0001-8708. DOI: http://dx.doi.org/10.1016/j.aim.2010.12.012.

[Har+17] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola. *Pyomo-optimization modeling in python*. Vol. 67. Springer, 2017.

[HK14] D. Henrion and M. Korda. "Convex computation of the region of attraction of polynomial control systems". In: *IEEE Transactions on Automatic Control* 59.2 (2014), pp. 297–312.

[HL05] D. Henrion and J.-B. Lasserre. "Detecting global optimality and extracting solutions in GloptiPoly". In: *Positive polynomials in control*. Springer, 2005, pp. 293–310.

[HL12] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2012.

[HLP52] G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. Cambridge university press, 1952.

[HN10] J. W. Helton and J. Nie. "Semidefinite representation of convex sets". In: *Mathematical Programming* 122.1 (2010), pp. 21–64.

[IBM76] IBM World Trade Corporation. *IBM Mathematical Programming System Extended/370 (MPS/370) Program Reference Manual*. Tech. rep. SH19-1095-1. New York: IBM, 1976.

[IN89] G. F. Italiano and U. Nanni. *Online maintenance of minimal directed hypergraphs*. Tech. rep. CUCS-435-89. Department of Computer Science, Columbia University Series, 1989.

[Ing71] A. Ingleton. "Conditions for representability and transversality of matroids". In: *Théorie des Matroïdes*. Springer, 1971, pp. 62–66.

[JCG14] R. M. Jungers, A. Cicone, and N. Guglielmi. "Lifted polytope methods for computing the joint spectral radius". In: *SIAM Journal on Matrix Analysis and Applications* 35.2 (2014), pp. 391–410.

[JL+03] A. Jadbabaie, J. Lin, et al. "Coordination of groups of mobile autonomous agents using nearest neighbor rules". In: *IEEE Transactions onAutomatic Control* 48.6 (2003), pp. 988–1001.

[Joh+99] K. H. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry. "On the regularization of Zeno hybrid automata". In: *Systems & control letters* 38.3 (1999), pp. 141–150.

[Joh14] F. John. "Extremum Problems with Inequalities as Subsidiary Conditions". In: *Traces and Emergence of Nonlinear Programming*. Ed. by G. Giorgi and T. H. Kjeldsen. First published in 1948. Springer Basel, 2014, pp. 197–215. ISBN: 978-3-0348-0438-7. DOI: 10.1007/978-3-0348-0439-4_9.

[Jun09] R. Jungers. *The joint spectral radius: theory and applications*. Vol. 385. Springer Science & Business Media, 2009.

[KA20] N. Kochdumper and M. Althoff. "Constrained Polynomial Zonotopes". In: *arXiv preprint arXiv:2005.08849* (2020).

[Kar78] R. M. Karp. "A characterization of the minimum cycle mean in a digraph". In: *Discrete Mathematics* 23.3 (1978), pp. 309–311. ISSN: 0012-365X. DOI: http://dx.doi.org/10.1016/0012-365X(78)90011-0.

[KHJ13] M. Korda, D. Henrion, and C. N. Jones. "Inner approximations of the region of attraction for polynomial dynamical systems". In: *IFAC Proceedings Volumes* 46.23 (2013), pp. 534–539.

[KHJ14] M. Korda, D. Henrion, and C. N. Jones. "Convex computation of the maximum controlled invariant set for polynomial control systems". In: *SIAM Journal on Control and Optimization* 52.5 (2014), pp. 2944–2969.

[KL51] S. Kullback and R. A. Leibler. "On information and sufficiency". In: *The annals of mathematical statistics* (1951), pp. 79–86.

[Koz14] V. Kozyakin. "The Berger–Wang formula for the Markovian joint spectral radius". In: *Linear Algebra and its Applications* 448 (2014), pp. 315–328.

[KV05] A. B. Kurzhanski and P. Varaiya. "On verification of controlled hybrid dynamics through ellipsoidal techniques". In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*. IEEE. 2005, pp. 4682–4687.

[Las09] J. B. Lasserre. *Moments, positive polynomials and their applications*. World Scientific, 2009.

[Lau09] M. Laurent. "Sums of squares, moment matrices and optimization over polynomials". In: *Emerging applications of algebraic geometry*. Springer, 2009, pp. 157–270.

