UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

Path-Complete and Neural Lyapunov Functions: Computation and Performance

Virginie Debauche

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

Dissertation committee:

Prof. Raphaël Jungers (UCLouvain, *Advisor*)
Ph.D. Matteo Della Rossa (Uniud, *Reader*)
Prof. Jean-Charles Delvenne (UCLouvain, *Reader*)
Prof. Alexandre Mauroy (UNamur, *Reader*)
Prof. Aneel Tanwani (LAAS/CNRS, *Reader*)
Prof. Pierre-Antoine Absil (UCLouvain, *Chair*)

Version of August 2024.

Whoever suffers alone suffers the most,
and loses his carefree nature and happy memories.
But when grief is shared with friends and companions,
the mind can rise above suffering.

Shakespeare, King Lear, Act 3 Scene 6

Acknowledgments

A thesis is never the achievement of one single person, but rather the outcome of countless personal and professional exchanges, discussions, and collaborations. This work is no different, and I would like to extend my gratitude to all the people I had the pleasure to meet and work with over the past years.

First of all, I would like to extend my warmest thanks to my supervisor, Raphaël Jungers, for his support over the past four and a half years. I would particularly like to underline his ability to always make time for his researchers, and his willingness to discuss research issues with them. Moreover, I could always count on him for advice and feedback on my work, especially when writing this thesis. His constant reminder to 'put myself in the reader's shoes' has been so ingrained in me that I now instinctively ask my shoes for feedback. Raphaël also attaches great importance to cohesion within his research team and he multiplies opportunities to strengthen it. I am thinking in particular of the group meetings every Tuesday, the many afterworks to celebrate the good news in the team, but also the annual research retreats which I will keep fond memories of. Raphaël also makes sure that we can move forward in good conditions and does everything possible to ensure that we can seize the opportunities that come our way. For instance, he gives everyone the opportunity to attend the most prestigious conferences, and encourages us to take part in internships at renowned universities, while providing us with recommendations and financial support. Finally, I am grateful for the freedom he gives us, while keeping an eye on us and pushing us to go further.

Along with Raphaël, I want to thank the members of my jury: Ph.D. Matteo Della Rossa, Prof. Jean-Charles Delvenne, Prof. Alexandre Mauroy, Prof. Aneel Tanwani and Prof. Pierre-Antoine Absil. Thank you very much for the comments you provided on the draft version of this manuscript, and for the discussions and ideas we shared during the private defense.

My sincere thanks to all the researchers with whom I have collaborated. In particular, I would like to thank Alessandro Abate for his keen interest in my work, his attentiveness, his guidance and his support. Working with you has been a fabulous experience and I hope we will have the opportunity to work together again in the future. Thanks also to Alec Edwards for his invaluable and much-needed help, as well as for his unstoppable expertise on SMT solvers, neural networks and the architecture of FOSSIL.

The Euler building and its occupants provide a wonderful working environment where I could grow professionally over the last few years. First of all, I would like to highlight the fantastic work and support of the administrative and technical staff. I have a special thought for Marie-Christine, Pascale and Astrid, who are always available and eager to help us. I also include François, who always strives to improve our workplace, and Alexandre, whom I have long called upon to resolve my IT problems. I would then like to thank the several doctoral students, post-doctoral fellows and various researchers whose path I have crossed at Euler: Adrien, Anne, Antoine A., Antoine Q., Astrid, Augustin, Bastien, Benoit, Charles, Clémence, Dânâ, Dany, Eléonore, Estelle, Guillaume, John, Julien, Loïc, Lucas, Martina, Matteo, Nizar, Olivier, Philémon, Rémi, Renato, Sébastien C., Sébastien M., Simon M., Simon V., Somya, Thiago, Wei, Yassine and Zheming. Thank you everyone for the daily breaks on the ground floor, the friendly atmosphere, the badminton sessions and the countless afterworks.

I would especially like to acknowledge the unwavering and unfailing support of Antoine, Guillaume, Julien, Matteo and Yassine, who have now become true friends. Working with you and meeting you every day has been a real pleasure. Thank you so much for all the discussions and wonderful memories, which were essential to bringing this thesis to completion.

I would also like to thank my friends and family for their support, love and friendship, which helped me get through the difficult times. I am thinking in particular of Bastien, Justine, Marine and Maxime who listened at length to my complaints and my despair, and never failed to motivate me immediately. I would also like to thank my parents and siblings, Caroline and Xavier, for their constant encouragement. In particular, Dad and Caroline for their careful proof-reading, even though they did not understand much of it. I am also grateful to my partner, FG, for his unwavering patience and his encouragement. His presence has been a steady source of comfort throughout this journey, and for that, I am deeply thankful.

Abstract

Switched systems are essential in modern engineering due to their ability to model complex systems with transitions between different modes of operation. Their stability poses significant challenges because of the interplay between discrete switching and dynamics, requiring advanced mathematical tools for analysis. While Lyapunov theory is widely used to prove stability, classical methods often struggle with the added complexity of switched systems. This has led to research on extending Lyapunov theory to better address these challenges.

The introduction of path-complete Lyapunov functions brought a new perspective by incorporating combinatorial structures to encode the switching signals of the switched system. This thesis extends the study of path-complete Lyapunov functions by addressing the template-dependent ordering of graphs, i.e., comparing stability certificates while considering specific classes of Lyapunov functions. We introduce template-dependent lifts. These are combinatorial operations on graphs, that characterize the ordering of graphs concerning templates that share a common closure property, such as addition or minimum. This novel approach enhances the understanding of conservatism in stability conditions and guides the selection of graph-template pairs for stability analysis.

Additionally, we explore neural Lyapunov functions as a modern approach to approximating the joint spectral radius (JSR) of linear switched systems. We present a framework that fine-tunes neural networks to approximate the JSR with theoretical and empirical guarantees of effectiveness. We leverage machine learning techniques and the CEGIS approach to provide formal correctness in neural Lyapunov functions, demonstrating promising results against classical methods.

Contents

Introduction

IN recent years, due to the outbreak of computerized devices of all kinds, the complexity of dynamical systems has dramatically increased, which has pushed researchers to improve and refine the modeling of numerous phenomena. One of the natural consequences of this evolution is that one can no longer apply classical methods to analyse these new classes of systems. This calls for the development of new appropriate tools and/or the extension of well-established methods for classical systems.

More specifically, *hybrid* and *switched dynamical systems* appeared and became essential in the engineering landscape. Hybrid systems involve the interaction of flows and discrete jumps, described by both continuous and discrete behaviours. In particular, *switched systems* are hybrid systems for which the discrete behaviour corresponds to a switch, i.e. a modification of the (discrete or continuous) dynamics. The switching signal determines properly these variations by defining at each time step the *mode*, i.e. the active dynamics, of the system. These systems have attracted massive research efforts in Systems and Control, because they provide a relatively simple framework for representing many complex engineering applications [SWM+07, DHvdWH11]. As a few examples, they have been used in bipedal robotics [HGB04], image processing [Jun09, Chapter 5], multi-hop control networks [ADJ+11], viral mutation models [HVCMB11] and communication networks [JHK16].

Stability of dynamical systems has probably been one of the major issues in system theory for ages. In order to tackle this problem, one of the most used solutions involves the Lyapunov theory. In practice, this approach consists in looking for a continuous positive-definite function that decreases along the system trajectories. This constraint results in a set of inequalities, said *Lyapunov inequalities*, that the function has to satisfy. The

Lyapunov approach is widely used in practice among others because for many classes of systems and functions, it boils down to solving a convex optimization problem. Their popularity also stems from the fact that the existence of such a function is widely known as a necessary and sufficient guarantee of the asymptotic stability of a dynamical system.

Recent advances have shown that the Lyapunov approach can be generalized to better capture the hybrid behaviour of switched systems. In order to adapt Lyapunov theory to switched systems, the most intuitive and simplest idea was first to use a single Lyapunov function, so-called a *common Lyapunov function*, that decreases along every dynamics of the system. It has been demonstrated that a switched system is stable if and only if there exists a common Lyapunov function, see [KT04]. In case of linear sub-dynamics, this result can be strengthened: stability is equivalent to the existence of a convex Lyapunov function homogeneous of degree 1 (i.e. a Lyapunov *norm*), see [BM99, Jun09]. Despite these appealing converse results, the nature of these "theoretical" Lyapunov functions/norms and, in particular, the complexity of approximating them are often prohibitive [BT97], even in the linear case. For that reason, several alternative approaches have been proposed, most of them [Bra98, Lib03, GHT06] relying on the concept of *multiple Lyapunov functions*. Instead of looking for a single Lyapunov function, this method consists in searching for a *set* of Lyapunov pieces whose joint decreasing behaviour implies stability.

It became even clearer in 2014 that multiple Lyapunov functions could be the key for the stability analysis of switched systems when the *path-complete Lyapunov functions* were introduced. This theory generalizes the one of multiple Lyapunov functions since path-complete Lyapunov functions can be seen as multiple Lyapunov functions whose decrease properties are regulated by an automaton that abstracts the discrete behaviour (the switches) of the system. More precisely, a path-complete Lyapunov function is made of two components, namely a combinatorial and an algebraic component. The combinatorial component is an automaton (i.e. a graph) $\mathcal{G} = (S, E)$ that regulates the switching signals of the system, while the second component is a set of Lyapunov pieces, one per node of the graph, whose decreasing properties are regulated by the graph edges. The graph has to be expressive enough to capture the discrete behaviour of the system. Two scenarios have been introduced in the literature: either any sequence of switching modes can be generated by the graph, in which case the graph is called *path-complete*, or the language of the graph is constrained, allowing to study the stability of the corresponding *constrained* *switched systems*. In both cases, the existence of a path-complete Lyapunov function is a sufficient condition for stability.

The introduction of the path-complete Lyapunov function theory brings about important open questions. Most of them derive from the fact that the stability of a switched dynamical system can be proved with different graphs and with functions belonging to different *templates*, i.e. sets of functions in which the Lyapunov functions are sought. This flexibility is illustrated in Figure 1. Indeed, this formalism allows to activate two levers to add complexity and therefore reduce the conservatism of a class of stability certificates: either by improving the graph or considering more complex templates. However, this usually goes with an increase of the computation time. On the contrary, enlarging the graph sometimes allows to use simpler templates. The aim is then to leverage the flexibility of this framework to identify which graphs and templates to use. In recent years, an increasing attention has been devoted to the comparison of path-complete graphs. In this context, a graph $\mathcal{G}_2$ is said better than another graph $\mathcal{G}_1$ when $\mathcal{G}_2$ is less conservative than $\mathcal{G}_1$, regardless of the template. Despite a decade of research on the question, the problem of comparing two path-complete Lyapunov functions has remained open. This question has eventually been partially answered in [PJ19] where the authors show that the general graph ordering relation can be characterized by intrinsic combinatorial properties of both graphs, such as the *simulation*.

In this thesis, we tackle the same question for finer ordering relations, i.e. when we restrict the solution to belong to a specific template (e.g., quadratic functions) or classes of templates (e.g., sum-closed templates). In particular, we address the characterization of *template-dependent* ordering of graphs for the family of templates which all share a common *closure property*. As we show in the thesis, it turns out that the template-dependent ordering of graphs with respect to such templates can be completely characterized by means of combinatorial operations on the graphs. These operations, called *template-dependent lifts*, leverage the inequalities encoded by a graph with respect to an operation, such as the addition, the minimum and the maximum. These lifts, with the simulation, turn out to be the key tools to characterize the template-dependent ordering of graphs. This complete characterization provides a better and completely novel understanding on the conservatism level of different multiple stability certificates, by taking into account the properties of the template we are using. This is an important step forward in guiding our choice of path-complete Lyapunov function structure.

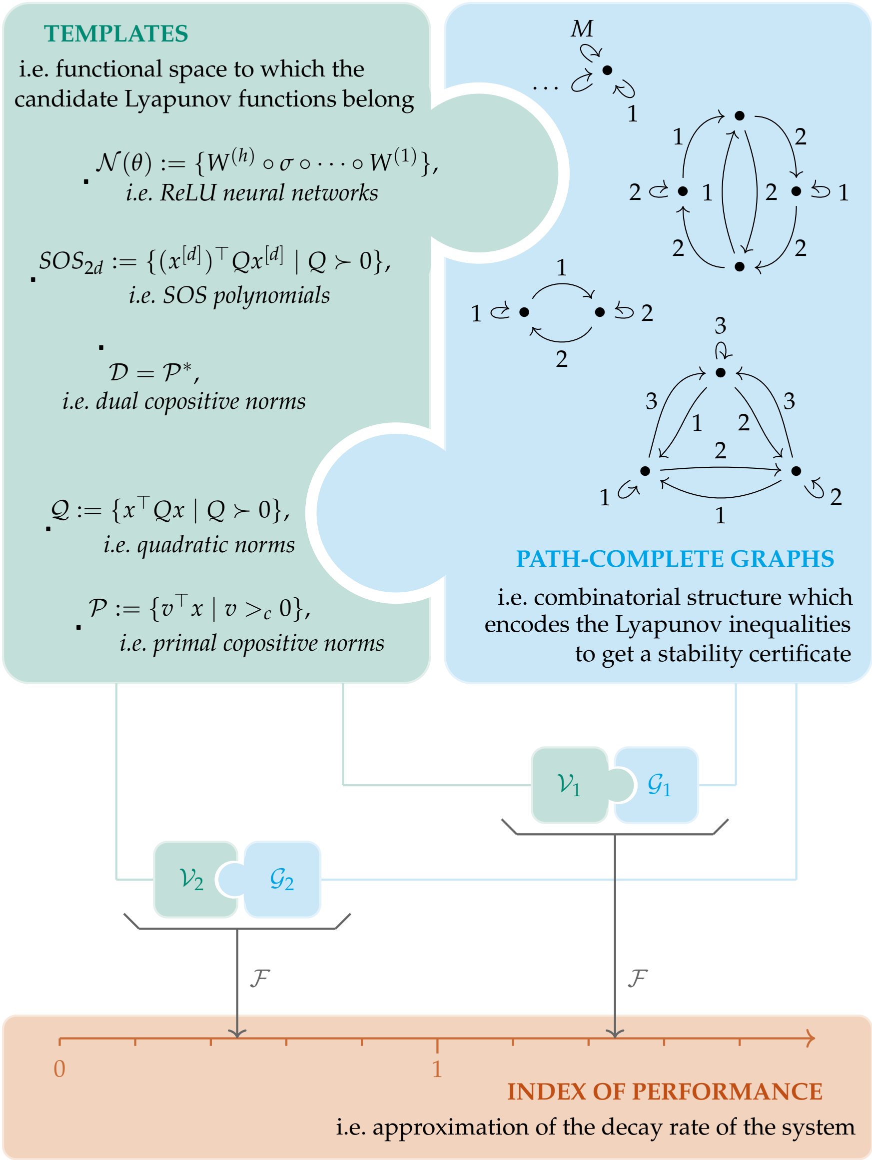


Fig. 1 Illustration of the path-complete Lyapunov formalism, parametrized by two components: the template and the path-complete graph. We measure and compare the conservatism of each pair template-graph by computing an index of performance, defined as the approximation of the decay rate of switched systems.

In particular, we consider *linear switched systems* and the approximation of the *joint spectral radius* as index of performance of a pair graph-template. This quantity, introduced in [RS60], is the maximum rate of growth of a linear switched system, and has been proved to be really hard to compute in practice, see [Koz90, BT97, BC08]. Despite these numerical obstacles, several approximation techniques have been developed and they often prove effective in practice. In our setting, a better graph is less conservative and therefore provides a finer JSR approximation. This will be used throughout this thesis to illustrate our theoretical results on the template-dependent comparison of path-complete graphs.

In addition, we take advantage of recent advances in theoretical and computational Machine Learning and consider the template of *neural Lyapunov functions*. Indeed, in recent years, neural networks have received an increasing amount of attention [Pro94, Ser05, CRG19, AAE+21, DQGF21, FLYL22, ZXQF23] to compute Lyapunov functions. However, several questions remain open such as the soundness of the training procedure, or the capabilities to provide good Lyapunov functions as a function of the network structure (e.g., width, depth, activation function). In this thesis, we fine-tune the loss function to provide an approximation of the JSR of the corresponding set of matrices. Moreover, we are able to link the network's approximation capabilities to its structure. In this setting, we provide both theoretical and empirical evidence for the effectiveness of the neural approximation of the JSR. Among others, we consider different approaches to train neural Lyapunov functions for switched systems that come with formal guarantees of correctness. In particular, we leverage the CEGIS approach [AAE+21, EPA24] that has already proved useful in various fields of system verification and control.

We review the outline of the thesis below, and briefly summarize each section.

⋆ Outline

Part I: Background

We review classical notions which will prove useful in the rest of this thesis.

Chapter 1: Preliminaries

In this section, we introduce and provide a brief summary on several topics which will be used throughout this thesis. Namely, we introduce the reader to *switched systems* which will be the heart of our study, and we provide different key notions to tackle their stability analysis, such as Lyapunov theory and convex theory.

Chapter 2: Joint spectral radius

To illustrate and experimentally validate our theoretical results throughout this thesis, we use the well-known, but still difficult, example of *linear switched systems*. In particular, the stability of these systems is characterized by the *joint spectral radius* whose computation is NP-hard. This chapter introduces this notion and summarizes classical approximation methods which we will later compare with the methods that we introduce in this thesis.

Part II: Neural Lyapunov functions

Motivated by the approximation capabilities and the numerical efficiency of neural networks, we use them to synthesize common Lyapunov functions. In particular, we train them to provide the best data-driven JSR approximation.

Chapter 3: Approximation of the JSR using polytopic norms

As an introduction, we study the template of polytopic norms. We review the classical properties of these objects, such as their complexity index, and motivate their use to approximate the joint spectral radius. We summarize the classical methods for calculating polytopic Lyapunov functions and show that this amounts to solving a bilinear program. In addition, we derive theoretical guarantees on the polytopic approximation of the JSR, as a function of the polytope complexity.

Chapter 4: Neural networks and their representation power

At first, we investigate the *representation power* of neural networks, that is the class of functions that they can represent. In particular, we relate their representation capabilities to their structural components, namely their width, depth, and the choice of the activation function. We focus especially on polynomial and Rectified Linear Units activation functions.

Chapter 5: Approximation of the JSR using ReLU neural networks

In this chapter, we use neural Lyapunov functions to learn a common Lyapunov function which provides the best JSR approximation. We merge the approximation guarantees on polytopic approximations with the representation power of ReLU neural networks to yield bounds on the structure of such a network to achieve a given precision on the JSR approximation. Then, we identify, based on numerical examples, some limitations of the neural approach, and propose various methods to overcome them. Moreover, a variety of numerical examples is provided in this chapter. Part of the results of this chapter have been published in [DEJA24], and the rest comes from unpublished work in collaboration with Alec Edwards and Alessandro Abate.

Part III: Template-dependent comparison of Path-Complete Lyapunov functions

Restrained by the computational complexity of finding common Lyapunov functions, *multiple Lyapunov functions* have emerged as a promising tool for balancing the complexity of inequalities and the template of Lyapunov functions. Due to the wide range of possibilities, we compare multiple Lyapunov stability criteria with respect to their conservatism within specific classes of templates. This is made possible by the *path-complete Lyapunov* formalism which encodes multiple Lyapunov inequalities with directed and labeled graphs. This part is partly based on a collaboration with Matteo Della Rossa.

Chapter 6: Path-complete Lyapunov functions and their comparison

This chapter introduces path-complete Lyapunov functions and formally defines the comparison of path-complete graphs with respect to their conservatism. In particular, we recall the characterization of the general ordering of graphs by the *simulation*, which relates the nodes of the graphs while preserving the edges. Moreover, we show its limitations.

Chapter 7: Template-dependent lifts and closure properties

We focus on the template-dependent ordering of graphs, i.e. when we restrict the comparison to specific templates. To this aim, we introduce combinatorial operations on path-complete graphs which leverage the closure properties of templates, such as the addition, the pointwise minimum and maximum and the composition. The simulation by these so-called template-dependent lifts turns out to be a sufficient condition for the comparison of graphs with respect to the class of templates which share a common closure property. This chapter gathers the results that have been published in [DJ20, DDJ21, DDJ22a].