[Leg+20] B. Legat, O. Dowson, J. D. Garcia, and M. Lubin. "MathOptInterface: a data structure for mathematical optimization problems". In: *arXiv preprint arXiv:2002.03447* (2020).

[Li03] T. Li. "Solving polynomial systems by homotopy continuation methodsl". In: *COMPUTER MATHEMATICS* (2003), p. 18.

[Lia07] A. Liapounoff. "Problème général de la stabilité du mouvement". In: *Annales de la Faculté des sciences de Toulouse : Mathématiques* 9 (1907), pp. 203–474.

[Lib12] D. Liberzon. *Switching in systems and control*. Springer Science & Business Media, 2012.

[LJP16] B. Legat, R. M. Jungers, and P. A. Parrilo. "Generating unstable trajectories for Switched Systems via Dual Sum-Of-Squares techniques". In: *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. HSCC '16. Vienna, Austria: ACM, 2016, pp. 51–60. ISBN: 978-1-4503-3955-1. DOI: 10.1145/2883817.2883821.

[LM95] D. Lind and B. Marcus. *An introduction to symbolic dynamics and coding*. Cambridge university press, 1995.

[Lof04] J. Lofberg. "YALMIP: A toolbox for modeling and optimization in MATLAB". In: *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE. 2004, pp. 284–289.

[LPJ17] B. Legat, P. A. Parrilo, and R. M. Jungers. "Certifying unstability of Switched Systems using Sum of Squares Programming". In: *ArXiv e-prints* (Oct. 2017). arXiv: 1710.01814 [math.OC].

[LPJ19a] B. Legat, P. A. Parrilo, and R. M. Jungers. *An entropy-based bound for the computational complexity of a switched system*. https://www.codeocean.com/. Version v1. June 2019. DOI: https://doi.org/10.24433/CO.0452244.v1.

[LPJ19b] B. Legat, P. A. Parrilo, and R. M. Jungers. "An entropy-based bound for the computational complexity of a switched system". In: *IEEE Transactions on Automatic Control* (2019). DOI: 10.1109/TAC.2019.2902625.

[LPJ19c] B. Legat, P. A. Parrilo, and R. M. Jungers. *Certifying unstability of Switched Systems using Sum of Squares Programming*. https://www.codeocean.com/. Version v1. Oct. 2019. DOI: 10.24433/CO.9148109.v1.

[LPJ20a] B. Legat, P. A. Parrilo, and R. M. Jungers. *Certifying unstability of Switched Systems using Sum of Squares Programming*. https://www.codeocean.com/. Version v2. May 2020. DOI: 10.24433/CO.9148109.v2.

[LPJ20b] B. Legat, P. A. Parrilo, and R. M. Jungers. "Certifying unstability of switched systems using Sum of Squares Programming". In: *SIAM Journal on Control and Optimization* (2020).

[LRJ20a] B. Legat, S. V. Raković, and R. M. Jungers. *Piecewise semi-ellipsoidal control invariant sets*. https://doi.org/10.24433/CO.6396918.v1. Version v1. May 2020. DOI: 10.24433/CO.6396918.v1.

[LRJ20b] B. Legat, S. Raković, and R. M. Jungers. "Piecewise semi-ellipsoidal control invariant sets". In: *IEEE Control Systems Letters* (2020).

[LT12] D. Liberzon and S. Trenn. "Switched nonlinear differential algebraic equations: Solution theory, Lyapunov functions, and stability". In: *Automatica* 48.5 (2012), pp. 954–963.

[LTJ18] B. Legat, P. Tabuada, and R. M. Jungers. "Computing controlled invariant sets for hybrid systems with applications to model-predictive control". In: vol. 51. 16. 6th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2018. 2018, pp. 193–198. DOI: https://doi.org/10.1016/j.ifacol.2018.08.033.

[LTJ20] B. Legat, P. Tabuada, and R. M. Jungers. "Sum-of-Squares methods for controlled invariant sets with applications to model-predictive control". In: *Nonlinear Analysis: Hybrid Systems* 36 (2020), p. 100858.

[LW95] J. C. Lagarias and Y. Wang. "The finiteness conjecture for the generalized spectral radius of a set of matrices". In: *Linear Algebra and its Applications* 214.0 (1995), pp. 17–42. ISSN: 0024-3795. DOI: http://dx.doi.org/10.1016/0024-3795(93)00052-2.

[Mat07] F. Matúš. "Two constructions on limits of entropy functions". In: *Information Theory, IEEE Transactions on* 53.1 (2007), pp. 320–330.

[May14] D. Q. Mayne. "Model predictive control: Recent developments and future promise". In: *Automatica* 50.12 (2014), pp. 2967–2986.

[MC13] F. Matúš and L. Csirmaz. "Entropy region and convolution". In: *ArXiv e-prints* (Oct. 2013). arXiv: 1310.5957 [cs.IT].

[MD95] D. Manocha and J. Demmel. "Algorithms for intersecting parametric and algebraic curves II: multiple intersections". In: *Graphical Models and Image Processing* 57.2 (1995), pp. 81–100.

[MK87] K. G. Murty and S. N. Kabadi. "Some NP-complete problems in quadratic and nonlinear programming". In: *Mathematical Programming: Series A and B* 39.2 (1987), pp. 117–129.

[MLB05] A. Magnani, S. Lall, and S. Boyd. "Tractable fitting with convex polynomials via sum-of-squares". In: *Proceedings of the 44th IEEE Conference on Decision and Control, and European Control Conference 2005*. IEEE. 2005, pp. 1672–1677.

[MMT10] M. Madiman, A. Marcus, and P. Tetali. "Information-theoretic inequalities in additive combinatorics". In: *Proc. of the 2010 IEEE Information Theory Workshop*. 2010, pp. 1–4.

[MP07] J. Martí-Farré and C. Padró. "On secret sharing schemes, matroids and polymatroids". In: *Theory of Cryptography*. Springer, 2007, pp. 273–290.

[MS95] F. Matúš and M. Studený. "Conditional independences among four random variables I". In: *Combinatorics, Probability and Computing* 4.03 (1995), pp. 269–278.

[MT00] R. Monteiro and M. Todd. "Path-following methods". In: *Handbook of semidefinite programming*. Springer, 2000, pp. 267–306.

[MZ99] R. D. C. Monteiro and P. Zanjácomo. "Implementation of primal-dual methods for semidefinite programming based on Monteiro and Tsuchiya Newton directions and their variants". In: *Optimization Methods and Software* 11.1-4 (1999), pp. 91–140. DOI: 10.1080/10556789908805749. eprint: https://doi.org/10.1080/10556789908805749.

[Nes00] Y. Nesterov. "Squared functional systems and optimization problems". In: *High performance optimization*. Springer, 2000, pp. 405–440.

[Nes04] Y. Nesterov. *Introductory lectures on convex optimization*. Vol. 87. Springer Science & Business Media, 2004.

[OD85] D. H. Owens and D. L. Debeljkovic. "Consistency and Liapunov stability of linear descriptor systems: A geometric analysis". In: *IMA Journal of Mathematical Control and Information* 2.2 (1985), pp. 139–151.

[ODo+16] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. "Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding". In: *Journal of Optimization Theory and Applications* 169.3 (2016), pp. 1042–1068.

[OPJ16] M. Ogura, V. M. Preciado, and R. M. Jungers. "Efficient method for computing lower bounds on the p-radius of switched linear systems". In: *Systems & Control Letters* 94 (2016), pp. 159–164.

[Orc84] W. Orchard-Hays. "History of Mathematical Programming Systems". In: *Annals of the History of Computing* 6.3 (1984), pp. 296–312.

[Oxl06] J. G. Oxley. *Matroid theory*. Vol. 3. Oxford University Press, USA, 2006.

[Pap18] A. Papavasiliou. *Stochastic Dual Dynamic Programming*. 2018.

[Par00] P. A. Parrilo. "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization". PhD thesis. Citeseer, 2000.

[Pet85] I. Petersen. "Quadratic stabilizability of uncertain linear systems: existence of a nonlinear stabilizing control does not imply existence of a linear stabilizing control". In: *IEEE Transactions on Automatic Control* 30.3 (1985), pp. 291–293.

[PF13] M. Pavon and A. Ferrante. "On the Geometry of Maximum Entropy Problems". In: *SIAM Review* 55.3 (2013), pp. 415–439. DOI: 10.1137/120862843. eprint: https://doi.org/10.1137/120862843.

[Phi+15] M. Philippe, R. Essick, G. Dullerud, and R. M. Jungers. "Stability of discrete-time switching systems with constrained switching sequences". In: *arXiv preprint arXiv:1503.06984* (2015).