Chapter 8: Characterization of the template-dependent ordering of graphs with lifts

We finally demonstrate that the simulation relations which involve the lifted graphs are a necessary and sufficient condition for the graph ordering with respect to the family of templates closed under the corresponding closure operation. However, all these theorems require ad-hoc auxiliary results. Therefore, we provide this characterization for the addition, and the pointwise minimum and maximum. We published the research for these results in [DDJ22b, DDJ23]. Moreover, we take a step back to provide a comparison of the different proofs and propose an harmonized proof method.

List of publications

HERE is the list of abstract, conference and journal papers written during this PhD thesis, including those whose content might not be included in this manuscript. It also includes repeatability packages that have accompanied the submission of two conference papers.

Abstract and conference proceedings:

[DJ20] Virginie Debauche and Raphaël M. Jungers. *On path-complete Lyapunov functions: comparison between a graph and its expansion*. In Proceedings of the 39th Benelux Meeting on Systems and Control, 2020.

[DDJ21] Virginie Debauche, Matteo Della Rossa, and Raphaël M. Jungers. *Template-dependent lifts for path-complete stability criteria and application to positive switching systems*. In Proceedings of the 7th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS). IFAC-PapersOnLine, 2021.

[DJ21] Virginie Debauche and Raphaël M. Jungers. *Comparison of Path-Complete Stability Criteria via Quantifier Elimination*. In Proceedings of the 40th Benelux Meeting on Systems and Control, 2021.

[DDJ22b] Virginie Debauche, Matteo Della Rossa, and Raphaël M. Jungers. *Necessary and sufficient conditions for template-dependent ordering of path-complete Lyapunov methods*. In Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control (HSCC). Association for Computing Machinery, 2022.

[DDJ23] Virginie Debauche, Matteo Della Rossa, and Raphaël M. Jungers. *Characterization of the ordering of path-complete stability certificates with*

addition-closed templates. In Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control (HSCC). Association for Computing Machinery, 2023.

[DESMA23] Virginie Debauche, Alec Edwards, Stella Simic, Raphaël M. Jungers and Alessandro Abate. *Formal Synthesis of Path-Complete Lyapunov Functions on Neural Templates*, In Proceedings of the 42th Benelux meeting on systems and control, 2023.

[DEJA24] Virginie Debauche, Alec Edwards, Raphaël M. Jungers, and Alessandro Abate. *Stability analysis of switched linear systems with neural Lyapunov functions*. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI '24. AAAI Press, 2024.

Journal paper:

[DDJ22a] Virginie Debauche, Matteo Della Rossa, and Raphaël M. Jungers. *Comparison of path-complete Lyapunov functions via template-dependent lifts*. Nonlinear Analysis: Hybrid Systems, 46, 2022.

Repeatability packages:

[Deb23] Virginie Debauche. *Lift and simulation of path-complete graphs*, 2022. Accessible on `https://codeocean.com/capsule/6872065/tree/v2`.

[Deb22] Virginie Debauche. *Characterization of ordering of path-complete graphs for addition-closed templates*, 2023. Accessible on `https://codeocean.com/capsule/0480330/tree/v1`.

List of symbols

The main symbols used in this thesis are listed here below.

Basics

$\mathbb{R}, \mathbb{Z}, \mathbb{N}$ — Real, natural and positive integer numbers (p. 18)

$\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$ — Positive and nonnegative real numbers (p. 20)

$\langle M \rangle$ — Set $\{1, \ldots, M\}$ where $M \in \mathbb{N}$ (p. 18)

$\langle M \rangle^k$ — Words of length k on the alphabet $\langle M \rangle$ (p. 19)

$\langle M \rangle^+$ — Kleene closure of the alphabet $\langle M \rangle$ (p. 19)

X^I — Set of indexed sets $\{X_i \in X : i \in I\}$ with values in X (p. 18)

$\lceil x \rceil$ — Ceiling of x (p. 76)

Vectors and matrices

$\mathbb{R}^n$ — Vectors of dimension n (p. 18)

$\mathbb{R}^{n \times n}$ — $n \times n$ matrices (p. 18)

$Q \succ 0$, $Q \succeq 0$ — Positive definitive and semi-definite matrix (p. 39)

$\mathcal{A}$ — Set of M matrices, i.e. $\mathcal{A} := \{A_1, \ldots, A_M\}$ (p. 34)

$\mathcal{A}^k$ — Products of length k in $\mathcal{A}$ (p. 34)

$\mathcal{A}_\gamma$ — Scaled set of matrices by γ, i.e. $\{A_1/\gamma, \ldots, A_M/\gamma\}$ (p. 38)

$\rho(\mathcal{A})$ — Joint spectral radius of $\mathcal{A}$ (p. 34)

$\rho_{\mathcal{Q}}(\mathcal{A})$ —— Quadratic approximation of the JSR (p. 40)

$\rho_{SOS,2d}(\mathcal{A})$ —— SOS approximation of the JSR of degree $2d$ (p. 44)

$\rho_{\mathcal{P}}(\mathcal{A})$ —— Primal linear copositive approximation of the JSR (p. 48)

$\rho_{\mathcal{D}}(\mathcal{A})$ —— Dual linear copositive approximation of the JSR (p. 48)

$\rho_{POL_H,d}(\mathcal{A})$ —— Polytopic approximation of the JSR of H-complexity d (p. 62)

$\rho_{POL_V,d}(\mathcal{A})$ —— Polytopic approximation of the JSR of V-complexity d (p. 62)

Convex analysis

$\mathbb{B}_f$ —— Unit ball of a function f, i.e. $\mathbb{B}_f := \{x \in \mathbb{R}^n \mid f(x) \leq 1\}$ (p. 25)

conv(C) —— Convex hull of a subset C (p. 25)

$\mathcal{V}(X)$ —— Set of norms on the vector space X (p. 23)

$\mathcal{K}(\mathbb{R}^n)$ —— Sets of closed, bounded, convex and symmetric subsets of $\mathbb{R}^n$ with $0 \in \text{int}(C)$ (p. 24)

C° —— Polar of a convex and closed subset $C \subset \mathbb{R}^n$ with $0 \in C$ (p. 26)

$f^\star$ —— The dual norm of a norm $f \in \mathcal{V}(\mathbb{R}^n)$ (p. 27)

Neural networks

$NN(\cdot)$ —— Function represented by a neural network (p. 68)

σ —— Activation function of a neural network (p. 68)

Graphs and path-complete formalism

$\mathcal{G} = (S, E)$ —— Directed and labeled graph, with nodes S and edges E (p. 118)

$\mathcal{G}^\top = (S, E^\top)$ —— Dual graph of $\mathcal{G} = (S, E)$ (p. 119)

$\mathcal{G}^e = (S^e, E^e)$ —— Expanded graph of $\mathcal{G} = (S, E)$ (p. 121)

$\overline{E} := S \times S \times \langle M \rangle \setminus E$ —— Non-existing edges of a graph $\mathcal{G} = (S, E)$ (p. 135)

$\mathcal{G}_0 = (\{s_0\}, \{(s_0, s_0, i)_{i \in \langle M \rangle}\}$ —— Common Lyapunov function graph (p. 119)

$\mathcal{G}_{db}^K$ —— De Bruijn graph of order $K - 1$ (p. 122)

$\mathcal{C}_+^0(\mathbb{R}^n, \mathbb{R})$ —— Set of candidate Lyapunov functions (p. 22)

Abbreviations

PART I
Background

1

Preliminaries

THIS chapter gives an introduction on switched systems and summarizes auxiliary key tools required for their stability analysis. In particular, Section 1.1 deals with the stability of switched systems over a language and presents Lyapunov theory, while Section 1.2 focuses on norms and summarizes convex and duality theory.

1.1 Switched systems, language theory and stability

In this thesis, we study *discrete-time switched dynamical systems*, a particular case of Cyber-Physical systems, which have been intensely studied [Lib03] in past decades. Switched systems are commonly used to model various physical or engineering phenomena [SWM+07, DHvdWH11], in which the state is possibly driven by several dynamic laws. To cite a few examples, they have been used in bipedal robotics [HGB04], in image processing [Jun09, Chapter 5], and multihop control networks [ADJ+11]. Closer to AI, switched systems have been used to model Q-learning algorithms [LH20], and classification techniques have been used in switched system identification [LB08]. In addition, switched systems bring several challenging problems from a theoretic point of view, see [LM99, Jun09].

Let us formally define *discrete-time switched dynamical systems*.

Definition 1.1 (Discrete-time switched system)**.** A *discrete-time switched dynamical system* with M dynamics of dimension $n \in \mathbb{N}$ is a dynamical system of the form

$$x(k+1) \; := \; f_{\sigma(k)}(x(k)), \tag{1.1}$$

where

- $k \in \mathbb{N}$ represents the discrete time;
- $x(k) \in \mathbb{R}^n$ is the *state* of the system at time $k \in \mathbb{N}$;
- the *switching signal* $\sigma : \mathbb{N} \to \langle M \rangle$ determines the *mode* $\sigma(k)$ of the system at each time $k \in \mathbb{N}$;
- $f_{\sigma(k)} \in F := \{f_i : i \in \langle M \rangle\}$ is called the dynamics of the system at time $k \in \mathbb{N}$.

Given a point $x_0 \in \mathbb{R}^n$ and a switching signal $\sigma : \mathbb{N} \to \langle M \rangle$, we denote with $x(k, x_0, \sigma)$ the trajectory starting at x_0, following the dynamics in Equation (1.1) with respect to σ and evaluated at instant $k \in \mathbb{N}$.

Here, the switching signal can be interpreted as an exogenous perturbation: one can think of an operator who may switch the operating mode of a system, or any other situation where the law of dynamics may switch from time to time e.g. due to external disturbance, or change of specification. The most general case is the so-called *arbitrarily switched systems*, which exhibit arbitrary switching sequences among a finite set of modes. If some switching sequences are forbidden, we speak of *constrained switched systems*.

In many instances, we will use a special case of switched systems to illustrate our results, namely *linear switched systems*.

Definition 1.2 (Linear switched system)**.** A *linear switched system* is a switched system $F := \{f_i : i \in \langle M \rangle\}$ for which each dynamics is linear, i.e.

$$\forall i \in \langle M \rangle, \, \forall x \in \mathbb{R}^n : \; f_i(x) := A_i x,$$

where $A_i \in \mathbb{R}^{n \times n}$ for each mode $i \in \langle M \rangle$.

Linear switched systems are used to model engineering applications in several fields, such as viral mutation models [HVCMB11] and communication networks [JHK16].

The subject of this thesis concerns the stability analysis of switched systems under arbitrary switching. In particular, we aim at analyzing and comparing different Lyapunov stability certificates with respect to their conservatism. For the sake of generality, we first need to introduce a few preliminaries about language theory before formally defining different notions of stability for switched systems.

1.1.1 Language theory

We start by viewing the different modes of a switched dynamical system as the symbols of an *alphabet*, and the admissible switching sequences as a *language* on this alphabet. The purpose of this chapter, mainly inspired by [CL10, Chapter 2], is to provide a brief summary of language theory.

An *alphabet* is defined as a finite set of *symbols*. In this manuscript, we will consider integers as symbols, i.e. we consider an alphabet of the form

$$\langle M \rangle := \{1, \dots, M\}.$$

A *word* $w := w(1)w(2)\dots w(k)$ on this alphabet is a finite sequence of symbols from this alphabet. The *length* $|w|$ of a word w is the number of symbols it contains, including the multiple occurrences of the same symbol. In particular, we denote by $\langle M \rangle^k$ the set of words of fixed length $k \in \mathbb{N}$ on the alphabet $\langle M \rangle$, and ϵ denotes the empty word. The reverse of a word $w = w(1)\dots w(k)$ is a word of same length with the symbols in the reverse order, i.e. $w^\top := w(k)\dots w(1)$.

The main operation on words involved in the language theory is the *concatenation*.

Definition 1.3 (Concatenation)**.** Given an alphabet $\langle M \rangle$, the *concatenation* of two words u and v, denoted by uv, is the word consisting of the symbols of u directly followed by the symbols of v.

One can define the smallest set which contains an alphabet and closed under concatenation.

Definition 1.4 (Kleene closure)**.** Given an alphabet $\langle M \rangle$, the *Kleene closure of* $\langle M \rangle$, denoted by $\langle M \rangle^+$, is the infinite set of all possible words of all possible lengths over $\langle M \rangle$, excluding the empty word, i.e.

$$\langle M \rangle^+ := \bigcup_{k \in \mathbb{N}} \langle M \rangle^k.$$

We say that y is a *subword* of ω if there exist $u, v \in \langle M \rangle^* := \langle M \rangle^+ \cup \{\epsilon\}$ such that $w = uyv$.

Then, we can define a *language* on an alphabet $\langle M \rangle$ as a set of finite words on $\langle M \rangle$.

Definition 1.5 (Language)**.** Given an alphabet $\langle M \rangle$, a *language* $\mathcal{L}$ *over* $\langle M \rangle$ is a (finite or infinite) subset of $\langle M \rangle^+$, i.e.

$$\mathcal{L} \subseteq \langle M \rangle^+. \tag{1.2}$$

The language is said *strict* if the inclusion in Equation (1.2) is strict.

In our setting, the different modes $\langle M \rangle$ of a switched system defines an alphabet and any word on this alphabet corresponds to a finite switching sequence. In particular, the language of arbitrarily switched systems corresponds to the Kleene closure, while the language of constrained switched systems is a strict language on $\langle M \rangle$.

1.1.2 Stability and Lyapunov functions

We start by defining a few properties on functions.

Definition 1.6. Consider a function $f : \mathbb{R}^n \to \mathbb{R}$.

- The function f is *homogeneous of degree d* if for all $x \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}_0$, $f(\alpha x) = \alpha^d f(x)$.
- The function f is *positively homogeneous of degree d* if for all $x \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}_{>0}$, $f(\alpha x) = \alpha^d f(x)$.
- The function f is *absolutely homogeneous* if for all $x \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, $f(\alpha x) = |\alpha| f(x)$.
- The function f is *positive semi-definite* if for all $x \in \mathbb{R}^n$, $f(x) \geq 0$ and $f(0) = 0$.
- The function f is *positive definite* if it is positive semi-definite and $f(x) = 0 \Leftrightarrow x = 0$.
- The function f is *radially unbounded* if $f(x) \to \infty$ when $\|x\| \to \infty$.
- The function f is *idempotent* if for all $x \in \mathbb{R}^n$, $f(f(x)) = f(x)$.
- The functions f is *involutory* if for all $x \in \mathbb{R}^n$, $f(f(x)) = x$.
- The function f is *symmetric* if for all $x \in \mathbb{R}^n$, $f(x) = f(-x)$.

We define the stability of switched systems using functions of class $\mathcal{K}$ and $\mathcal{K}_\infty$, defined as follows.

Definition 1.7. A scalar function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is *of class* $\mathcal{K}$ if it is continuous, positive definite and strictly increasing. Moreover, the scalar function α is *of class* $\mathcal{K}_\infty$ if it is of class $\mathcal{K}$ and unbounded.

For the sake of generality, we define the notion of stability of switched systems over a given language. This formalism encompasses both the stability under *arbitrary switching* and *constrained switching sequences*. We make this choice because the path-complete Lyapunov formalism introduced in Chapter 6 allows to deal with both cases.

From this point onwards, $\|\cdot\|$ denotes the euclidean norm.

Definition 1.8 (Stability over a language)**.** Consider a switched system $F := \{f_i : i \in \langle M \rangle\}$ of the form (1.1) and a language $\mathcal{L}$ over $\langle M \rangle$.

1. The system F is *stable over* $\mathcal{L}$ if there exists a function α of class $\mathcal{K}_\infty$ such that
$$\forall w \in \mathcal{L},\ \forall x_0 \in \mathbb{R}^n,\ \forall k \in \mathbb{N} : \ \|x(k, x_0, w)\| \ \leq \ \alpha\left(\|x_0\|\right); \tag{1.3}$$
2. The system F is *asymptotically stable* over $\mathcal{L}$ if it is stable over $\mathcal{L}$ and
$$\forall w \in \mathcal{L},\ \forall x_0 \in \mathbb{R}^n : \ \lim_{k \to \infty} \|x(k, x_0, w)\| = 0; \tag{1.4}$$
3. The system is *exponentially stable* over $\mathcal{L}$ if there exist $\rho < 1$ and $K \geq 1$ such that
$$\forall w \in \mathcal{L},\ \forall x_0 \in \mathbb{R}^n : \ \|x(k, x_0, w)\| \leq K\rho^k \|x_0\|. \tag{1.5}$$

If the language is the Kleene closure, we say that the switched system is *stable over arbitrary switching*.

Remark 1.9. If the dynamics are homogeneous of degree 1, the stability in Equation (1.3) is equivalent to the following expression:
$$\exists K \geq 1,\ \forall w \in \mathcal{L},\ \forall x_0 \in \mathbb{R}^n,\ \forall k \in \mathbb{N} : \ \|x(k, x_0, w)\| \ \leq \ K\|x_0\|. \tag{1.6}$$

Indeed, by homogeneity of the trajectory $x(k, x_0, w)$ with respect to the initial condition x_0,
$$\frac{\|x(k, x_0, w)\|}{\|x_0\|} \ = \ \left\| x\left(k, \frac{x_0}{\|x_0\|}, w\right)\right\| \ \leq \ \alpha(1)$$
and then Equation (1.6) holds with $K := \alpha(1)$. △

One of the possible ways to assess the stability of switched systems is to use the *Lyapunov theory*, and the *common Lyapunov functions* (CLFs) in particular. This approach consists in finding a single positive definite function that decreases along any dynamics of the system.

Let us formally define the *candidate Lyapunov functions*.

Definition 1.10 (Candidate Lyapunov function)**.** A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a *candidate Lyapunov function* if it is continuous, positive definite and radially unbounded. The set of candidate Lyapunov functions on $\mathbb{R}^n$ is denoted by $\mathcal{C}^0_+(\mathbb{R}^n, \mathbb{R})$.

Norms, which will be further studied in Section 1.2, are candidate Lyapunov functions for instance. However, candidate Lyapunov functions do not need to be convex nor subadditive.

The following lemma provides a complete characterization of candidate Lyapunov functions.

Lemma 1.11 (Annex A.3 in [Lib03])**.** *A function $V : \mathbb{R}^n \to \mathbb{R}$ is a candidate Lyapunov function if and only if there exist two functions α, β of class $\mathcal{K}_\infty$ such that*

$$\forall x \in \mathbb{R}^n : \ \alpha(\|x\|) \ \leq \ V(x) \ \leq \ \beta(\|x\|). \tag{1.7}$$

The following theorem demonstrates that the existence of a *common Lyapunov functions* is a necessary and sufficient condition for stability (see for example [Jun09] and [KT04] for the nonlinear case).

Theorem 1.12 ([KT04])**.** *A switched system $F := \{f_i : i \in \langle M \rangle\}$ is stable under arbitrary switching (i.e. over $\langle M \rangle^+$) if and only if there exists a candidate Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$ such that*

$$\forall i \in \langle M \rangle, \ \forall x \in \mathbb{R}^n : \ V(f_i(x)) \ \leq \ V(x). \tag{1.8}$$

Such a function is called a common Lyapunov function *(CLF for short).*

The case of *asymptotic stability* can be studied within the same framework, simply considering strict inequalities on the edges, i.e. in Equation (1.8).

Remark 1.13. Note that the radially unboundedness is implied by the positivity and the homogeneity. Moreover, in the context of homogeneous switched systems such as linear switched systems for instance, there is no conservatism in requiring the candidate Lyapunov functions to be homogeneous, see [Ros92]. In this case, Lemma 1.11 and in particular the expression (1.7) can be replaced by

$$\exists \alpha, \beta \in \mathbb{R} \text{ s.t } 0 < \alpha \leq \beta,\ \forall x \in \mathbb{R}^n : \ \alpha \|x\|^d \ \leq \ V(x) \ \leq \ \alpha \|x\|^d ,$$

where d is the degree of homogeneity of V. △

Although Theorem 1.12 provides a complete characterization of the stability of switched systems, it is largely offset by the computing complexity required by the "search" of this Lyapunov function. In particular, this drawback will be the focus of this thesis.