[Phi+16] M. Philippe, R. Essick, G. E. Dullerud, and R. M. Jungers. "Stability of discrete-time switching systems with constrained switching sequences". In: *Automatica* 72 (2016), pp. 242–250.

[Pip03] N. Pippenger. "The inequalities of quantum information theory". In: *Information Theory, IEEE Transactions on* 49.4 (2003), pp. 773–789.

[PJ08] P. A. Parrilo and A. Jadbabaie. "Approximation of the joint spectral radius using sum of squares". In: *Linear Algebra and its Applications* 428.10 (2008), pp. 2385–2402.

[PL03] P. A. Parrilo and S. Lall. "Semidefinite programming relaxations and algebraic optimization in control". In: *European Journal of Control* 9.2-3 (2003), pp. 307–321.

[Pol20] A. Polyakov. *Generalized Homogeneity in Systems and Control.* 2020.

[Pro16] V. Y. Protasov. "Spectral simplex method". In: *Mathematical Programming* 156.1-2 (2016), pp. 485–511.

[Pro97] V. Y. Protasov. "The generalized joint spectral radius. A geometric approach". In: *Izvestiya: Mathematics* 61.5 (1997), p. 995.

[PT07] I. Pólik and T. Terlaky. "A survey of the S-lemma". In: *SIAM review* 49.3 (2007), pp. 371–418.

[Put93] M. Putinar. "Positive polynomials on compact semi-algebraic sets". In: *Indiana University Mathematics Journal* 42.3 (1993), pp. 969–984.

[QB03] S. J. Qin and T. A. Badgwell. "A survey of industrial model predictive control technology". In: *Control engineering practice* 11.7 (2003), pp. 733–764.

[Rak17] S. V. Raković. "The Minkowski–Lyapunov equation". In: *Automatica* 75 (2017), pp. 32–36.

[Rak20] S. V. Raković. "Robust Control Minkowski–Lyapunov Functions". In: *Automatica* (2020). In review.

[RB10] S. V. Rakovic and M. Baric. "Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks". In: *IEEE Transactions on Automatic Control* 55.7 (2010), pp. 1599–1614.

[Rez78] B. Reznick. "Extremal PSD forms with few terms". In: *Duke Math. J.* 45.2 (June 1978), pp. 363–374. DOI: 10.1215/S0012-7094-78-04519-2.

[RMT13] M. Rungger, M. Mazo Jr, and P. Tabuada. "Specification-guided controller synthesis for linear systems and safe linear-time temporal logic". In: *Proceedings of the 16th international conference on Hybrid systems: computation and control.* ACM. 2013, pp. 333–342.

[Roc15] R. T. Rockafellar. *Convex analysis.* Princeton university press, 2015.

[RS60] G.-C. Rota and W. Strang. "A note on the joint spectral radius". In: *Proceedings of the Netherlands Academy* (1960). 22:379–381.

[RT17] M. Rungger and P. Tabuada. "Computing robust controlled invariant sets of linear systems". In: *IEEE Transactions on Automatic Control* (2017).

[RW98] R. T. Rockafellar and R. J. Wets. "Variational Analysis". In: 317 (1998). ISSN: 0072-7830. DOI: 10.1007/978-3-642-02431-3.

[Sch13] R. Schneider. *Convex bodies: the Brunn–Minkowski theory*. 151. Cambridge University Press, 2013.

[Sch91] K. Schmüdgen. "Thek-moment problem for compact semi-algebraic sets". In: *Mathematische Annalen* 289.1 (1991), pp. 203–206.

[SGV18] M. Souto, J. D. Garcia, and Á. Veiga. "Exploiting Low-Rank Structure in Semidefinite Programming by Approximate Operator Splitting". In: *arXiv preprint arXiv:1810.05231* (2018).

[Sha48] C. E. Shannon. "A mathematical theory of communication". In: *Bell System Technical Journal* 27 (1948), 379–423 and 623–656.

[Sho87] N. Shor. "Class of global minimum bounds of polynomial functions". In: *Cybernetics and Systems Analysis* 23.6 (1987), pp. 731–734.

[Sip13] M. Sipser. *Introduction to the Theory of Computation*. 2013. ISBN: 978-1-133-18781-3.

[SN05] A. Shapiro and A. Nemirovski. "On complexity of stochastic programming problems". In: *Continuous optimization*. Springer, 2005, pp. 111–146.