1.2 Norms, convex set and duality

In this short section, we provide a comprehensive and detailed summary of the convex duality results. These classic statements are useful when studying the particular case of (primal and dual) copositive linear norms as template of candidate Lyapunov functions for positive linear systems. The notation and terminology of this summary are introduced in [Roc70, Part III], in which the readers wishing to learn more about this topic can find the corresponding formal proofs. For notational simplicity we develop the theory on $\mathbb{R}^n$; the corresponding statements for the self-dual cone $\mathbb{R}^n_{\geq 0}$ (as in Section 2.2.3) are straightforwardly obtained, mutatis mutandis.

Norms satisfy the definition of candidate Lyapunov functions by definition.

Definition 1.14 (Norm). Given a vector space X, a *(semi-)norm* is a scalar function $\|\cdot\| : X \to \mathbb{R}$ that is

(1) *absolutely homogeneous of order* 1, i.e.

$$\forall \lambda \in \mathbb{R},\ \forall x \in X : \ \|\lambda x\| = |\lambda| \, \|x\| ,$$

(2) *subbaditive*, i.e.

$$\forall x, y \in X : \ \|x + y\| \ \leq \ \|x\| + \|y\| ,$$

(3) *positive (semi-)definite*.

We denote by $\mathcal{V}(X)$, the set of norms on X. If $X = \mathbb{R}^n$, $\|\cdot\|$ is a *vector norm* while it is called a *matrix norm* if $X = \mathbb{R}^{n \times n}$.

Definition 1.15 (Submultiplicativity)**.** A matrix norm $\|\cdot\| : \mathbb{R}^{n\times n} \to \mathbb{R}$ is *submultiplicative* if for all $A, B \in \mathbb{R}^{n\times n}$,

$$\|AB\| \;\leq\; \|A\|\,\|B\|\,.$$

Definition 1.16 (Matrix norm induced by a vector norm)**.** Given a vector norm $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$, the *induced matrix norm* of a matrix $A \in \mathbb{R}^{n\times n}$ is defined as

$$\|A\| \;:=\; \max_{x\in\mathbb{R}^n} \frac{\|Ax\|}{\|x\|} \;=\; \max_{\|x\|=1} \|Ax\|\,.$$

We consider five operations which preserve the properties of norms, namely the *sum*, the *pointwise maximum*, the *infimal convolution*, the *inverse summation* and the *linear transform*.

Property 1.17. *Given $f_1, f_2 \in \mathcal{V}(\mathbb{R}^n)$, and $A \in \mathbb{R}^{n\times n}$ invertible, we have*

1. (Sum): $f_1 + f_2 \in \mathcal{V}(\mathbb{R}^n)$,
2. (Pointwise maximum): $\max\{f_1, f_2\} \in \mathcal{V}(\mathbb{R}^n)$,
3. (Infimal Convolution): *Defining $f_1 \,\square\, f_2 : \mathbb{R}^n \to \mathbb{R}$ by*

   $$f_1 \,\square\, f_2(x) \;:=\; \inf_{x=x_1+x_2} \{f_1(x_1) + f_2(x_2)\},$$

   if f_1 and $f_2 \in \mathcal{V}(\mathbb{R}^n)$, it holds that $f_1 \,\square\, f_2 \in \mathcal{V}(\mathbb{R}^n)$. (Note that $f_1 \,\square\, f_2 = conv\{\min\{f_1, f_2\}\}$, where $conv(f)$ denotes the largest convex function majorized by f),
4. (Inverse Summation): *Defining $f_1 \sharp f_2 : \mathbb{R}^n \to \mathbb{R}$ by*

   $$f_1 \sharp f_2(x) \;:=\; \inf_{x=x_1+x_2} \{\max\{f_1(x_1), f_2(x_2)\},$$

   we have $f_1 \sharp f_2 \in \mathcal{V}(\mathbb{R}^n)$,
5. (Linear Transform): $f_1 \circ A \in \mathcal{V}(\mathbb{R}^n)$.

Let us now consider *convex sets*, i.e. sets which are closed under convex combination.

Definition 1.18. Given $n \in \mathbb{N}$, $\mathcal{K}(\mathbb{R}^n)$ denotes the family of sets $C \subset \mathbb{R}^n$ such that C is closed, bounded, convex, symmetric ($x \in C$ if and only if $-x \in C$) and $0 \in \text{int}(C)$.

Similarly to Property 1.17, we introduce five operations that preserve the convexity, which are the *sum*, the *intersection*, the *convex hull of the union*, the *inverse sum* and the *linear transform*.

Property 1.19. *Consider* $C_1, C_2 \in \mathcal{K}(\mathbb{R}^n)$, $A \in \mathbb{R}^{n\times n}$ *invertible. Then:*

1. (Sum): $C_1 + C_2 \in \mathcal{K}(\mathbb{R}^n)$,
2. (Intersection): $C_1 \cap C_2 \in \mathcal{K}(\mathbb{R}^n)$,
3. (Convex Hull of Union): $C_1 \,\square\, C_2 := conv\{C_1 \cup C_2\} \in \mathcal{K}(\mathbb{R}^n)$,
4. (Inverse Sum): $C_1 \,\sharp\, C_2 := \bigcup_{\lambda\in[0,1]} \lambda C_1 \cap (1-\lambda)C_2 \ \in \mathcal{K}(\mathbb{R}^n)$,
5. (Linear Transform): $AC_1 \in \mathcal{K}(\mathbb{R}^n)$.

There is a 1-to-1 correspondence between sets in $\mathcal{K}(\mathbb{R}^n)$ and norms on $\mathbb{R}^n$. This correspondence is induced by the unit-sublevel sets of norms and by the *Gauge* or *Minkowski* functions of sets (respectively) in $\mathcal{K}(\mathbb{R}^n)$, as explained here below.

Definition 1.20. Given a set $C \in \mathcal{K}(\mathbb{R}^n)$, we define the *Gauge* or *Minkowski function associated to* C, $g(\,\cdot\,|\,C) : \mathbb{R}^n \to \mathbb{R}$ by

$$g(x|C) \ := \ \inf\{\gamma \in \mathbb{R} \mid x \in \gamma C,\ \gamma \geq 0\}.$$

The following lemma provides the correspondence between norms and convex sets. Note that, given $f \in \mathcal{V}(\mathbb{R}^n)$, we denote with $\mathbb{B}_f := \{x \in \mathbb{R}^n \mid f(x) \leq 1\}$, the *unit ball* of the norm f.

Lemma 1.21 (Correspondence between norms and convex sets)**.** *If* $f \in \mathcal{V}(\mathbb{R}^n)$ *then* $\mathbb{B}_f \in \mathcal{K}(\mathbb{R}^n)$*, and moreover*

$$f(x) \ = \ g(x|\mathbb{B}_f), \ \ \forall\, x \in \mathbb{R}^n.$$

Conversely, for any $C \in \mathcal{K}(\mathbb{R}^n)$, $g(\cdot\,|\,C) \in \mathcal{V}(\mathbb{R}^n)$*. More explicitly,* $\mathcal{V}(\mathbb{R}^n) = \{g(\,\cdot\,|\,C) : \mathbb{R}^n \to \mathbb{R} \mid C \in \mathcal{K}(\mathbb{R}^n)\}$ *and* $\mathcal{K}(\mathbb{R}^n) = \Big\{\mathbb{B}_f \subset \mathbb{R}^n \mid f \in \mathcal{V}(\mathbb{R}^n)\Big\}$.

In other words, a set is the 1-sublevel set of the Minkowski function to which it is associated and the Minkowki function is the unique homogeneous function for which this property is satisfied.

Remark 1.22. Note that the symmetry of the gauge function $g(\cdot,|\ C)$ is equivalent to the symmetry of the convex set C. Moreover, the one-to-one correspondence between gauge functions and compact convex sets including the origin as an interior point holds. Therefore, the notion of gauge function generalizes the notion of norm to functions with similar properties (positivity, positively homogeneity, sub-additivity and convexity) but allows to consider non-symmetric unit balls.

In addition, the notion of Minkowski function in Definition 1.20 can also be defined for a closed convex (but not necessarily bounded) set including the origin in its interior. In this case, the Minkowski function is convex and positive semi-definite and for symmetric set, the Minkowski function is a semi-norm. △

Every operation on norms in Property 1.17 can be associated to an operation in Property 1.19 on their 1-sublevel sets.

Lemma 1.23 (Correspondence with Unit Balls). *Given $f_1, f_2 \in \mathcal{V}(\mathbb{R}^n)$ and $A \in \mathbb{R}^{n\times n}$ invertible, it holds that*

1. *For every $\gamma > 0$, $\mathbb{B}_{\gamma f_1} = \frac{1}{\gamma}\mathbb{B}_{f_1}$*
2. $\mathbb{B}_{f_1+f_2} = \mathbb{B}_{f_1} \sharp \mathbb{B}_{f_2}$,
3. $\mathbb{B}_{f_1 \sharp f_2} = \mathbb{B}_{f_1} + \mathbb{B}_{f_2}$,
4. $\mathbb{B}_{\max\{f_1,f_2\}} = \mathbb{B}_{f_1} \cap \mathbb{B}_{f_2}$,
5. $\mathbb{B}_{f_1 \square f_2} = \mathbb{B}_{f_1} \square \mathbb{B}_{f_2}$
6. $\mathbb{B}_{f_1 \circ A} = A^{-1}\mathbb{B}_{f_1}$
7. $\mathbb{B}_{f_1} \subseteq \mathbb{B}_{f_2} \Leftrightarrow \forall x \in \mathbb{R}^n, f_2(x) \leq f_1(x)$

We now introduce the duality of convex sets and the duality of norms, and we make the link between both of them.

Definition 1.24 (Polar Sets). Given $C \subset \mathbb{R}^n$ convex, closed and such that $0 \in C$, we define the *polar* of C, denoted by C°, by

$$C^\circ := \left\{ x \in \mathbb{R}^n \mid \sup_{y \in C} \langle y, x \rangle \leq 1 \right\}.$$

It can be proved that C° is closed, convex and $0 \in C^\circ$ and moreover, $(C^\circ)^\circ = C$.

Then, for any subset C of $\mathbb{R}^n$, $C \in \mathcal{K}(\mathbb{R}^n)$ if and only if $C^\circ \in \mathcal{K}(\mathbb{R}^n)$. Moreover we have the following relations.

Lemma 1.25. *Consider $C_1, C_2 \in \mathcal{K}(\mathbb{R}^n)$, and $A \in \mathbb{R}^{n\times n}$. Then*

1. $C_1 \subset C_2 \Leftrightarrow C_2^\circ \subset C_1^\circ$,

2. *For every* $\gamma > 0$, $(\gamma C_1)^\circ = \frac{1}{\gamma} C_1^\circ$,
3. $(C_1 + C_2)^\circ = C_1^\circ \sharp C_2^\circ$,
4. $(C_1 \sharp C_2)^\circ = C_1^\circ + C_2^\circ$,
5. $(C_1 \cap C_2)^\circ = C_1^\circ \,\square\, C_2^\circ$,
6. $(C_1 \,\square\, C_2)^\circ = C_1^\circ \cap C_2^\circ$,
7. $(AC_1)^\circ = A^{-\top} C_1^\circ$.

Definition 1.26 (Dual Norm)**.** Given $f \in \mathcal{V}(\mathbb{R}^n)$, we define the *dual norm of* f, denoted by $f^\star : \mathbb{R}^n \to \mathbb{R}$, as

$$f^\star(x) := \sup_{y \in \mathbb{R}^n \setminus \{0\}} \frac{\langle y, x\rangle}{f(y)} = \sup_{y \in \mathbb{R}^n,\, f(y)=1} \langle y, x\rangle.$$

It can be proved that $f^\star = g(\cdot \mid \mathbb{B}_f^\circ)$ (and thus $\mathbb{B}_{f^\star} = \mathbb{B}_f^\circ$) and $(f^\star)^\star = f$.

The following statement results from the consecutive application of Lemmas 1.23 and 1.25.

Lemma 1.27. *Given* $f_1, f_2 \in \mathcal{V}(\mathbb{R}^n)$ *and* $A \in \mathbb{R}^{n\times n}$ *invertible, it holds that*

1. $\forall x \in \mathbb{R}^n,\ f_1(x) \le f_2(x) \ \Leftrightarrow\ \forall x \in \mathbb{R}^n,\ f_2^\star(x) \le f_1^\star(x)$,
2. *For every* $\gamma > 0$, $(\gamma f_1)^\star = \frac{1}{\gamma} f_1^\star$
3. $(f_1 + f_2)^\star = f_1^\star \sharp f_2^\star$,
4. $(f_1 \sharp f_2)^\star = f_1^\star + f_2^\star$,
5. $(\max\{f_1, f_2\})^\star = f_1^\star \,\square\, f_2^\star$,
6. $(f_1 \,\square\, f_2)^\star = \max\{f_1^\star, f_2^\star\}$,
7. $(f_1 \circ A)^\star = f_1^\star \circ A^{-\top}$.

These results can be generalized for finite numbers of norms as summarized in the following Table 1.1.

Finally, we will see that the following result is particularly helpful for the Lyapunov formalism since it provides a characterization of a Lyapunov inequality in terms of the dual Lyapunov functions and the transpose matrix.

Lemma 1.28. *Consider* $f_1, f_2 \in \mathcal{V}(\mathbb{R}^n)$ *and* $A \in \mathbb{R}^{n\times n}$, *then*

$$(\forall\, x \in \mathbb{R}^n,\ f_2(Ax) \le f_1(x)) \ \Leftrightarrow\ \left(\forall\, x \in \mathbb{R}^n,\ f_1^\star(A^\top x) \le f_2^\star(x)\right).$$

Operation $Op(\{f_k\})$	**Unit Ball** $\mathbb{B}_{Op(\{f_k\})}$	**Dual** $(Op(\{f_k\}))^\star$	**Dual Unit Ball** $\mathbb{B}_{(Op(\{f_k\}))^\star}$
γf	$\frac{1}{\gamma}\mathbb{B}_f$	$\frac{1}{\gamma} f^\star$	$\gamma \mathbb{B}^\circ$
$\oplus_k f_k := f_1 + \cdots + f_K$	$\sharp_k \mathbb{B}_{f_k}$	$\sharp_k f_k^\star$	$\oplus_k \mathbb{B}^\circ_{f_k}$
$\sharp_k f_k := f_1 \sharp \ldots \sharp f_K$	$\oplus_k \mathbb{B}_{f_k}$	$\oplus_k f_k^\star$	$\sharp_k \mathbb{B}^\circ_{f_k}$
$\max_k\{f_k\}$	$\bigcap_k \mathbb{B}_{f_k}$	$\square_k f_k^\star$	$\square_k \mathbb{B}^\circ_{f_k}$
$\square_k f_k := f_1 \square \ldots \square f_K$	$\square_k \mathbb{B}_{f_k}$	$\max_k\{f_k^\star\}$	$\bigcap_k \mathbb{B}^\circ_{f_k}$
$f \circ A$	$A^{-1}\mathbb{B}_f$	$f^\star \circ A^{-\top}$	$A^\top \mathbb{B}^\circ_{f^\star}$

Table 1.1 Generalization of the correspondence between operations on norms and unit balls (and duality) for a finite number K of norms $\{f_k\}_{k=1,\ldots,K} \subset \mathcal{V}(\mathbb{R}^n)$.

Proof. Recalling the definitions we have $f_2(Ax) \leq f_1(x)$, $\forall x \in \mathbb{R}^n$, if and only if

$$x \in \mathbb{B}_{f_1} \;\Rightarrow\; Ax \in \mathbb{B}_{f_2}. \tag{1.9}$$

We show that Equation (1.9) is true if and only if $x \in \mathbb{B}^\circ_{f_2} \;\Rightarrow A^\top x \in \mathbb{B}^\circ_{f_1}$, which is equivalent to $f_1^\star(A^\top x) \leq f_2^\star(x), \forall x \in \mathbb{R}^n$. Suppose Equation (1.9) is true. Consider $x \in \mathbb{B}^\circ_{f_2}$, applying the definitions we have

$$\begin{aligned} x \in \mathbb{B}^\circ_{f_2} \quad &\Leftrightarrow \quad \forall y \in \mathbb{B}_{f_2},\ \langle x, y\rangle \;\leq\; 1, \\ &\Rightarrow \quad \forall z \in \mathbb{B}_{f_1},\ \langle x, Az\rangle \;\leq\; 1, \\ &\Leftrightarrow \quad \forall z \in \mathbb{B}_{f_1},\ \langle A^\top x, z\rangle \;\leq\; 1, \\ &\Leftrightarrow \quad A^\top x \in \mathbb{B}^\circ_{f_1}. \end{aligned}$$

The other direction is equivalent, once recalled that $(\mathbb{B}^\circ_{f_i})^\circ = \mathbb{B}_{f_i}$. □

It is well known that the pointwise minimum operation does not preserve the convexity property. However, we have defined the notion of infimal convolution in Property 1.17 which can be interpreted as the "convexification" of the minimum operation. The following lemma discusses the link between these operations in the Lyapunov framework.

Lemma 1.29. *Consider $f_1, f_2, g_1, g_2 \in \mathcal{V}(\mathbb{R}^n)$ four norms, and a matrix $A \in \mathbb{R}^{n\times n}$. Then*

$$\min\{f_2(Ax), g_2(Ax)\} \leq \min\{f_1(x), g_1(x)\}, \forall x \in \mathbb{R}^n \tag{1.10}$$

implies that

$$(f_2 \,\square\, g_2)(Ax) \leq (f_1 \,\square\, g_1)(x), \forall x \in \mathbb{R}^n_{\geq 0}.$$

Proof. Without loss of generality, consider $x \in \mathbb{R}^n$ such that $x \in \mathbb{B}_{f_1 \square g_1}$, i.e. $f_1 \square g_1(x) \leq 1$. By Lemma 1.23 and by definition of the infimal convolution in Property 1.17,

$$x \in \mathbb{B}_{f_1 \square g_1} = \mathbb{B}_{f_1} \square \mathbb{B}_{g_1} = \operatorname{conv}\left\{\mathbb{B}_{f_1} \cup \mathbb{B}_{g_1}\right\}.$$

Using the same arguments, we have to prove that $Ax \in \operatorname{co}\{\mathbb{B}_{f_2} \cup \mathbb{B}_{g_2}\}$. Let us write $x := \lambda x_1 + (1-\lambda)x_2$ with $x_1, x_2 \in \mathbb{B}_{f_1} \cup \mathbb{B}_{g_1}$. By Equation (1.10), we know that Ax_1, $Ax_2 \in \mathbb{B}_{f_2} \cup \mathbb{B}_{g_2}$. Then

$$Ax = A(\lambda x_1 + (1-\lambda)x_2) = \lambda Ax_1 + (1-\lambda)Ax_2 \in \operatorname{conv}\left\{\mathbb{B}_{f_2} \cup \mathbb{B}_{g_2}\right\},$$

which concludes the proof. □

This result essentially indicates that, in the context of a linear switched system, if we have an inequality involving minima, we can derive a corresponding inequality involving their convex hull, i.e. the infimal convolution. For that reason, in this setting, for a family closed under infimal convolution, we can use all we know for the minimum. While this result holds true in the context of linear switched systems, they do not generalize to the generic nonlinear setting.

1.3 Templates and closure properties

So far, we have introduced Lyapunov theory to synthesize stability certificates for switched systems. In practice, we usually restrict our search to subclasses of candidate Lyapunov functions, called *templates*, such as quadratic norms or sum-of-squares polynomials, see Section 2.2. However the conservatism of the corresponding stability certificate depends on the template that we use. In particular, this thesis will highlight the importance of the template and especially its *closure properties* in the conservatism-based comparison of stability certificates.

In order to further introduce *path-complete Lyapunov functions* and properly define their comparison in Chapter 6, we need to formally define the notion of *template* (independent of the dimension).

Definition 1.30 (Template of Lyapunov functions)**.** A *template* of candidate Lyapunov functions is defined as a family of countably many sets of Lyapunov functions of fixed dimension, i.e.

$$\mathcal{V} := \bigcup_{n \in \mathbb{N}} \mathcal{V}_n,$$

where $\mathcal{V}_n \subseteq \mathcal{C}^0_+(\mathbb{R}^n, \mathbb{R})$.

Remark 1.31. In Definition 1.30, we define a template as a countable infinite union of fixed dimensional templates $\mathcal{V}_n \subseteq \mathcal{C}^0_+(\mathbb{R}^n, \mathbb{R})$. This choice is partly motivated by our interest in the closure properties of these templates, as properly defined in Definition 1.33, which do not generally depend on the dimension. However, given a n-dimensional switched system, the practical search for a candidate Lyapunov function within such a template is performed in $\mathcal{V}_n$. △

This definition allows us to consider classical Lyapunov functions for instance, such as the quadratic ones. In this case, the set $\mathcal{V}_n$ for $n \in \mathbb{N}$ will contain all the quadratic functions $V(x) = x^\top P x$ with $P \in \mathbb{R}^{n \times n}$ positive definite.

In the linear case, we can consider a *template of norms* without loss of conservatism (see Theorem 2.9), i.e. a template whose elements are norms. Using duality theory introduced in Section 1.2, we can define the dual of a given template of norms.

Definition 1.32 (Dual template)**.** Given a template $\mathcal{V}$ of norms, the *dual template*, denoted by $\mathcal{V}^*$, is defined as the set of the dual norms, i.e.

$$\mathcal{V}^* := \{g^* \mid g \in \mathcal{V}\}.$$

It is well known that the template influences the conservatism of a stability certificate. Moreover, we will demonstrate in Part III with theoretical results and through numerous examples that the *closure properties* of this template play a key role in the comparison of the corresponding stability certificates with respect to their conservatism.

Similarly to Property 1.17 where we introduce operations which preserve convesity, we need to define *families of binary operations* which *preserve the template*.

Definition 1.33 (Closure properties of a template)**.** Consider a template $\mathcal{V} = \cup_{n\in\mathbb{N}}\mathcal{V}_n$ of candidate Lyapunov functions and a family of binary operations $\{\star_n : \mathcal{C}^0_+(\mathbb{R}^n, \mathbb{R}) \times \mathcal{C}^0_+(\mathbb{R}^n, \mathbb{R}) \to \mathcal{C}^0_+(\mathbb{R}^n, \mathbb{R})\}_{n\in\mathbb{N}}$.

(a) For a fixed dimension $n \in \mathbb{N}$, we say that the set of functions $\mathcal{V}_n$ is *closed under the binary operation* $\star_n$ if

$$\forall f_1, f_2 \in \mathcal{V}_n : \ f_1 \star_n f_2 \in \mathcal{V}_n.$$

(b) We say that the *template* $\mathcal{V}$ *is closed under the family of binary operations* $\{\star_n\}_{n\in\mathbb{N}}$ *if for all* $n \in \mathbb{N}$*, the set* $\mathcal{V}_n$ *is closed under* $\star_n$*.*

Moreover, we say that *an operation preserves the Lyapunov properties* if the template of candidate Lyapunov functions, i.e. $\mathcal{V}_n := \mathcal{C}^0_+(\mathbb{R}^n, \mathbb{R})$, is closed under this operation.

In Chapter 7, we will investigate the operations of addition, pointwise minimum and maximum, and finally the composition with the dynamics.

2

Joint spectral radius

In classical linear systems theory, a typical question is whether a *discrete-time linear system* $x_{k+1} := A\, x_k$ is *asymptotically stable*, that is, whether all the trajectories asymptotically tend to zero. A necessary and sufficient condition for this is that the *spectral radius* of the square matrix $A \in \mathbb{R}^{n \times n}$ with $n \in \mathbb{N}$ defined by

$$\rho(A) \;:=\; \lim_{k \to \infty} \left\| A^k \right\|^{1/k}$$

is less than one. The spectral radius expresses the maximal rate of growth of the system, and has been proved to be equal to the maximal modulus of the eigenvalues of the matrix A.

However, when we consider a linear switched system (1.2), the dynamical system is defined by *several* matrices $\mathcal{A} := \{A_1, \ldots, A_M\}$ rather than a single one, as is the case for classical linear systems. As a consequence, the stability problem becomes far more complex since for instance, each matrix can be stable without the switched system being stable. This has motivated the generalization of the spectral radius to the finite set of square matrices $\mathcal{A}$ by the introduction of the *joint spectral radius* (JSR for short) denoted by $\rho(\mathcal{A})$ and introduced in [RS60]. Similarly to the linear case above, the JSR expresses the maximal asymptotic behaviour of a switched linear system. The joint spectral radius of $\mathcal{A}$ can be related to the stability of (1.2) by the following statement: the linear switched system $\mathcal{A}$ is asymptotically stable (under arbitrary switching) if and only if $\rho(\mathcal{A}) < 1$.

Unfortunately, the JSR does not share the same nice algebraic properties as the spectral radius and is extremely hard to compute in practice. Indeed, approximating the JSR is NP-hard [BT97], whereas deciding whether the JSR is smaller than one or not is Turing-indecidable [BC08]. Moreover, there does not exist any algebraic criterion to decide the stability (non-algebraicity) of switched systems [Koz90]. Despite these theoretical limitations, the approximation of the JSR has been investigated by many researchers. In particular, Lyapunov methods have been exploited. In practice, one usually looks through a particular Lyapunov function for which the computation can be performed more easily. This is for instance the case for the *ellipsoidal* approximation of the JSR, as studied in [BNT05]. A similar approach using *SOS polynomials* instead of quadratic functions has been developed in [PJ08].

In this section, we summarize the most popular Lyapunov techniques, the corresponding computation methods and the approximation guarantees using these templates.

2.1 Definition and properties

Given a linear switched system

$$x(k+1) \;=\; A_{\sigma(k)}x(k), \tag{2.1}$$

with $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n \times n}$, let us define the *joint spectral radius* of $\mathcal{A}$ as the maximum rate of growth of the corresponding linear switched system. Note that in the rest of this section, we assume that the norms are submultiplicative.

Definition 2.1 (Joint spectral radius)**.** Given a finite set of M square matrices $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n \times n}$, the *joint spectral radius* (i.e. JSR for short), denoted by $\rho(\mathcal{A})$, is defined as

$$\rho(\mathcal{A}) \;:=\; \lim_{k\to\infty} \max_{A \in \mathcal{A}^k} \|A\|^{1/k},$$

where $\mathcal{A}^k := \{A_{i_1} \ldots A_{i_k} \mid A_{i_j} \in \mathcal{A}, j = 1, \ldots, k\}$ for any $k \in \mathbb{N}$.

This quantity does not provably depend on the norm by equivalence of the norms in finite dimension.

Let us recall some basic properties of the JSR.

Proposition 2.2 (Proposition 1.2 in [Jun09])**.** *For any finite set $\mathcal{A} \subset \mathbb{R}^{n\times n}$ and for any real number λ,*

$$\rho(\lambda\mathcal{A}) \;=\; |\lambda|\,\rho(\mathcal{A}).$$

The following proposition provides both upper and lower bound on the JSR. This result is often used to compute the JSR of a given set of matrices.

Proposition 2.3 (Three members inequalities, Proposition 1.6 in [Jun09])**.** *For any finite set of $\mathcal{A} \subset \mathbb{R}^{n\times n}$ and for any $t \in \mathbb{N}$,*

$$\rho_t(\mathcal{A}) \;\leq\; \rho(\mathcal{A}) \;\leq\; \hat{\rho}_t(\mathcal{A}, \|\cdot\|),$$

where $\rho_t(\mathcal{A}) := \max_{A\in\mathcal{A}^t} \rho(A)^{1/t}$ and $\hat{\rho}_t(\mathcal{A}, \|\cdot\|) := \max_{A\in\mathcal{A}^t} \|A\|^{1/t}$.

Example 2.4 (Example 1.1 in [Jun09]). Consider the set of two matrices

$$\mathcal{A} \;:=\; \left\{ \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \right\}.$$

One can prove that $\rho(\mathcal{A}) = \sqrt{2}$ using Proposition 2.3. Indeed, by multiplying the two matrices, one obtains the matrix

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \;=\; \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}.$$

The square root of the spectral radius of this matrix, that is $\sqrt{2}$, is a lower bound on the JSR of $\mathcal{A}$. Moreover, $\hat{\rho}_2(\mathcal{A}, \|\cdot\|_1) = \sqrt{2}$. Then, one can conclude by Proposition 2.3 that $\rho(\mathcal{A}) = \sqrt{2}$. △

The following result characterizes the JSR in terms of the matrices in $\mathcal{A}$, without any product of these matrices.

Proposition 2.5 (Proposition 1.4 in [Jun09])**.** *For any finite set $\mathcal{A} \subset \mathbb{R}^{n\times n}$ such that $\rho(\mathcal{A}) \neq 0$, the joint spectral radius can be defined as*

$$\rho(\mathcal{A}) \;=\; \inf_{\|\cdot\|} \max_{A\in\mathcal{A}} \{\|A\|\}.$$

This result is useful because one can bound the JSR by finding a good norm to obtain an estimate of the JSR, since for any matrix norm $\|\cdot\|$,

$$\rho(\mathcal{A}) \;\leq\; \max_{A\in\mathcal{A}} \|A\|\,.$$

If the latter inequality is tight, the norm is called *extremal*.

Definition 2.6 (Extremal matrix norm)**.** A matrix norm $\|\cdot\|$ on $\mathbb{R}^{n\times n}$ is *extremal* for a finite set of matrices $\mathcal{A} \subset \mathbb{R}^{n\times n}$ if for all $A \in \mathcal{A}$,

$$\|A\| \;\leq\; \rho(A).$$

By Proposition 2.5, this implies that $\rho(\mathcal{A}) = \sup_{A\in\mathcal{A}} \|A\|$. However, such a norm does not always exist, see [Jun09, Chapter 2.1] for a complete discussion on this topic.

Definition 2.7 (Extremal vector norm)**.** A vector norm $\|\cdot\|$ on $\mathbb{R}^n$ is *extremal* for a finite set of matrices $\mathcal{A} \subset \mathbb{R}^{n\times n}$ if for all $x \in \mathbb{R}^n$ and for all $A \in \mathcal{A}$,

$$\|Ax\| \;\leq\; \rho(\mathcal{A})\,\|x\|\,.$$

Then, the matrix norm induced by an extremal vector norm is an extremal matrix norm. As with the extremal matrix norm, there does not always exist an extremal vector norm. However, the following proposition states that there exists a vector norm which provides an ε-close approximation of the JSR, for any precision $\varepsilon > 0$.

Proposition 2.8 (Theorem 2.1 in [PJ08])**.** *Consider a finite set of matrices* $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n\times n}$*. For any* $\varepsilon > 0$*, there exists a (vector) norm* $\|\cdot\|_\varepsilon$ *on* $\mathbb{R}^n$ *such that*

$$\|A_i x\|_\varepsilon \;\leq\; (\rho(\mathcal{A}) + \varepsilon)\,\|x\|_\varepsilon\,,\ \forall x \in \mathbb{R}^n, \forall i \in \langle M\rangle.$$

The following theorem presents the main property of the JSR since it characterizes the stability of a linear switched system.

Theorem 2.9 (Corollary 1.1 in [Jun09])**.** *For any finite set of matrices* $\mathcal{A} \subset \mathbb{R}^{n\times n}$*, the corresponding linear switched system is stable (under arbitrary switching) if and only if* $\rho(\mathcal{A}) < 1$*.*

A similar characterization for constrained linear switched systems can be stated using the corresponding *constrained joint spectral radius*. See [Dai12, Phi17] for further details on this topic.

Example 2.10 (Example from [Jun09])*.* Consider the set of matrices $\mathcal{A} := \{A_1, A_2\} \subset \mathbb{R}^{2\times 2}$ defined by

$$A_1 \;:=\; \frac{2}{3}\begin{bmatrix} \cos(1.5) & \sin(1.5) \\ -2\sin(1.5) & 2\cos(1.5) \end{bmatrix} \;\text{ and }\; A_2 \;:=\; \frac{2}{3}\begin{bmatrix} 2\cos(1.5) & 2\sin(1.5) \\ -\sin(1.5) & 2\cos(1.5) \end{bmatrix}.$$

Both matrices are stable since $\rho(A_1) = \rho(A_2) = 0.9428$ but if we combine the two matrices alternatively, then it becomes unstable. In particular, $\sqrt{\rho(A_1A_2)} = \sqrt{1.751} = 1.323 > 1$. Then, by Proposition 2.3, $\rho(\mathcal{A}) \geq 1$ and the corresponding switched system is unstable by Theorem 2.9.

Although Theorem 2.9 might be appealing, the following results show that the numerical computation/approximation of the JSR can be difficult in practice. In particular, it is NP-hard to approximate, as close as desired, the joint spectral radius.

Theorem 2.11 (NP-hardness, Theorem 2.4 in [Jun09])**.** *Unless $P = NP$, there is no algorithm that, given a set of matrices $\mathcal{A}$ and a relative accuracy ε, returns an estimate $\tilde{\rho}(\mathcal{A})$ of $\rho(\mathcal{A})$ such that $|\tilde{\rho}(\mathcal{A}) - \rho(\mathcal{A})| \leq \varepsilon\rho(\mathcal{A})$ in a number of steps that is polynomial in the size of $\mathcal{A}$ and ε.*

In addition, it has been proved that it is not possible to construct an algorithm that always leads to a correct answer to the question of deciding whether the joint spectral radius of a finite set of matrices is smaller than 1.

Theorem 2.12 (Undecidability, Theorem 2.6 in [Jun09])**.** *The problem of determining, given a set of matrices $\mathcal{A}$, if $\rho(\mathcal{A}) \leq 1$ is Turing-undecidable.*

2.2 Classical Lyapunov computation techniques

Even though Theorems 2.11 and 2.12 suggest that the computation of the JSR might be laborious, these statements have not prevented researchers from developing several approximation techniques. For instance, the JSR toolbox [VHJ14] compiles several recent computation and approximation methods, by selecting the most appropriate methods based on an automatic study of the matrix set provided. In particular, two different types of approximation methods can be distinguished.

The first class of methods takes advantage of the three members inequalities in Proposition 2.3 with recursively longer and longer products of matrices. See Branch and bound methods summarized in [Jun09, Section 2.3.3].

The second class of methods of the JSR approximation relies on the Lyapunov theory, previously introduced in Section 1.1.2. Without loss of generality, Theorem 2.9 and Proposition 2.5 ensure that we can look for norms as candidate Lyapunov functions. However, this common contractive norm (when it exists) is usually hard to find and not finitely constructible. The following result relaxes the convexity property, and states that any continuous, positive and homogeneous function provides an approximation on the JSR.

Theorem 2.13. *Consider a finite set of matrices $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^n$. If there exists a positive, continuous and homogeneous function $V : \mathbb{R}^n \to \mathbb{R}$ such that*

$$\forall i \in \langle M \rangle,\ \forall x \in \mathbb{R}^n :\ V\left(\frac{A_i}{\gamma} x\right) \ \leq\ V(x) \tag{2.2}$$

then $\rho(\mathcal{A}) \leq \gamma$.

Sketch of the proof. Assume that the candidate Lyapunov function V is homogeneous of degree d. Then, by Lemma 1.11 and Remark 1.13, there exist α, β with $0 < \alpha \leq \beta$ such that

$$\forall x \in \mathbb{R}^n :\ \alpha \left\|x\right\|^d \ \leq\ V(x) \ \leq\ \beta \left\|x\right\|^d .$$

Consider any arbitrary finite product $A_{w(k)} \ldots A_{w(1)}$ of length $k \in \mathbb{N}$ of the matrices in $\mathcal{A}$. The consecutive application of the Lyapunov inequalities in Equation (2.2) implies that

$$\forall x \in \mathbb{R}^n :\ V\left(\frac{A_{w(k)} \ldots A_{w(1)} x}{\gamma^k}\right) \ \leq\ V(x).$$

By homogeneity and after rearrangement, we have

$$\forall x \in \mathbb{R}^n :\ \left(\frac{V(A_{w(k)} \ldots A_{w(1)} x)}{V(x)}\right)^{1/d} \ \leq\ \gamma^k.$$

Therefore, one can bound the norm of the product $A_{w(k)} \ldots A_{w(1)}$ as follows:

$$\begin{aligned}
\left\|A_{w(k)} \ldots A_{w(1)}\right\| \ &\leq\ \max_x \frac{\left\|A_{w(k)} \ldots A_{w(1)} x\right\|}{\|x\|}, \\
&\leq\ \left(\frac{\beta}{\alpha}\right)^{1/d} \max_x \frac{V(A_{w(k)} \ldots A_{w(1)} x)^{1/d}}{V(x)^{1/d}}, \\
&\leq\ \left(\frac{\beta}{\alpha}\right)^{1/d} \gamma^k.
\end{aligned}$$

Then, by Definition 2.1 and after taking the k-th root and the limit when $k \to \infty$, we can conclude that $\rho(\mathcal{A}) \leq \gamma$. □

In the rest of this section, we summarize classical Lyapunov JSR approximation methods using three different templates, namely the quadratic Lyapunov functions, the sum-of-squares polynomials and the linear copositive norms. For each template, we provide both approximation guarantees and practical computation techniques.

2.2.1 Quadratic functions

In this section, we consider the approximation of the JSR using a common *quadratic Lyapunov norm* whose computation can be expressed as a convex optimization problem. This method has been studied in details in [BNT05], and summarized in [Jun09, Section 2.3.7].

Let us first formally define a quadratic norm, also called ellipsoidal norm.

Definition 2.14 (Quadratic/ellipsoidal form). Let Q be a square positive definite matrix, then

$$\|x\|_Q := \sqrt{x^\top Q x}, \tag{2.3}$$

is called the *quadratic* or the *ellipsoidal (vector) norm associated to* Q. The induced matrix norm

$$\|A\|_Q := \max_{\|x\|_Q = 1} \|Ax\|_Q$$

is called the *quadratic* or *ellipsoidal (matrix) norm associated to* Q.

Note that this denomination stems from the shape of the unit ball $\mathcal{B}_{\|\cdot\|_Q}$ of this norm.

Remark 2.15. The matrix representation $p(x) = x^T Q x$ is not unique since several positive definite matrices lead to the same quadratic polynomial. However, it is possible to define a unique *symmetric* positive definite matrix by defining $(Q + Q^\top)/2$. Then, in the rest of this manuscript, we will assume without loss of generality that the matrix Q in Definition 2.14 is symmetric. △

One can show that the quadratic template is self-dual in the sense of Definition 1.26.

Proposition 2.16 (Self-duality)**.** *The quadratic template is self-dual, i.e. the dual norm of a quadratic norm is a quadratic norm.*

Sketch of the proof: Given a quadratic norm $\|\cdot\|_Q$ on $\mathbb{R}^n$ with $Q \in \mathbb{R}^{n\times n}$ a symmetric positive definite matrix, one can prove that the dual norm is the quadratic norm $\|\cdot\|_{Q^{-1}}$. □

The following proposition characterizes the quadratic matrix norm, and provides a method to compute it.

Proposition 2.17 (Proposition 2.7 in [Jun09])**.** *Given a symmetric positive definite matrix Q, the quadratic matrix norm $\|A\|_Q$ of a matrix A is the smallest $\gamma \in \mathbb{R}^+$ such that the following equation has a solution:*

$$A^\top Q A \preceq \gamma^2 Q.$$

Then, the computation of this minimal γ can be efficiently done since it amounts to compute the spectral radius of the following matrix

$$\gamma^2 := \rho(L^{-1} A^\top Q A L^{-1^\top}),$$

where $Q = LL^\top$ is the Cholesky factorisation of Q.

We define the *quadratic approximation of the JSR* as the best JSR approximation using the template of quadratic norms.

Definition 2.18 (Quadratic/ellipsoidal approximation of the JSR)**.** Given a finite set of matrices $\mathcal{A} := \{A_1, \dots, A_M\} \subset \mathbb{R}^{n\times n}$, the (common) *quadratic* or *ellipsoidal* approximation of the JSR, denoted by $\rho_Q(\mathcal{A})$ is defined by

$$\rho_Q(\mathcal{A}) := \inf_{Q\succ 0} \max_{A\in\mathcal{A}} \|A\|_Q .$$

From a computational point of view, the ellipsoidal approximation $\rho_Q(\mathcal{A})$ can be computed efficiently thanks to SDP, see [BNT05] for details.

Proposition 2.19 (Proposition 2.8 in [Jun09])**.** *For any finite set of matrices $\mathcal{A} := \{A_1, \dots, A_M\}$ and positive value $\gamma > 0$, if there is a solution Q to the following SDP program:*

$$\begin{aligned} {A_i}^\top Q A_i &\preceq \gamma^2 Q, \quad \forall A_i \in \mathcal{A}, \\ Q &\succ 0, \end{aligned} \tag{2.4}$$

then $\rho(\mathcal{A}) \leq \gamma$.