[SNO16] S. W. Smith, P. Nilsson, and N. Ozay. "Interdependence quantification for compositional control synthesis with an application in vehicle safety systems". In: *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE. 2016, pp. 5700–5707.

[Son83] E. D. Sontag. "A Lyapunov-like characterization of asymptotic controllability". In: *SIAM Journal on Control and Optimization* 21.3 (1983), pp. 462–471.

[Stu99] J. F. Sturm. "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones". In: *Optimization methods and software* 11.1-4 (1999), pp. 625–653.

[Tab09] P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.

[TO95] O. Toker and H. Ozbay. "On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback". In: *American Control Conference, Proceedings of the 1995*. Vol. 4. IEEE. 1995, pp. 2525–2526.

[TPS98] C. Tomlin, G. J. Pappas, and S. Sastry. "Conflict resolution for air traffic management: A study in multiagent hybrid systems". In: *IEEE Transactions on automatic control* 43.4 (1998), pp. 509–521.

[TTT03] R. H. Tütüncü, K.-C. Toh, and M. J. Todd. "Solving semidefinite-quadratic-linear programs using SDPT3". In: *Mathematical programming* 95.2 (2003), pp. 189–217.

[Tut58a] W. T. Tutte. "A homotopy theorem for matroids, I". In: *Transactions of the American Mathematical Society* 88.1 (1958), pp. 144–160.

[Tut58b] W. Tutte. "A homotopy theorem for matroids, II". In: *Transactions of the American Mathematical Society* 88.1 (1958), pp. 161–174.

[Ude+14] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd. "Convex Optimization in Julia". In: *SC14 Workshop on High Performance Technical Computing in Dynamic Languages* (2014). arXiv: 1410.4821 [math-oc].

[Vam68] P. Vamos. "On the representation of independence structures". In: *Unpublished manuscript* (1968).

[Wan+14] Y. Wang, N. Roohi, G. E. Dullerud, and M. Viswanathan. "Stability of linear autonomous systems under regular switching sequences". In: *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE. 2014, pp. 5445–5450.

[Wel10] D. J. A. Welsh. *Matroid theory*. Courier Corporation, 2010.

[Won85] W. M. Wonham. "Linear multivariable control: A Geometric Approach". In: *Applications of Mathematics*. Third. Vol. 10. Springer, 1985. DOI: 10.1007/978-1-4612-1082-5.

[WP13] J. C. Willems and J. W. Polderman. *Introduction to mathematical systems theory: a behavioral approach*. Vol. 26. Springer Science & Business Media, 2013.

[YFK03] M. Yamashita, K. Fujisawa, and M. Kojima. "Implementation and evaluation of SDPA 6.0 (semidefinite programming algorithm 6.0)". In: *Optimization Methods and Software* 18.4 (2003), pp. 491–505.

[YM10] S. Yu and P. G. Mehta. "The Kullback-Leibler rate pseudo-metric for comparing dynamical systems". In: *IEEE Transactions on Automatic Control* 55.7 (2010), pp. 1585–1598.

[YST15] L. Yang, D. Sun, and K.-C. Toh. "SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints". In: *Mathematical Programming Computation* 7.3 (2015), pp. 331–366.

[Zha+05] L. Zhang, Y. Shi, T. Chen, and B. Huang. "A new method for stabilization of networked control systems with random delays". In: *IEEE Transactions on automatic control* 50.8 (2005), pp. 1177–1181.

[Zhe+20] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn. "Chordal decomposition in operator-splitting methods for sparse semidefinite programs". In: *Mathematical Programming* 180.1 (2020), pp. 489–532.

[Zho02] D.-X. Zhou. "The p-norm joint spectral radius and its applications in wavelet analysis". In: *AMS IP Studies in Advanced Mathematics* 25 (2002), pp. 305–326.

[Zie95] G. M. Ziegler. *Lectures on polytopes*. Vol. 152. Springer Science & Business Media, 1995.

[ZY97] Z. Zhang and R. W. Yeung. "A non-Shannon-type conditional inequality of information quantities". In: *IEEE Transactions on Information Theory* 43.6 (1997), pp. 1982–1986.

[ZY98] Z. Zhang and R. W. Yeung. "On characterization of entropy function via information inequalities". In: *Information Theory, IEEE Transactions on* 44.4 (1998), pp. 1440–1452.