The quality of the quadratic approximation of the JSR can be measured. These guarantees mainly rely on the approximation capabilities of convex sets by ellipsoids, stated in the following lemma.

Lemma 2.20 (John's ellipsoid theorem, [Joh48])**.** *Let $K \in \mathbb{R}^n$ be a compact convex set with nonempty interior. Then there is an ellipsoid E with center c such that the inclusions $E \subset K \subset n(E-c)+c$ hold. If K is symmetric about the origin ($K = -K$), the constant n can be changed into $\sqrt{n}$.*

It is finally possible to derive the following approximation guarantees using quadratic norms as Lyapunov functions.

Theorem 2.21 (Theorem 14 in [BNT05])**.** *Let $\rho(\mathcal{A})$ be the joint spectral radius of a finite set of matrices $\mathcal{A}$ of dimension $n \in \mathbb{N}$. Then*

$$\frac{1}{\tau_{\mathcal{Q}}}\rho_{\mathcal{Q}}(\mathcal{A}) \;\le\; \rho(\mathcal{A}) \;\le\; \rho_{\mathcal{Q}}(\mathcal{A}), \tag{2.5}$$

where $\tau_{\mathcal{Q}} := \sqrt{n}$.

Sketch of the proof. By Proposition 2.8, for any $\varepsilon > 0$, there exists a vector norm $\|\cdot\|_\varepsilon$ for which

$$\forall i \in \langle M \rangle,\ \forall x \in \mathbb{R}^n :\ \|A_i x\|_\varepsilon \;\le\; (\rho(\mathcal{A})+\varepsilon)\,\|x\|_\varepsilon\,.$$

The sublevel sets of this (symmetric) norm can be approximated using ellipsoids and Lemma 2.20 measures the approximation quality, i.e. there exists a quadratic norm $\|\cdot\|_P := x^\top P x$ with $P \succ 0$ such that

$$\forall x \in \mathbb{R}^n :\ \|x\|_P \;\le\; \|x\|_\varepsilon \;\le\; \sqrt{n}\,\|x\|_P\,.$$

Therefore, for every mode $i \in \langle M \rangle$:

$$\forall x \in \mathbb{R}^n :\ \|A_i x\|_P \;\le\; \|A_i x\|_\varepsilon \;\le\; q\,\|x\|_\varepsilon \;\le\; q\sqrt{n}\,\|x\|_P\,,$$

which is equivalent to requiring

$$A_i{}^\top P A_i - q^2 n P \;\preceq\; 0.$$

Then, the ellipsoidal approximation of $\mathcal{A}$ is smaller than $q\sqrt{n}$ for any $q > \rho(\mathcal{A})$. Then, at worst, we have that

$$\rho_{\mathcal{Q}}(\mathcal{A}) \;\le\; \rho(\mathcal{A})\sqrt{n}.$$

□

2.2.2 Sum of squares polynomials

Unfortunately, it can be proved that the quadratic template is not *universal*, meaning that there does not exist a quadratic Lyapunov function for any stable switched system. In particular, [PJ08, Example 2.8] provides an example of a finite set of matrices which are stable but there does not exist a common quadratic Lyapunov function.

In this section, we investigate the idea of considering homogeneous polynomials of higher degree as candidate Lyapunov functions to approximate the JSR. This has been extensively introduced in [PJ08] and summarized in [Jun09, Section 2.3.7].

Theorem 2.22 (Theorem 2.2 in [PJ08])**.** *Consider a finite set of matrices* $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n\times n}$ *and* $\gamma > 0$*. Let* $p(x)$ *be a strictly positive homogeneous polynomial of degree* $2d$ *that satisfies*

$$p(A_i x) \ \le\ \gamma^{2d} p(x),\ \forall x \in \mathbb{R}^n,\ \forall i \in \langle M \rangle.$$

Then, $\rho(\mathcal{A}) \le \gamma$*.*

However, it is computationally hard to characterize positive polynomials. Then, we consider the *sum-of-squares* (SOS for short) relaxation which requires that the polynomial must admit a sum of squares decomposition, as defined below.

Definition 2.23 (Sum-of-squares polynomial)**.** A polynomial p of degree $2d$ is a *sum-of-squares* (SOS for short) if there exist some polynomials q_1, $\ldots$, q_M (not all zero) of degree d such that

$$p(x) \ :=\ \sum_{i=1}^{M} q_i(x)^2.$$

It follows directly that every SOS polynomial is positive semi-definite. However, the converse is not always true. One of the most common counterexample is probably the Motzkin polynomial, i.e. $p(x,y) = x^2y^4 + x^4y^2 + 1 - 3x^2y^2$, which is positive semi-definite but it is not a sum of squares.

In fact, it has been proved in [Hil88] that the characterization of positivity by SOS polynomials is only valid in three particular cases.

Theorem 2.24 (Characterization of positive polynomials by SOS polynomials [Hil88]). *The equivalence between the nonnegativity of the polynomials of n variables and degree $2d$ and the existence of a sum of squares decomposition holds in the three following cases:*

(a) $n = 1$ *(univariate polynomials);*

(b) $2d = 2$ *(quadratic polynomials);*

(c) $n = 2$ *et* $2d = 4$ *(bivariate quartics).*

However, one of the advantages of SOS polynomial is that it is easy to check (as opposed to check if a polynomial is positive definite, which is NP-hard). Indeed, consider a polynomial $f(x)$ of degree $2d$ in dimension n with $d, n \in \mathbb{N}$ and z the vector which encodes all the monomials of degree d in dimension n of dimension $N := \frac{(n+d-1)!}{d!(n-1)!}$. Then, f is a SOS-polynomial if and only if

$$\exists Q \succ 0 \ : \ f(x) = z^\top Qz,$$

which can be solved by an SDP. Then, by the Cholesky decomposition theorem, we can find a triangular matrix L such that $Q = L^\top L$. Therefore,

$$f(x) \ = \ z^\top L^\top Lz \ = \ \sum_i (Lz)_i^{\ 2},$$

where $(Lz)_i$ is a polynomial of degree d. Then, each SOS-polynomial of degree $2d$ can be written as the sum of (at most) N squared polynomials of degree d, and the number of squares in the SOS-decomposition is equal to the rank of the matrix Q.

Theorem 2.25 (Theorem 2.3 in [PJ08]). *A homogeneous multivariate polynomial $p(x)$ of degree $2d$ is a sum of squares if and only if*

$$p(x) \ := \ \left(x^{[d]}\right)^\top Qx^{[d]},$$

where $x^{[d]}$ is a vector whose entries are (possibly scaled) monomials of degree d in the variables x_i, and Q is a symmetric positive semidefinite matrix.

We can now define the JSR approximation using SOS polynomial by requiring that the polynomials $p(x)$ and $\gamma^{2d}p(x) - p(A_i x)$ are SOS in Theorem 2.22. Then we look for the infimum γ for which there exists $p(x)$ such that this condition is satisfied.

Definition 2.26 (SOS approximation of the JSR)**.** Given a finite set of matrices $\mathcal{A} := \{A_1, \ldots, A_M\}$, the *SOS approximation of the JSR of degree* $2d$, denoted by $\rho_{SOS,2d}(\mathcal{A})$, is defined by

$$\rho_{SOS,2d}(\mathcal{A}) := \inf_{p \in \mathbb{R}_{2d}[x], \gamma > 0} \gamma \text{ s.t. } \begin{cases} p(x) & \text{is SOS,} \\ \gamma^{2d} p(x) - p(A_i x) & \text{is SOS,} \end{cases} \tag{2.6}$$

where $\mathbb{R}_{2d}[x]$ refers to the set of homogeneous polynomials of degree $2d$.

Remark 2.27*.* Theorem 2.22 initially requires a positive definite polynomial, while strict positivity is not required in Equation (2.6). However, since current algorithms to solve SDPs, such as the interior-point methods, always produce solutions in the relative interior of the SOS cone, this is automatically satisfied if the problem is feasible. See [PJ08, Remark 2.5] for further details. △

Therefore, the SOS relaxation yileds to an upper-bound on the JSR, i.e. $\rho(\mathcal{A}) \leq \rho_{SOS,2d}(\mathcal{A})$. The following lemma quantifies the quality of this approximation.

Lemma 2.28 (Theorem 2.6 in [PJ08])**.** *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. For any integer $d \geq 1$, there exists a homogeneous polynomial $p(x)$ in n variables of degree $2d$ such that*

(1) The polynomial $p(x)$ is SOS,

(2) For all $x \in \mathbb{R}^n$,

$$p(x)^{1/2d} \leq \|x\| \leq k(n,d)\, p(x)^{1/2d},$$

where $k(n,d) := \binom{n+d-1}{d}^{1/2d}$.

It is now possible to derive approximation guarantees with the template of SOS polynomials.

Theorem 2.29 (Theorem 3.4 in [PJ08])**.** *The SOS approximation of degree $2d$, $d \in \mathbb{N}$ of the JSR denoted by $\rho_{SOS,2d}(\mathcal{A})$ satisfies the following inequalities*

$$\frac{1}{\tau_{SOS}(n,d)^{1/2d}} \rho_{SOS,2d}(\mathcal{A}) \leq \rho(\mathcal{A}) \leq \rho_{SOS,2d}(\mathcal{A}), \tag{2.7}$$

where $\tau_{SOS}(n,d) = k(n,d)$ in Lemma 2.28.

Sketch of the proof: The proof follows the same path of ideas as in the proof of Theorem 2.21. In particular, it relies on the consecutive application of Proposition 2.8 and Lemma 2.28 which discusses the precision of the approximation of a norm by a SOS polynomial. Additional results are needed to formally end the proof, see [PJ08] for further details. □

2.2.3 Linear copositive norms

In this section, we consider and develop two particular functions templates, the *primal/dual linear copositive norms* to study the stability of *positive linear switched systems* of the form

$$x(k+1) = A_{\sigma(k)}x(k) \tag{2.8}$$

where $\sigma : \mathbb{N} \to \langle M \rangle$, and $\mathcal{A} := \{A_1, \dots, A_M\} \subset \mathbb{R}^{n \times n}$ is a set of nonnegative matrices. Positive switched systems (2.8) are popular for modeling the dynamics of phenomena constrained in the positive cone $\mathbb{R}^n_{\geq 0}$, and from our point of view, they provide a simple "practical" setting in order to illustrate the developments of the following sections. In the context of positive switched systems, primal norms as in Equation (2.9) were considered in [MS07, FV12, JZHZH13].

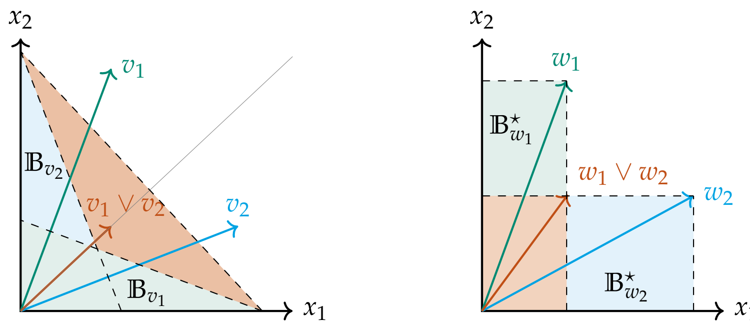
Definition 2.30 (Primal and dual copositive norms)**.** Given $v \in \mathbb{R}^n_{>0}$, we define the *primal* and *dual linear copositive norms induced by* v on $\mathbb{R}^n_{\geq 0}$ by

$$g_v(x) := v^\top x, \tag{2.9}$$

$$\text{and} \quad g^\star_v(x) := \max_i \left\{ \frac{x_i}{v_i} \right\}, \tag{2.10}$$

for all $x \in \mathbb{R}^n_{\geq 0}$. We denote with $\mathcal{P}$ and $\mathcal{D}$ the set of all primal and dual copositive norms, respectively.

One can show that the norms defined in Equation (2.10) are exactly the dual of the ones in Equation (2.9), in the sense of Definition 1.26 (see Figure 2.1 for a graphical interpretation of this class of functions). It follows from Definition 2.30 that, given any $v \in \mathbb{R}^n_{>0}$, the functions $g(\cdot)_v, g(\cdot)^\star_v : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$ are positive definite and radially unbounded, and thus $\mathcal{P}$ and $\mathcal{D}$ represent legitimate templates when studying stability of (2.8).



(a) The unit balls $\mathbb{B}_{v_1}$ and $\mathbb{B}_{v_2}$ are represented in red and blue, respectively. In orange, the ball $\mathbb{B}_{v_1 \vee v_2} = \mathbb{B}_{v_1} \square \mathbb{B}_{v_2}$.

(b) The unit balls $\mathbb{B}^{\star}_{w_1}$ and $\mathbb{B}^{\star}_{w_2}$ are represented in red and blue, respectively. In orange, we represent $\mathbb{B}^{\star}_{w_1 \vee w_2} = \mathbb{B}^{\star}_{w_1} \cap \mathbb{B}^{\star}_{w_2}$.

Fig. 2.1 Examples of primal and dual copositive norms on $\mathbb{R}^2_{\geq 0}$, with the corresponding unit balls.

In order to highlight the useful properties of these templates, we adopt the following notation.

Definition 2.31. Given $v, w \in \mathbb{R}^n_{>0}$, we define $v \vee w \in \mathbb{R}^n_{>0}$ as

$$v \vee w \;:=\; \sum_i \min\{v_i, w_i\}\, \mathbf{e}_i, \tag{2.11}$$

i.e. the componentwise minimum between v and w.

We introduce the following useful properties.

Proposition 2.32. *Given any vectors $v, w \in \mathbb{R}^n_{>0}$, any matrix $A \in \mathbb{R}^{n\times n}_{\geq 0}$ and any $\lambda > 0$, we have*

(1) $g_{v+w} \;=\; g_v + g_w$*;*

(2) $g_{\lambda v} \;=\; \lambda g_v$*;*

(3) $g_{v \vee w} \;=\; g_v \,\square\, g_w$ *(infimal convolution of primal norms is a primal norm);*

(4) $\left(\forall x \in \mathbb{R}^n_{\geq 0},\; g_v(Ax) \leq g_w(x)\right) \;\Leftrightarrow\; A^\top v \leq_c w.$

Proof. Items *(1)*, *(2)* and *(4)* follow directly from Definition 2.30 and 2.31. For Item *(3)*, we need to prove that

$$\mathbb{B}_{g_{v\vee w}} \;=\; \mathbb{B}_{g_v \square g_w},$$

using the correspondence between norms and unit balls in Lemma 1.21. By Lemma 1.23, we know that $\mathbb{B}_{g_v \square g_w} = \mathbb{B}_{g_v} \square \mathbb{B}_{g_w}$. We thus need to prove that

$$\mathbb{B}_{g_{v\vee w}} \; = \; \mathrm{conv}\{\mathbb{B}_{g_v} \cup \mathbb{B}_{g_w}\},$$

then the result follows from the correspondence between norms and their unit balls, see Lemma 1.21.

($\subset$): Without loss of generality, consider $x \in \mathbb{B}_{g_{v\vee w}}$ such that $g_{v\vee w}(x) = 1$, i.e. $\sum_i \min\{v_i, w_i\} x_i = 1$. Define the index sets

$$\begin{aligned} I_v &:= \{i \in \{1,\dots,n\} \mid v_i < w_i\}, \\ \text{and} \quad I_w &:= \{i \in \{1,\dots,n\} \mid w_i \leq v_i\}, \end{aligned}$$

and the states

$$x_1 \; := \; \frac{\sum_{i\in I_v} x_i \mathbf{e}_i}{\sum_{i\in I_v} x_i v_i} \quad \text{and} \quad x_2 \; := \; \frac{\sum_{i\in I_w} x_i \mathbf{e}_i}{\sum_{i\in I_w} x_i w_i}.$$

Note that, by definition, $g_v(x_1) = 1$ and $g_w(x_2) = 1$. Consider the positive scalar $\lambda = \sum_{i\in I_v} x_i v_i$, we have that $1-\lambda = \sum_{i\in I_w} x_i w_i$, since

$$\sum_{i\in I_v} x_i v_i \; + \; \sum_{i\in I_w} x_i w_i \; = \; \sum_i x_i \min\{v_i, w_i\} \; = \; 1.$$

Now, computing, we have $\lambda x_1 + (1-\lambda)x_2 = x$, proving that $x \in \mathrm{conv}\{\mathbb{B}_{g_v} \cup \mathbb{B}_{g_w}\}$.

($\supset$): The inclusions $\mathbb{B}_{g_v} \subset \mathbb{B}_{g_{v\vee w}}$ and $\mathbb{B}_{g_w} \subset \mathbb{B}_{g_{v\vee w}}$ are trivial. Consider thus $y \in \mathbb{B}_{g_v}$, $z \in \mathbb{B}_{g_w}$ (i.e. $\sum_i v_i y_i \leq 1$ and $\sum_i w_i z_i \leq 1$) and any $\lambda \in [0,1]$. Computing

$$\begin{aligned} \sum_i \min\{v_i, w_i\}(\lambda y_i + (1-\lambda) z_i) \;&\leq\; \sum_i v_i \lambda y_i + w_i(1-\lambda) z_i), \\ &=\; \lambda \sum_i v_i y_i + (1-\lambda) \sum_i w_i z_i, \\ &\leq\; 1, \end{aligned}$$

concluding the proof. $\square$

In the next proposition, we provide the corresponding properties for dual copositive norms, and moreover we show how, thanks to the convex-duality theory summarized in Section 1.2, we can derive them from Proposition 2.32 for primal norms.

Proposition 2.33. *Given any $v, w \in \mathbb{R}^n_{>0}$, any $A \in \mathbb{R}^{n\times n}_{\geq 0}$, any $\lambda > 0$ we have*

(1^d) $g^\star_{v+w} \;=\; g^\star_v \sharp g^\star_w$ *(inverse summation of dual norms is a dual norm) ;*

(2^d) $g^\star_{\lambda v} \;=\; \frac{1}{\lambda} g^\star_v$;

(3^d) $g^\star_{v\vee w} \;=\; \max\{g^\star_v, g^\star_w\}$ *(max of dual norms is a dual norm);*

(4^d) $\Big(\forall x \in \mathbb{R}^n_{\geq 0},\;\; g^\star_v(Ax) \leq g^\star_w(x)\Big) \;\;\Leftrightarrow\;\; Aw \leq_c v.$

Proof. Combining Proposition 2.32 and Lemma 1.27, we can directly prove Items (1^d), (2^d) and (3^d). For Item (4^d), we combine Proposition 2.32 and Lemma 1.28, which concludes the proof. □

In estimating the JSR of a set of non-negative matrices $\mathcal{A} = \{A_1, \ldots A_M\} \subset \mathbb{R}^{n\times n}_{\geq 0}$, we define the JSR approximations using both linear copositive templates $\mathcal{P}$ and $\mathcal{D}$, already introduced in [PJB10].

Definition 2.34 (Primal/dual linear copositive JSR approximation)**.** Given a finite set of matrices $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n\times n}$, the (common) *primal linear copositive approximation of the JSR*, denoted by $\rho_{\mathcal{P}}(\mathcal{A})$ is defined as

$$\begin{aligned} \rho_{\mathcal{P}}(\mathcal{A}) \;:=\; & \inf_{v\in\mathbb{R}^n_{>0}, \gamma\geq 0} \gamma \\ & A_i^\top v - \gamma v \leq 0, \quad \forall i \in \langle M\rangle. \end{aligned} \tag{2.12}$$

Similarly, the (common) *dual linear copositive approximation of the JSR*, denoted by $\rho_{\mathcal{D}}(\mathcal{A})$, is defined as

$$\begin{aligned} \rho_{\mathcal{D}}(\mathcal{A}) \;:=\; & \inf_{v\in\mathbb{R}^n_{>0}, \gamma\geq 0} \gamma \\ & A_i v - \gamma v \leq 0, \quad \forall i \in \langle M\rangle. \end{aligned} \tag{2.13}$$

Intuitively, $\rho_{\mathcal{P}}(\mathcal{A})$ represents the best estimate of the joint spectral radius of $\mathcal{A}$ one can obtain considering *common copositive primal Lyapunov norms*, recall Item (4) of Proposition 2.32. Again, $\rho_{\mathcal{D}}(\mathcal{A})$ represents the best bound on the JSR of $\mathcal{A}$ considering *common dual copositive Lyapunov norms*, recall Item (4^d) of Proposition 2.33.

We provide approximation guarantees on the JSR approximations provided by the linear copositive templates.

Theorem 2.35 ([PJB10])**.** *Consider $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n\times n}_{\geq 0}$. We have*

1. $\frac{1}{n}\rho_{\mathcal{P}}(\mathcal{A}) \leq \rho(\mathcal{A}) \leq \rho_{\mathcal{P}}(\mathcal{A})$,

2. $\frac{1}{n}\rho_{\mathcal{D}}(\mathcal{A}) \leq \rho(\mathcal{A}) \leq \rho_{\mathcal{D}}(\mathcal{A})$,
3. $\rho_{\mathcal{P}}(\mathcal{A}) = \rho_{\mathcal{D}}(\mathcal{A}^\top)$.

Proof. For Item *1.* see [PJB10, Theorem 2.6]. Item *3.* follows by definition, and finally Item *2.* follows by Items *1.* and *3.* recalling that transposition of matrices does not modify the JSR, i.e. $\rho(\mathcal{A}) = \rho(\mathcal{A}^\top)$. □

Finally, the following lemma is a particular case of Lemma 1.29 for the template of primal linear copositive norms.

Corollary 2.36. *If $v_1, w_1, v_2, w_2 \in \mathbb{R}^n_{>0}$ and $A \in \mathbb{R}^{n\times n}_{\geq 0}$. Then*

$$\begin{gathered}(\forall\, x \in \mathbb{R}^n_{\geq 0},\ \ \min\{g_{v_2}(Ax), g_{w_2}(Ax)\} \leq \min\{g_{v_1}(x), g_{w_1}(x)\}) \\ \Downarrow \\ (\,\forall\, x \in \mathbb{R}^n_{\geq 0},\ \ g_{v_2\vee w_2}(Ax) \leq g_{v_1\vee w_1}(x)\,).\end{gathered}$$

Proof. We recall that level sets of a pointwise-minimum function $f := \min\{f_a, f_b\}$ are union of levels sets of f_a and f_b. We thus need to prove that $x \in \mathbb{B}_{g_{v_1}} \cup \mathbb{B}_{g_{w_1}} \Rightarrow Ax \in \mathbb{B}_{g_{v_2}} \cup \mathbb{B}_{g_{w_2}}$ implies $x \in \mathbb{B}_{g_{v_1\vee w_1}} \Rightarrow Ax \in \mathbb{B}_{g_{v_2\vee w_2}}$.

Consider $x \in \mathbb{B}_{g_{v_1\vee w_1}}$, recalling proof of Item ((3)) in Proposition 2.32, we know that $x \in \operatorname{conv}\{\mathbb{B}_{g_{v_1}} \cup \mathbb{B}_{g_{w_1}}\}$ i.e. there exist $x_1, x_2 \in \mathbb{B}_{g_{v_1}} \cup \mathbb{B}_{g_{w_1}}$, and $\lambda \in [0,1]$ such that $x = \lambda x_1 + (1-\lambda)x_2$. Computing

$$Ax \;=\; A\lambda x_1 + A(1-\lambda)x_2 \;=\; \lambda Ax_1 + (1-\lambda)Ax_2.$$

Since $Ax_1, Ax_2 \in \mathbb{B}_{g_{v_2}} \cup \mathbb{B}_{g_{w_2}}$, we have proved that $Ax \in \operatorname{conv}(\mathbb{B}_{g_{v_2}} \cup \mathbb{B}_{g_{w_2}}) = \mathbb{B}_{g_{v_2\vee w_2}}$, concluding the proof. □

This result will be used to prove Theorem 7.43.

PART II
Neural Lyapunov functions

3

Approximation of the JSR using polytopic norms

INSPIRED by the theoretical guarantees on the JSR approximation in Theorems 2.21 and 2.29 using respectively quadratic norms and SOS polynomials, we aim at exploiting Proposition 2.5 for the particular case of *polytopic norms*. This template has already been widely used to compute Lyapunov functions, such as in [AJ19, WJ20, BS23] for instance. In particular, the *polytopic* template has been investigated to approximate the joint spectral radius in [GWZ05, GZ08, JCG14].

The use of this template is motivated and justified by its flexibility since it is well known that any convex body can be approximated arbitrarily closely by a polytope. Unfortunately, this flexibility comes at the price of a greater complexity than the templates we have discussed in Section 2.2, as summarized in Sections 3.1 and 3.2 . In particular, the representation complexity of polyhedra and polytopes does not only depend on the dimension, as it is the case for quadratic norms, but it depends on the number of faces/vertices. Moreover, finding a (common) polytopic Lyapunov function involves solving a bilinear program, which is generally hard to solve computationally.

In this chapter, we derive approximation guarantees similar to Theorems 2.21, 2.29 and 2.35 for the template of polytopic norms as a function of the number of faces/vertices of the corresponding polytope. To the best of our knowledge, such approximation guarantees have not been derived so far for the specific template of polytopic norms.

3.1 Polytopes and continuous piecewise linear functions

In this section, we introduce and summarize classical results on polyhedral and polytopic subsets, and their dual hyperplane and vertex representations. This section is mainly based on [BM07, Chapters 3 and 4] and [Roc70, Section 19].

Definition 3.1 (H-representation of a polyhedral set)**.** A convex polyhedral set $\mathcal{S} \subseteq \mathbb{R}^n$ is defined as the intersection of finitely many halfspaces, i.e.

$$\begin{aligned} \mathcal{S} := P(F,g) &= \{x \in \mathbb{R}^n \mid Fx \leq_c g\}, \\ &= \bigcap_{i=1}^{s} \{x \in \mathbb{R}^n \mid F_i x \leq g_i\}, \end{aligned} \tag{3.1}$$

where $g \in \mathbb{R}^s$ and F_i denotes the i-th row of the matrix $F \in \mathbb{R}^{s \times n}$. Equation (3.1) is referred as the *hyperplane* or *H-representation* of a polyhedral set.

A polyhedral set contains the origin if and only if $g \geq_c 0$, and contains the origin in its interior if and only if $g >_c 0$. In this case, we can assume without loss of generality that $g = \mathbf{1}$, where $\mathbf{1}$ denotes the vector with all components equal to 1. In this case, the polytope $P(F,\mathbf{1})$ is denoted by $P(F)$. Moreover, a *symmetric* polyhedral set $\overline{P}(F,g)$ can be represented in the form

$$\overline{P}(F,g) := \{x \in \mathbb{R}^n \mid |Fx| \leq g\}. \tag{3.2}$$

Similarly, a symmetric polyhedral set $\overline{P}(F,\mathbf{1})$ is denoted by $\overline{P}(F)$.

In Definition 3.1, a polyhedral set is defined by its *H-representation*, i.e. as a intersection of a finite collection of halfspaces. Equivalently, one can define the dual *V-representation* of a polyhedral set.

Definition 3.2 (V-representation of a polyhedral set)**.** A convex polyhedral set $\mathcal{S} \subseteq \mathbb{R}^n$ admits a *vertex* or *V-representation* of the form

$$\mathcal{S} := V(X_w, X_y) = \left\{ x = X_w w + X_y y \;\middle|\; \sum_{i=1}^{q} w_i = 1,\; w, y \geq_c 0 \right\}, \tag{3.3}$$

where the columns of the matrix $X_w \in \mathbb{R}^{n \times q}$ represent the set of finite *vertices* while those of the matrix $X_y \in \mathbb{R}^{n \times p}$ represent the set of *infinite directions* or *rays*.

In the symmetric case, the following representation holds

$$\overline{V}(X_w, X_y) \;:=\; \left\{ x = X_w w + X_y y \mid \sum_{i=1}^{p} |w_i| = 1,\; y \text{ arbitrary} \right\},$$

where the vertices are given by the columns of X_w and their opposite, as for the infinite directions and X_y.

Polyhedral sets are closed and convex by definition, but they might not be bounded. This observation motivates the introduction of *polytopes*.

Definition 3.3 (Polytope)**.** A *polytope* is the convex hull of a finite set of points.

Thanks to Weyl–Minkowski's theorem, there is an equivalence between polytope and bounded polyhedron.

Theorem 3.4 (Weyl–Minkowski's theorem)**.** *Every polytope is a polyhedron. Every bounded polyhedron is a polytope.*

In particular, a necessary and sufficient condition for the V-representation in Equation (3.3) to represent a polytope is that $X_y = 0$. Similarly, the H-representation in Equation (3.2) represents a polytope if and only if the matrix F has full column rank. Without loss of generality, the vertex representation of a polytope which contains the origin in its interior can be written in the form

$$V(X, c) \;:=\; \left\{ x = Xw \mid c^\top w \leq 1,\; w \geq_c 0 \right\}$$

where $X \in \mathbb{R}^{n \times q}$, $q \geq n+1$, $\operatorname{rank}(X) = n$ and $c >_c 0$. Polytopes of the form $V(X, \mathbf{1})$ will be denoted by $V(X)$, while symmetric polytopes are denoted by $\overline{V}(X, c)$ and $\overline{V}(X)$.

The class of polyhedral sets is closed under basic operations already considered in Section 1.2.

Proposition 3.5. *If A and B are (symmetric) polyhedra, $\lambda \geq 0$, and f is an linear map then:*

1. *The image $f(A)$ is a (symmetric) polyhedron;*
2. *The scaled set λA is a (symmetric) polyhedron;*
3. *$A \cap B$ is a (symmetric) polyhedron;*

4. *The sum $A + B$ is a (symmetric) polyhedron;*
5. *The convex hull of the union, i.e. $conv(A \cup B)$, is a (symmetric) polyhedron.*

The same properties hold for polytopes.

Proof. Items *1.* and *2.* directly follow from the H and V representations of polyhedral sets.

3. This property directly follows from the H-representation of polyhedral sets, since the intersection of two polyhedra is the set of points in $\mathbb{R}^n$ which satisfy all inequalities associated to both polyhedra. Indeed, if $A = P(F^A, g^A)$ and $B = P(F^B, g^B)$, then

$$A \cap B \; = \; P\left(\begin{bmatrix} F^A \\ F^B \end{bmatrix}, \begin{bmatrix} g^A \\ g^B \end{bmatrix} \right).$$

4. For brevity, we assume that $A := V(X^A)$ and $B = V(X^B)$ are polytopes. In this case, the sum is the polytope defined as

$$A + B \; := \; V(X^{AB})$$

where X^{AB} is obtained by adding a vertex of A and a vertex of B in pairs in all possible ways. In particular, a vector $x \in A + B$ if it is of the form

$$x \; := \; X^A w^A + X^B w^B \; = \; \sum_i X_i^A w_i^A + \sum_j X_j^B w_j^B,$$

where

$$\sum_i w_i^A = 1, \; \sum_j w_j^B = 1, \; \text{and} \; w_i^A, w_j^B \geq 0.$$

Then,

$$\begin{aligned} x \; &= \; \sum_{ij} X_i^A w_i^A w_j^B + \sum_{ij} X_j^B w_i^A w_j^B, \\ &= \; \sum_{ij} w_i^A w_j^B (X_i^A + X_j^B), \\ &:= \; X^{AB} w^{AB}, \end{aligned}$$

where the vectors $w^A B$ are of the form $w_i^A + w_j^B \geq_c 0$, and $\sum_k w_k^{AB} = 1$. Therefore $A + B \subseteq V(X^{AB})$, the convex hull of all the points $X_i^A + X_j^B$. On the other hand, all these points belong to the sum since both X_i^A and X_j^B do, then $\text{conv}\{A + B\} = V(X^{AB})$. Since $A + B$ is a convex set, $A + B = \text{conv}\{A + B\}$ which ends the proof.

5. Here again, only the case of bounded polyhedra is considered for brevity. The convex hull of the union is the convex hull of all the vertices of A and B. Therefore,

$$\mathrm{conv}\{A \cup B\} \;=\; V\left(\left[X^A \; X^B\right]\right).$$

□

We claimed that there is a duality between H and V-representations of polytopes. The following proposition formally explains their dual relation.

Proposition 3.6. *Consider a polytope $P = P(F) = V(X)$ including the origin as an interior point and its polar set P°. Then the following properties hold:*

$$P(F)^\circ = V(F^\top) \quad and \quad \overline{V}(X)^\circ = \overline{P}(X^\top).$$

Proposition 3.6 implies in particular that, for any polytope which includes the origin in its interior with n_p planes and n_v vertices, the polar P° has exactly $n_p^\circ = n_v$ planes and $n_v^\circ = n_p$ vertices. However, passing from a representation to the other, namely determining the vertices from the planes and vice-versa, cannot be done easily, especially in high dimension. Moreover, the algorithms which involve polyhedra computations are usually very demanding in terms of computational complexity. Therefore, it is often recommended to work with *minimal representations* to keep the complexity as low as possible.

Definition 3.7 (Minimal representation)**.** A plane or vertex representation is *minimal* if and only if there is no other representation of the same set involving a smaller (with respect to dimensions) F or X.

Minimal representation of a set can be achieved by removing all the "redundant" planes (vertices) from the plane (vertex) representation, which requires the resolution of a linear programming problem. Therefore, computing a minimal representation also turns out to be a computational demanding task.

Therefore, we define the *H-complexity index of a polyhedral set* as the number of rows of the matrix F in the H-representation in Equation (3.1), or similarly the *V-complexity index* as the number of vertices in the V-representation in Equation (3.3). By duality, none of these representations can be considered as more convenient than the other in general. When the two complexity indices can be used interchangeably, we speak of the *complexity* of the polytope.

In what follows, we consider polytopes which contains the origin in their interior, of the form

$$\mathcal{S} \;:=\; \{x \in \mathbb{R}^n \mid Gx \leq w\},$$

with $G \in \mathbb{R}^{m\times n}$, $m \geq n+1$, $rank(G) = n$ and $w > 0$. The dual equivalent representation of $\mathcal{S}$ is

$$\mathcal{S} \;:=\; \{Vy \mid c^\top y \leq 1, y \geq_c 0\},$$

where $V \in \mathbb{R}^{n\times q}$, $q \geq n+1$, $rank(V) = n$ and $c > 0$. Using the Minkowksi/gauge function in Definition 1.20, these polytopic sets can be written as the 1-sublevel set of the function

$$V(x) \;:=\; \max_{i=1,\ldots,m} \left\{ \frac{(Gx)_i}{w_i} \right\}, \tag{3.4}$$

or, equivalently,

$$V(x) \;:=\; \min_{y\geq 0} \left\{ c^\top y \mid x = Vy \right\}. \tag{3.5}$$

These functions are called *convex continuous piecewise linear functions*.

Definition 3.8 (Convex continuous piecewise linear function)**.** Given a polytope $\mathcal{S}$ which contains the origin in its interior, a *convex continuous piecewise linear* function is defined as the Minkowski function of the form of Equation (3.4) or equivalently Equation (3.5). The *number of pieces* of a convex continuous piecewise linear functions is defined as the H-complexity of the underlying polytope.

In practice, these functions admit an underlying polyhedral covering of the state space and the functions differ over each region while guaranteeing the continuity on the boundary. We can therefore define *continuous piecewise linear functions* by relaxing the convexity property.

Definition 3.9 (Continuous piecewise linear function)**.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is *continuous piecewise linear* (CPWL for short) if there exists a finite set of $d \in \mathbb{N}$ polyhedra whose union is $\mathbb{R}^n$, and f is linear over each polyhedron. The *number of pieces* of f is the smallest $d \in \mathbb{N}$ for which such a covering exists.

A continuous piecewise linear function can be proved to be written as the difference of two convex continuous piecewise linear functions, as stated below.

Proposition 3.10 (Proposition 4.3. in [HBDSS21]). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a CPWL function with p linear pieces. Then, f can be written as $f = g - h$ where both g and h are convex CPWL functions with at most p^{2n+1} pieces.*

Note that the notion of CPWL function in Definition 3.9 can be generalized to the notion of continuous piecewise *affine* (CPWA for short) functions for which the function is affine over each cell of the polyhedral covering. Moreover, Proposition 3.10 holds for CPWA functions and convex CPWA functions respectively.

3.2 Computation technique

In this section, we explain how to compute numerically a (common) polytopic Lyapunov norm by translating Lyapunov inequalities in bilinear matrix inequalities. Due to their computational complexity, we derive similar results for the particular case of symmetric polyhedra, which only require the resolution of a linear program. This section summarizes the results in [BM07, Chapters 3 and 4] and [AJ19].

First, the following result provides an interesting condition to check the inclusion between two polyhedra.

Lemma 3.11. *The inclusion*

$$P(F^{(1)}, g^{(1)}) \subseteq P(F^{(2)}, g^{(2)}) \tag{3.6}$$

holds if and only if there exists a nonnegative matrix H such that

$$\begin{array}{rcl} HF^{(1)} & = & F^{(2)}, \\ Hg^{(1)} & \leq & g^{(2)}. \end{array} \tag{3.7}$$

Proof. Let us assume first that the inclusion in Equation (3.6) holds. In this case, if we denote $F_i^{(k)}$ the i-th row of matrix $F^{(k)}$, the solution of the following linear program

$$\begin{array}{rcll} \mu_i & := & \max\limits_{x} & F_i^{(2)} x \\ & & \text{s.t.} & F^{(1)} x \leq g^{(1)} \end{array}$$

satisfies that $\mu_i \leq g_i^{(2)}$. We consider the dual of this program, i.e.

$$\begin{array}{rll} \mu_i := & \min\limits_{h} & hg^{(1)} \\ & \text{s.t.} & hF^{(1)} = F_i^{(2)}, \\ & & h \geq 0 \end{array}$$

and we denote $h^{(i)}$ the nonnegative row vector which is solution of the dual. Then, the square matrix H whose i-th row is $h^{(i)}$ satisfies the conditions in Equation (3.7).

Conversely, let us assume that a nonnegative matrix H satisfies the conditions in Equation (3.7). For all $x \in \mathbb{R}^n$ such that $F^{(1)}x \leq g^{(1)}$,

$$\begin{array}{rcl} F^{(2)}x & = & HF^{(1)}x, \\ & \leq & Hg^{(1)}, \\ & \leq & g^{(2)}, \end{array}$$

which ends the proof. □

Lemma 3.11 admits "a dual version" which involves the vertex representation of polytopes.

Corollary 3.12. *The inclusion*

$$V(X^{(1)}, c^{(1)}) \subseteq V(X^{(2)}, c^{(2)})$$

holds if and only if there exists a nonnegative matrix P such that

$$\begin{array}{rcl} X^{(2)}P & = & X^{(1)}, \\ {c^{(2)}}^\top P & \leq & {c^{(1)}}^\top. \end{array}$$

The two previous results can be used to characterize Lyapunov inequalities when the Lyapunov function is defined as the Minkowski function of a polytope. This leads to the following theorem.

Theorem 3.13. *Consider two polytopes which admit the origin in their interior, denoted by $\mathcal{S}_d, \mathcal{S}_s \subset \mathbb{R}^n$ with matrices $G_d \in \mathbb{R}^{m_d \times n}$ and $G_s \in \mathbb{R}^{m_s \times n}$, and $X_d \in \mathbb{R}^{n \times q_d}$ and $X_s \in \mathbb{R}^{n \times q_s}$ and vectors w_d, w_s, c_d, c_s. Consider $\lambda > 0$, a matrix set $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n \times n}$, and $i \in \langle M \rangle$. The following are equivalent:*

(i) $V_d(A_i x) \leq \lambda V_s(x)$;

(ii) *There exists a nonnegative matrix $H \in \mathbb{R}^{m_d \times m_s}$, such that $G_d A_\sigma = HG_s$ and $Hw_s \leq \lambda w_d$;*

(iii) There exists a nonnegative matrix $P \in \mathbb{R}^{q_d \times q_s}$ *such that* $A_\sigma X_s = X_d P$ *and* ${c_d}^\top P \leq \lambda {c_s}^\top$.

Unfortunately, the matrix inequalities involved in Theorem 3.13 are difficult to solve numerically since bilinear matrix equations appear in both H and V representations. However, it turns out that these inequalities are simplified when symmetric polytopes are considered.

Corollary 3.14. *Consider two symmetric polytopes which admit the origin in their interior, denoted by* $\mathcal{S}_d, \mathcal{S}_s \subset \mathbb{R}^n$ *with matrices* $G_d \in \mathbb{R}^{n\times n}$ *and* $G_s \in \mathbb{R}^{n\times n}$*, and* $X_d \in \mathbb{R}^{n\times n}$ *and* $X_s \in \mathbb{R}^{n\times n}$ *and vectors* w_d*,* w_s*,* c_d*,* c_s*. Consider* $\lambda > 0$*, a matrix set* $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n\times n}$*, and* $i \in \langle M \rangle$*. The following are equivalent:*

(i) $V_d(A_i x) \leq \lambda V_s(x)$*;*

(ii) $|G_d A_i {G_s}^{-1}| \leq \lambda w_d$*;*

(iii) ${c_d}^\top |{X_d}^{-1} A_i X_s| \leq \lambda {c_s}^\top$.

In this case, the conditions in Corollary 3.14 can be verified by solving a linear program for a given choice of the matrices G and V.

3.3 Approximation guarantees on the JSR

In the previous sections, we have emphasized that the representation complexity of polyhedral sets, contrary to ellipsoidal sets, does not only depend on the space dimension, but may be arbitrarily high. The counterpart of the computational burden of polyhedral sets, summarized in Section 3.1, is their flexibility. Indeed, any convex and compact set can be approximated arbitrarily closely by a polyhedron, see [Lay44]. In particular, if S is a convex compact set with the origin in its interior, then for all $0 < \varepsilon < 1$ there exists a polytope P such that

$$(1 - \varepsilon)\, P \ \subseteq \ C \ \subseteq \ P.$$

In this section, we aim to provide approximation guarantees on the polytopic approximation of the JSR as a function of the complexity of the polytope, i.e. the number of planes/vertices.

First, let us define formally the polytopic approximation of the joint spectral radius of H-complexity (resp. V-complexity) d.

Definition 3.15 (Polytopic approximation of the JSR)**.** Given a finite set of matrices $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n\times n}$, the (common) *polytopic approximation of the JSR of H-complexity* d, denoted by $\rho_{POL_H,d}(\mathcal{A})$ is defined by

$$\rho_{POL_H,d}(\mathcal{A}) := \inf_{P\in POL_H(n,d)} \max_{A\in\mathcal{A}} \|A\|_P, \tag{3.8}$$

where $POL_H(n,d)$ refers to the set of polytopes $P \subseteq \mathbb{R}^n$ of H-complexity d at most, and $\|\cdot\|_P$ denotes the corresponding polytopic norm in Equation (3.4). Similarly, $\rho_{POL_V,d}(\mathcal{A})$ denotes the (common) polytopic approximation of the JSR of V-complexity d.

Similarly to Lemma 2.20 for the quadratic and SOS template respectively, the following result (Theorem 1.1 in [Bar13]) discusses the approximation power of a convex set by a polytope as a function of the dimension and the number of vertices of the polytope. Using duality, we can prove the similar result to get a bound on the required number of faces for the approximating polytope, rather than the vertices.

Lemma 3.16. *Let n and k be two positive integers and $\tau > 1$ be a real number such that*

$$\left(\tau - \sqrt{\tau^2-1}\right)^k + \left(\tau + \sqrt{\tau^2-1}\right)^k \geq 6\, D(n,k)^{1/2}, \tag{3.9}$$

where

$$D(n,k) := \sum_{m=0}^{\lfloor k/2 \rfloor} \binom{n+k-1-2m}{k-2m}. \tag{3.10}$$

Then, for any symmetric, compact and convex set $C \subseteq \mathbb{R}^n$ with non-empty interior (and therefore containing the origin), there exists a symmetric polytope $P \subseteq \mathbb{R}^n$ with complexity at most $8D(n,k)$ such that

$$P \subset C \subset \tau P. \tag{3.11}$$

Proof. The theorem is initially stated in [Bar13, Theorem 1.1] with a bound on the number of *vertices* of the polytope. We derive below the same result with a bound on the number of *facets* using duality, which ends the proof.

Consider a symmetric, compact and convex set K with non empty interior in dimension $n \in \mathbb{N}$, a precision $\tau > 1$ and an integer $k \in \mathbb{N}$ which satisfies Equation (3.9). The polar of C, denoted by C°, retains its original properties. By Lemma 3.16, there exists $\widetilde{P} \subset \mathbb{R}^n$ with at most $8D(n,k)$ vertices such that

$$\widetilde{P} \subseteq C^\circ \subseteq \tau\widetilde{P}.$$

By Lemma 1.25, we have

$$\begin{aligned} & (\tau\widetilde{P})^\circ && \subseteq && C^{\circ\circ} && \subseteq && \widetilde{P}^\circ, \\ \Leftrightarrow \quad & \frac{1}{\tau}\widetilde{P}^\circ && \subseteq && C && \subseteq && \widetilde{P}^\circ, \end{aligned}$$

where $\widetilde{P}^\circ$ has as many facets as $\widetilde{P}$ has vertices, recalling Proposition 3.6. Posing $P := \frac{1}{\tau}\widetilde{P}^\circ$ ends the proof. □

In [Bar13, Corollary 1.2], the authors provide a corollary of Lemma 3.16 where they take τ arbitrarily close to 1; This result is recalled below.

Corollary 3.17. *For any $n \in \mathbb{N}$ any symmetric, compact and convex set $C \subseteq \mathbb{R}^n$ with non-empty interior, there exists a symmetric polytope $P \subseteq \mathbb{R}^n$ with at most $\gamma(n)\varepsilon^{-(n-1)/2}$ vertices (resp. faces) which approximates C within a factor of $1+\varepsilon$, i.e.*

$$(1+\varepsilon)\,P \subseteq C \subseteq P,$$

where $\gamma(n)$ is of the order $n^{n/4}$.

Using Lemma 3.16, we can now derive theoretical approximation guarantees on the polytopic approximation of the JSR, as a function of the complexity of the polytope.

Theorem 3.18. *Let $\rho(\mathcal{A})$ be the joint spectral radius of a finite set of matrices $\mathcal{A}$ of dimension $n \in \mathbb{N}$. For any $\tau > 1$ and $k_\tau \in \mathbb{N}$ such that relation (3.9) is satisfied, the following relation holds:*

$$\frac{1}{\tau}\rho_{POL_H,d}(\mathcal{A}) \leq \rho(\mathcal{A}) \leq \rho_{POL_H,d}(\mathcal{A}). \tag{3.12}$$

where $d := 8D(n,k_\tau)$ as defined in Equation (3.10). The same result holds for $\rho_{POL_V,d}(\mathcal{A})$.

Proof. Consider a finite set of m matrices $\mathcal{A} := \{A_1,\ldots,A_M\} \subset \mathbb{R}^{n\times n}$ of dimension $n \in \mathbb{N}$, and $\rho(\mathcal{A})$ its joint spectral radius. By Proposition 2.8, for any value $\varepsilon > 0$, there exists an ε-norm $\|\cdot\|_\varepsilon$ such that $\forall i = 1,\ldots,m$

$$\forall x \in \mathbb{R}^n,\ \|A_i x\|_\varepsilon \leq (\rho(\mathcal{A})+\varepsilon)\,\|x\|_\varepsilon\,.$$

This norm defines a convex set $K \subset \mathbb{R}^n$ (which contains the origin) that can be approximated by a polytope. By Lemma 3.16, for any positive integer d and any real number $\tau > 1$ satisfying Equation (3.9), there exists a symmetric polytope $P \subset \mathbb{R}^n$ with at most $8D(n,k)$ vertices such that

$$P \subseteq K \subseteq \tau P.$$

Therefore, the norm $\|\cdot\|_{\tau P}$ whose 1-level set is τP, satisfies that for all $x \in \mathbb{R}^n$,

$$\|x\|_{\tau P} \leq \|x\|_{\varepsilon} \leq \tau \|x\|_{\tau P},$$

and then for any $i = 1, \ldots, M$ and any $x \in \mathbb{R}^n$:

$$\begin{aligned} \|A_i x\|_{\tau P} &\leq \|A_i x\|_{\varepsilon}, \\ &\leq (\rho(\mathcal{A}) + \varepsilon) \|x\|_{\varepsilon}, \\ &\leq (\rho(\mathcal{A}) + \varepsilon) \tau \|x\|_{\tau P}. \end{aligned}$$

Then, $\|A_i\|_{\tau P} \leq (\rho(\mathcal{A}) + \varepsilon)\tau$ for all $i = 1, \ldots, m$ and for any $\varepsilon > 0$. Therefore, at worst, we have that

$$\frac{1}{\tau} \rho_{POL,8D(n,k_\tau)}(\mathcal{A}) \leq \rho(\mathcal{A}) \leq \rho_{POL,8D(n,k_\tau)}(\mathcal{A}),$$

where the second inequality follows directly from Proposition 2.5. □

Therefore, by merging Theorem 3.18 and Corollary 3.17, this means that we can approximate as close as possible the JSR using polytopic norms with increasing number of vertices.

Using this theorem, we can compute an upper bound on the number of *program variables* required to achieve a given precision τ as the number of vertices multiplied by the dimension of the state space. In Figure 3.1, we show the evolution, as a function of the dimension, of this number of variables, for different values of τ. For the sake of comparing with the performance of the SOS approximations of degree 3 and 4, we selected $\tau = \tau_{SOS}(n,d)$ in Equation (2.7) for $d = 3$ and $d = 4$, i.e. the approximation guarantee using SOS polynomials of degree 6 and 8 recalled in Theorem 2.29. These results show that for small dimensions, the polytopic approach requires slightly more variables than the SDP approach. Conversely, when we consider higher dimensions, the trend is reversed. These observations motivate the polytopic approach rather than the poly-

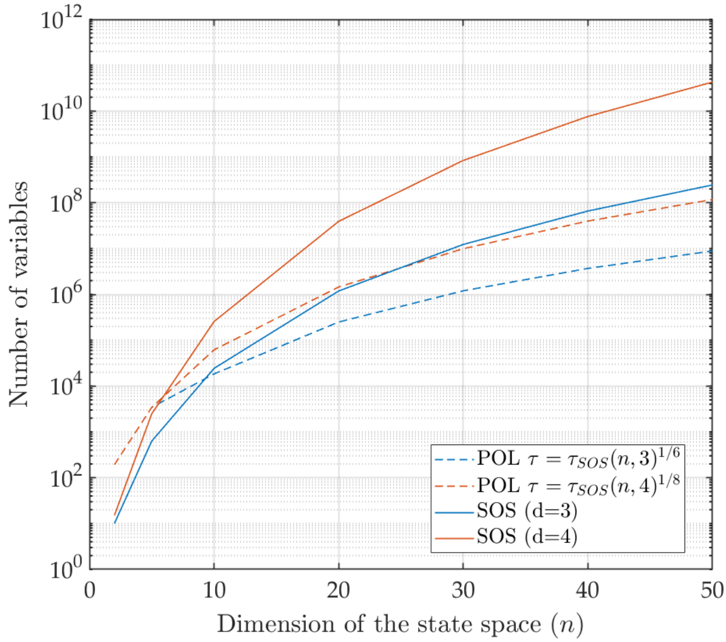


Fig. 3.1 Evolution, as a function of the dimension (n), of the number of variables required to achieve precision values τ on the JSR approximation with the polytopic approach (dashed line) and with the SOS approach (continuous line). In blue, $\tau = \tau_{SOS}(n, 3)$, in red, $\tau = \tau_{SOS}(n, 4)$. These values are the minimal required accuracy to outperform the guarantee in Equation (2.7) for, respectively, degree-3 and -4 SOS polynomials. One can see that for large values of n, the polytopic approximation needs far fewer variables, for any given precision.

nomial approach since we need less variables in high dimension. However, computation methods rapidly suffer from the curse of dimensionality: the complexity is polynomial for SDPs while it is exponential for the polytopic norms, which necessitates solving a bilinear program as previously explained. For this reason, we will investigate in the following Chapters 4 and 5 generating polytopic approximations provided by ReLU neural networks.

3.4 Summary

This section introduces a new template for candidate Lyapunov functions, namely the polytopic norms.

Summary of Chapter 3

This chapter tackles the design of Lyapunov functions and in particular the approximation of the joint spectral radius using polytopic norms.

Section 3.1: Polytopes and continuous piecewise linear functions
In this section, we provide a summary of polyhedral and polytopic sets, their closure properties and the polar duality between their H and V-representations. Based on this, we define the complexity index of a polytope as the number of its faces/vertices.

Section 3.2: Computation technique
We show in this section that finding a common polytopic norm amounts to the resolution of a bilinear program which is usually computationally expensive. This cost is greatly reduced when symmetrical polytopes are considered.

Section 3.3: Approximation guarantees on the JSR
We tackle the approximation of the JSR using polytopic norms. In this setting, we have provided guarantees on this approximation, which demonstrate that it is possible to approximate as closely as possible the JSR by increasing the number of vertices of the polytope. These guarantees are competitive with classical SDP-based Lyapunov approaches in terms of number of decision variables.

Although this template is highly flexible and therefore very powerful, its use is generally limited by its computational cost. The following two chapters try to overcome this drawback, and focus on the synthesis of Lyapunov functions using neural networks. In particular, polytopic Lyapunov functions can be represented by ReLU-activated neural networks. Moreover, Theorem 3.18 will prove useful to derive a bound on the network's size (width and detph) to achieve a given precision on the joint spectral radius approximation.

4

Neural networks and their representation power

FOR several decades now, it has been known that *neural networks* provide a powerful framework for achieving a wide range of different tasks, such as classification [Zha00], facial recognition [GZ19], language processing [OMK21], etc. Therefore, their empirical effectiveness made them popular. However, although they have proved their worth in practice, our deep understanding of neural networks remains limited, and some theoretical explanation is still missing. These applications stem from approximating [Cyb89, Hor91, AB99, LS17] an unknown function from data observations. Then, their success on challenging tasks must rely on their ability to produce complex functions. The representation power of neural networks [LPW+17, ABMM18, HBDSS21, DK22], which is the subject of this chapter, emerges as a central element that helps us understand them better.

We begin with a formal introduction of *feedforward neural networks*. Then, we consider two sorts of activation functions. We first introduce the *polynomial activation functions* which allow us to represent polynomials. Secondly, we review recent results about the representation power of *ReLU neural networks* which encode *continuous piecewise linear functions*. In particular, we investigate the set of functions which are *representable* by a neural network with respect to their structure, i.e. the number of layers and neurons in each of them.

4.1 Introduction to neural networks

A *neural network* involves several components, that are *layers* of *neurons*, *weights*, *biases*, *activation function* and a *learning rule*. In particular, a neuron, that is the fundamental unit of a neural network, receives input which are multiplied by the adjacent weights, adds them up and this sum is finally passed through the activation function.

Definition 4.1 (Feedforward neural network)**.** A $(L+1)$*-layer feedforward neural network* with $L \in \mathbb{N}$ hidden layers is defined by L affine transformations $T^{(j)} : \mathbb{R}^{n_{j-1}} \to \mathbb{R}^{n_j}$, $x \mapsto W^{(j)}x + b^{(j)}$ for $j \in \langle L \rangle$, and a linear transformation $T^{(L+1)} : \mathbb{R}^{n_L} \to \mathbb{R}^{n_{L+1}}$, $x \mapsto W^{(L+1)}x$. The network *represents* the function $NN : \mathbb{R}^{n_0} \to \mathbb{R}^{n_{L+1}}$ given by

$$NN(\cdot) \; := \; T^{(L+1)} \circ \sigma \circ \cdots \circ T^{(2)} \circ \sigma \circ T^{(1)}(\cdot), \tag{4.1}$$

where

(a) the function $\sigma(\cdot)$ is called the *activation function* and is applied componentwise;

(b) the matrices $W^{(l)}$ and the vectors $b^{(l)}$ are respectively called the *weights* and the *biases* of the l-th layer ;

(c) n_l is the *width* of the l-th layer, and the maximum width of all the hidden layers is called the *width* of the neural network;

(d) the *depth* of the network is the number of hidden layers, that is $L+1$.

Figure 4.1 provides an illustration of a neural network. Note that we assume by definition that the last layer of a neural network is activation-free and without biases.

Inspired by [HBDSS21, Definition 2.2], we define the notion of *homogenized neural network*.

Definition 4.2. For a neural network given by affine transformations $T^{(l)}(x) = W^{(l)}x + b^{(l)}$, we define the *corresponding homogenized neural network* given by $\widetilde{T}^{(l)}(x) = W^{(l)}x$ with all biases set to zero.

This notion will prove useful later in this chapter, in particular for demonstrating that there is no loss of generality in considering bias-free neural networks to represent candidate Lyapunov functions.

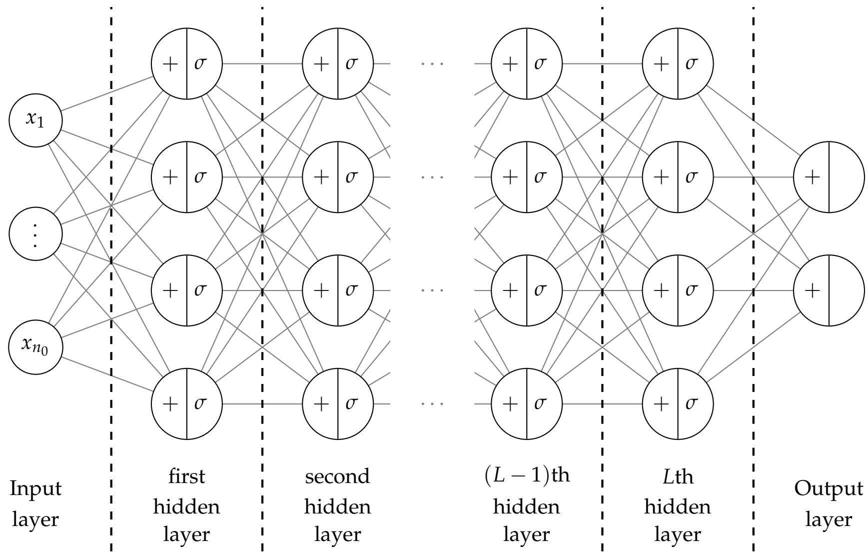


Fig. 4.1 Illustration of a $(L+1)$-layer neural network. The input layer admits 3 neurons, the output layer 2 neurons and all the other layers admit 4 neurons. The depth of this network is $L+1$ while the width is 4.

A neural network can be seen as a multivariate function which depends on its weights and biases. And therefore, determining these parameters amounts to determining the function represented by the network. Various methods can be used, but this is generally done iteratively, with an objective to reach; we usually speak of the *training* of the network. Although there are several training methods (see [SP24] for instance), the most classical of them, which we will use in this thesis, exploits gradient descent.

We first need to quantify how good this function is performing a given task with respect to a given set of data points. This is done by defining a *loss function* whose minimization is the objective of the *learning* or *training process*. In practice, the *gradient descent* is used to reduce this loss function by adjusting all the variables based on the corresponding partial derivative of the loss function. Two basic learning approaches can be identified from their loss function.

- In *supervised learning*, the training is guided by a supervisor which has access to the pairs input-output. The loss function usually compares the neural prediction with the correct answer and is interpreted as the error. Then, we iteratively update the parameters of the network and we stop the training when the precision threshold is reached.

- When the training is unsupervised, no supervisor is provided so that we do not know the correct output. Therefore the objective of the *unsupervised learning* is to identify patterns, underlying structure from the data without explicit guidance.

In a nutshell, the training involves three steps, namely the *forward propagation*, the *loss calculation* and the *backpropagation* using gradient descent. The *forward propagation* refers to the calculation and storage of intermediate variables (including outputs) for a neural network in order from the input layer to the output layer. Input data is fed into the network and passed through various layers: each neuron in these layers processes the input and iteratively passes it to the next layer, ultimately leading to the output layer. *Backpropagation*, on the other hand, is the learning phase. Once the forward propagation is complete and an output is produced, one can compute the loss function which is then used to adjust the network's weights and biases. This process is iterative and involves moving backward through the network to compute the gradient of neural network parameters. In short, the method traverses the network in reverse order, from the output to the input layer, according to the chain rule from calculus.

Note that several variants of gradient descent have been introduced in recent years, such as the *stochastic gradient descent* which computes a stochastic approximation of the gradient based on a randomly subset of the sample points. *Adam*, introduced in [KB15], and all its variants [LH17], are currently the most efficient algorithms.

4.2 Representation power of neural networks

The *approximation* capabilities of neural networks have been studied for years in the literature. Previous works focused on the network's architecture, meaning the ability for a neural network to approximate as close as desired a given function according to its depth and its width. It is well established [Cyb89, Hor91, AB99, LS17] that single layer networks can approximate arbitrarily well any continuous functions. But it does not provide insights into the class of functions which can be exactly represented.

In this section, we discuss the *representation power* of neural networks, as we aim to establish an *exact representation* of a function. In other words, we study the relation between the architecture of a network and its representation capabilities. We focus in particular on the representation of *Lyapunov* *functions*, i.e. continuous, positive definite and homogeneous functions by Definition 1.10. In this section, we consider two types of activation functions, namely the *polynomial* and *Rectified Linear Units* activation functions. Even though recent research has focused on the representation power of these networks, it remains a challenging question.

4.2.1 Polynomial activation function

We begin by considering the simplest case of polynomial activation functions, that is the *square activation function*.

Definition 4.3 (Square activation function)**.** The *square activation function* is the component-wise square function, i.e.

$$\sigma_2(x) := \left(x_1{}^2, \dots, x_n{}^2\right).$$

Then, a neural network with square activation functions, so-called a *square neural network*, is a consecutive composition of affine maps and component-wise square functions. Any function represented by such a network is therefore a polynomial whose degree d directly depends on the number of hidden layers L through the relation $d = 2^L$.

As previously mentioned, we want to study the capabilities of a square neural network to represent quadratic Lyapunov functions. Therefore the represented function must be a scalar function defined on $\mathbb{R}^n$. This means in practice that the input layer admits n neurons and the output layer 1 single neuron.

As first example, let us consider a square neural network with 1 layer of p neurons. In this case, there exists $W^{(1)} \in \mathbb{R}^{p\times n}$, $W^{(2)} \in \mathbb{R}^{1\times p}$ and $b^{(1)} \in \mathbb{R}^p$ such that

$$NN(x) := \sum_{i=1}^{p} W_i^{(2)} \left(\sum_{j=1}^{n} W_{ij}^{(1)}\, x_j + b_j^{(1)} \right)^2 .$$

The represented function is then a quadratic form, defined as the weighted sum of affine maps squared. In this setting, can we derive a lower bound (probably depending on the dimension n of the system) on p (i.e. the number of neurons) such that the network is able to generate all the positive definite quadratic forms? Using the Cholesky decomposition, we can prove that each positive definite quadratic form in dimension n can be written as the sum of n squared linear combinations of x. This result is summarized in the following theorem called Sylvester's Theorem.

Theorem 4.4 (Sylvester's Theorem)**.** *Any positive definite quadratic form* (2.3) *can be written as*

$$p(x_1, \ldots, x_n) = \sum_{k=1}^{n} (w_{k,k}\, x_k + w_{k,k+1}\, x_{k+1} + \cdots + w_{k,n}\, x_n)^2 .$$

Proof. Consider $p(x) := x^\top Q x$ with $Q \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. Using the Cholesky decomposition, there exists a lower triangular matrix $L \in \mathbb{R}^{n\times n}$ such that $Q = LL^\top$. Therefore,

$$\begin{aligned} p(x) &= x^\top L L^\top x, \\ &= \sum_{k=1}^{n} \left(L^\top x\right)_k^2, \\ &= \sum_{k=1}^{n} (L_{kk}\, x_k + L_{k+1,k}\, x_{k+1} + \ldots + L_{n,k}\, x_n)^2, \end{aligned}$$

which ends the proof. □

We can deduce a few interesting facts from Theorem 4.4. In the context of representing a quadratic Lyapunov function with a 1-layer neural network, we can first assume without loss of generality that all the biases are set at 0. Moreover, we can assume that all the output weights are fixed at 1 without modifying the representative power of the square neural network. Finally, we can provide an upper bound on the width of the network to be able to represent any quadratic Lyapunov function. The result is summarized in the following theorem.

Theorem 4.5. *Any positive definite quadratic form in dimension $n \in \mathbb{N}$ can be represented by a square neural network with* 1 *layer of width n at most.*

Let us now consider a square neural network with 2 hidden layers and a single output. In this case, the represented function is of the form

$$NN(x) := \sum_{i=1}^{p_2} W_i^{(3)} \left(\sum_{j=1}^{p_1} W_{ij}^{(2)} \left(\sum_{k=1}^{n} W_{kj}^{(1)} x_j + b_k^{(1)} \right)^2 + b_j^{(2)} \right)^2,$$

where $W^{(1)} \in \mathbb{R}^{n \times p_1}$, $W^{(2)} \in \mathbb{R}^{p_1 \times p_2}$, $W^{(3)} \in \mathbb{R}^{p_2 \times 1}$, $b^{(1)} \in \mathbb{R}^{p_1}$ and $b^{(2)} \in \mathbb{R}^{p_2}$, with p_1 and p_2 the width of layer 1 and 2 respectively. The represented function is therefore a polynomial of degree 4, defined by construction as the sum of squared polynomials of degree 2. We will prove that, by considering a sufficiently wide square neural network with 2 hidden layers, it is possible to represent any SOS polynomial of degree 4.

Following the path of ideas in the proof of Theorem 4.4 and using the Cholesky decomposition, we know that any SOS polynomial of degree 4 can be written as the sum of $N := \binom{n+1}{2}$ squared homogeneous polynomial of degree 2. This suggests that the width of the second layer should be N at most, and that the output weights can be fixed at 1. As a consequence, a square neural network with 2 hidden layers can represent any SOS polynomial of degree 4 if and only if the first hidden layer can generate any homogeneous polynomial of degree 2. In other words, can any polynomial of degree 2 be written as a weighted sum of squared affine form of x? The following theorem initially stated in [BBS08] answers in the affirmative.

Theorem 4.6 (Corollary 1 in [BBS08])**.** *Let $p(x)$ be a homogeneous polynomial in dimension n of degree d. There exists N linear forms $L_1, \ldots, L_N$ such that $p(x)$ can be written as*

$$p(x) := \sum_{i=1}^{N} \alpha_i L_i(x)^d.$$

Moreover, the number of terms in the sum is bounded by $N \leq \binom{n+d-2}{d-1}$.

This result implies a bound on the number of neurons required to express each polynomial of degree 2 in the Cholesky decomposition. Moreover, it suggests to fix the biases at 0 since the forms are linear.

Therefore, we can derive a bound on the structure of a square neural network to be able to represent any SOS polynomial of degree 4.

Theorem 4.7. *Any SOS polynomial of degree 4 in dimension $n \in \mathbb{N}$ can be represented by a square neural network with 2 hidden layers of width*

$$n_1 := \frac{n^2(n+1)}{2} \quad \text{and} \quad n_2 := \frac{n(n+1)}{2}$$

at most and unitary output weights.

Proof. Consider an SOS polynomial $p : \mathbb{R}^n \to \mathbb{R}$ of degree 4. Using the Cholesky decomposition, p can be written as the sum of n_2 homogeneous quadratic polynomials squared. Using Theorem 4.6, each of these polynomials can be written as a weighted sum of squared linear forms. Then, the second layer needs at most n_2 neurons, and each of these neurons needs at most $\binom{n}{1} = n$ neurons. Then, the first layer needs as many neurons as the product of these two numbers, i.e. n_1 neurons. □

Using similar arguments, we can prove that any SOS polynomial can be represented by a neural network with 2 hidden layers. To this aim, we need to introduce the *polynomial activation functions*.

Definition 4.8 (Polynomial activation function)**.** The *polynomial activation function of degree d* is the component-wise power function of degree d, i.e.

$$\sigma_d(x) \;:=\; \left(x_1{}^d, \ldots, x_n{}^d\right).$$

The following theorem provides a bound on the width of a polynomial neural network to represent any SOS polynomial as a function of the dimension and the degree.

Theorem 4.9. *Any SOS polynomial of degree* $2d$ *in dimension* $n \in \mathbb{N}$ *can be represented by a neural network with* 2 *hidden layers of width*

$$n_1 := \binom{n+d-1}{d}\binom{n+d-2}{d-1} \;\; \text{and} \;\; n_2 := \binom{n+d-1}{d}$$

at most, unitary output weights and with σ_d *and* σ_2 *as activation functions respectively.*

Proof. Consider an SOS polynomial $p : \mathbb{R}^n \to \mathbb{R}$ of degree $2d \in \mathbb{N}$ defined by $p(x) = (x^{[d]})^\top Q x^{[d]}$ where Q is a positive definite matrix. Using the Cholesky decomposition, the polynomial p can be written as the sum of squared linear combinations of the monomials of degree d, i.e. homogeneous polynomials of degree d. Moreover, this sum admits as many terms as the dimension of $x^{[d]}$, i.e. $\binom{n+d-1}{d}$. Finally, using Theorem 4.6, each term of this sum can be expressed as a linear combination of $\binom{n+d-2}{d-1}$ linear forms raised to the power d. The whole procedure is illustrated in Figure 4.2. □

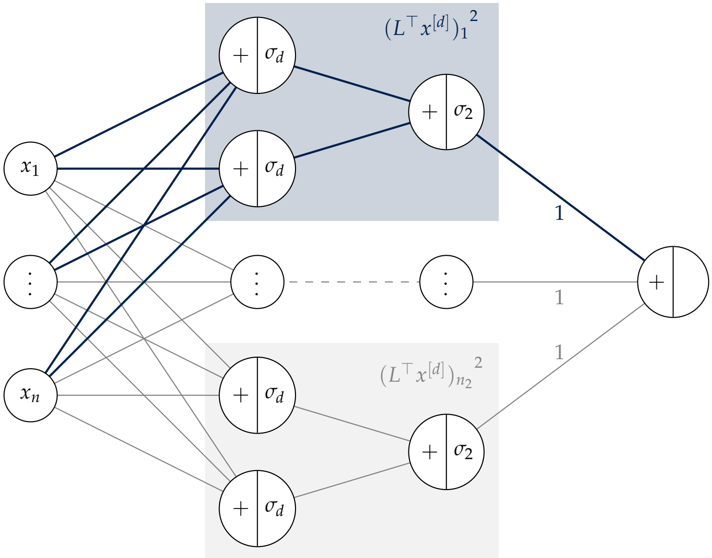


Fig. 4.2 Illustration of the procedure to represent any SOS polynomial $p : x \in \mathbb{R}^n \mapsto x^\top Q x$ with a polynomial neural network in Theorem 4.9. Each term of the Cholesky's decomposition ($Q = LL^\top$) is associated with one neuron in the second hidden layer and requires several neurons in the first hidden layer, as highlighted in bold.

4.2.2 ReLU activation function

In this section, we consider *Rectified Linear Units* activation functions. Those networks have been widely used because they represent a large family of functions using relatively few parameters. We summarize results from the literature which show how structural properties of ReLU neural networks (in particular their width and their depth) affect their representation power. This question has recently been the subject of numerous studies; we refer in particular to the following references [LPW+17, ABMM18, HBDSS21, DK22].

Let us begin with the formal definition of the ReLU activation function.

Definition 4.10 (ReLU activation function)**.** The *Rectified Linear Units (ReLU for short) activation function* is the component-wise rectifier function, i.e.

$$\sigma(x) = (\max\{0, x_1\}, \dots, \max\{0, x_n\}) .$$

The following theorem not only states that every function represented by a ReLU neural network is a continuous piecewise linear function but it also proves that the reverse is also true, i.e. every CPWL function can be represented by a ReLU neural network and a bound of the depth of this network is also provided.

Theorem 4.11 (Theorem 2.1 in [ABMM18]). *Every ReLU neural network $NN : \mathbb{R}^n \to \mathbb{R}$ represents a piecewise linear function, and every piecewise linear function on $\mathbb{R}^n$ can be represented by a ReLU neural network with at most $\lceil \log_2(n+1) \rceil + 1$ depth.*

Remark 4.12. Note that the demonstration of Theorem 4.11 mainly relies on two auxiliary and complementary results.

First, [WS05, Theorem 1] states that any piecewise linear (resp. affine) function $f : \mathbb{R}^n \to \mathbb{R}$ can be expressed as

$$f \;:=\; \sum_{j=1}^{p} s_j \left(\max_{i \in S_j} l_i \right),$$

where $l_1, \ldots, l_k$ are linear (resp. affine) functions and $S_1, \ldots, S_p \subseteq \langle k \rangle$ of cardinality at most $n+1$.

The second result concerns the minimal representation of the maximum of a fixed number of scalars, i.e. the function

$$f(x) \;:=\; \max\{x_1, \ldots, x_k\},$$

using a ReLU neural network. Initially, it is possible to represent the maximum of two numbers x_1 and x_2 with a ReLU neural network of a single hidden layer of 3 neurons. By induction and using the associativity of the maximum operation, the maximum of k numbers can be represented by a ReLU neural network of depth $\lceil \log_2(n) \rceil + 1$ at most. Numerous articles have focused on the conservatism of this bound, such as [MB17, ABMM18, HBDSS21]. It is known that $\max\{0, x_1, x_2\}$ cannot be computed with 2 layers, while [HBDSS21, Theorem 2.5] states that the function $\max\{0, x_1, x_2, x_3, x_4\}$ cannot be computed by a ReLU neural network with 3 layers, although the authors require an additional assumption. To the best of our knowledge, the question remains open for more than 5 numbers. △

Let us consider an example of a ReLU neural network to see how its structure influences the represented function and the corresponding polyhedral partition of the state space.

Example 4.13. Consider a ReLU neural network with an input layer of 2 neurons, two hidden layers of 3 neurons without biases and an output layer of a single neuron such that the network represents a CPWL function $NN : \mathbb{R}^2 \to \mathbb{R}$. We assume that the weight matrices $W^{(1)}$, $W^{(2)}$ and $W^{(3)}$ are given by

$$W^{(1)} := \begin{bmatrix} -v_1- \\ -v_2- \\ -v_3- \end{bmatrix} := \begin{bmatrix} 0.7 & 0.5 \\ -0.2 & -0.5 \\ -0.5 & 0.4 \end{bmatrix},$$

$$W^{(2)} := \begin{bmatrix} -w_1- \\ -w_2- \\ -w_3- \end{bmatrix} := \begin{bmatrix} 0.1 & 0.2 & -0.5 \\ -0.2 & 0.5 & 0.6 \\ 0.4 & 0.3 & 0.5 \end{bmatrix},$$

and

$$W^{(3)} := \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.25 & 0.5 \end{bmatrix}.$$

Each layer plays a different role in the underlying partition of the represented function, as illustrated in Figure 4.3. Let us take a closer look at the effect of each layer.

The first layer of the network induces a *global* partition of the state space, as shown in Figure 4.3a. Indeed, each neuron defines a half-space where the ReLU activation function is activated, i.e.

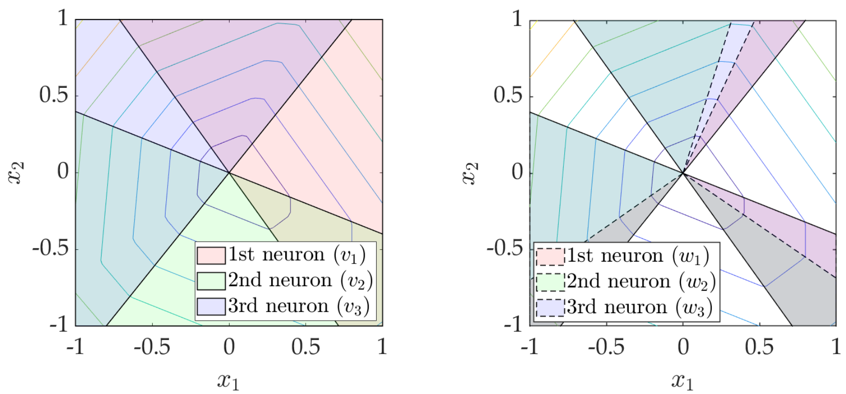
$$\begin{aligned} & \{x \in \mathbb{R}^2 \mid \sigma(v_i x) > 0\}, \\ \Leftrightarrow \quad & \{x \in \mathbb{R}^2 \mid v_i x > 0\}. \end{aligned}$$

Then, the global partition is the outcome of the intersections of the 3 half-spaces. In this example, the partition consists of 6 cells, 3 of which have only one active neuron and the 3 others admit 2 active neurons.

Each cell of the global partition is then partitioned by the second layer, as reflected in Figure 4.3b. Indeed, each neuron of the second layer computes a linear combination of the active and inactive neurons of the first layer, and then passes it through the ReLU activation function which defines half-spaces, i.e.

$$\left\{x \in \mathbb{R}^2 \mid w_i(1)\sigma(v_1 x) + w_i(2)\sigma(v_2 x) + w_i(3)\sigma(v_3 x) > 0\right\}$$

where the active neurons of the first layer depend on the cell in Figure 4.3a. Therefore, the intersection of those half-spaces refines each cell of the global partition. The only cells that are not modified are the ones where a single neuron of the first layer is active.



(a) Global partition induced by the first hidden layer. For each neuron, we colour the portion of the state space where it is active, i.e. $\sigma(v_i x) > 0$. We identify 6 global cells, 3 of them where a single neuron is active and the 3 others where 2 neurons are simultaneously active.

(b) Local refinement of the partition induced by the second hidden layer. For each cell of the global partition, we colour the portion of the cell where each neuron is active, i.e. $\sigma\left(w_i\left[\sigma(v_1 x);\sigma(v_2 x);\sigma(v_3 x)\right]\right) > 0$. We deliberately omit the cells where a single neuron of the first layer is active.

Fig. 4.3 Illustration of the sublevel sets of the function represented by the ReLU neural network with 2 hidden layers of 3 neurons in Example 4.13 and the corresponding partition of the state space.

The final partition involves 10 polyhedral cells over which the function is linear such that the sublevel sets of the function are polytopes, as illustrated in Figure 4.3. $\triangle$

While Theorem 4.11 provides a bound on the depth, it does not provide insights on the size of the corresponding network to represent a given piecewise linear function. For $n = 1$, the following result provides a tight bound.

Theorem 4.14 (Theorem 2.2 in [ABMM18]). *Given any piecewise linear function $f : \mathbb{R} \to \mathbb{R}$ with p pieces, there exists a 2-layer ReLU neural network with at most p nodes that can represent f. Moreover, any 2-layer ReLU neural network that represents f has size at least $p - 1$.*

The problem has been further studied in [HBDSS21] to any dimension $n \in \mathbb{N}$. First, the authors derive a similar result for the case of convex CPWL functions, and then use it to generalize this result to the non-convex case using Proposition 3.10.

Theorem 4.15 (Theorem 4.2 in [HBDSS21]). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex CPWL function with p affine pieces. Then f can be represented by a ReLU neural network with depth $\lceil \log_2(n+1) \rceil + 1$ and width $\mathcal{O}(p^{n+1})$.*

Theorem 4.16 (Theorem 4.4 in [HBDSS21]). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a CPWL function with p affine pieces. Then, f can be represented by a ReLU neural network with depth $\lceil \log_2(n+1) \rceil + 1$ and width $\mathcal{O}(p^{2n^2+3n+1})$.*

Remark 4.17. Note that these results follow the same philosophy as the results stated in [LS17]. However, Theorems 4.11, 4.15 and 4.16 discuss the *exact representation* of CPWL functions by ReLU neural networks with respect to their structures, whereas [LS17] is interested in their ε-close approximation capabilities. △

In order to represent candidate Lyapunov functions, that are especially positively homogeneous functions by Definition 1.10, the following proposition ensures that we can consider bias-free ReLU neural networks without affecting their representation capabilities.

Proposition 4.18 (Proposition 2.3 in [HBDSS21]). *If a ReLU neural network computes a positively homogeneous function, then the corresponding homogenized neural network computes the same function.*

This proposition mainly rests on the positive homogeneity (of degree 1) of ReLU activation function. This implies in particular that each output of a ReLU neural network is positively homogeneous as well. Therefore, by Proposition 4.18, we can consider a ReLU neural network without biases while keeping the same expressitivity power.

4.3 Summary

With the aim of using neural networks to synthesize Lyapunov functions, we focused in this chapter on their representation power. In particular, we have expressed the class of functions that a network can represent with respect to its structure, namely its depth and width, and the activation function.

Summary of Chapter 4

In this chapter, we have studied and summarized recent results on the representation power of polynomial and ReLU neural networks, i.e. their capabilities on the exact representation of polynomials and CPWL functions respectively. In particular, we focused on the representation of Lyapunov functions. For both activation functions, we have demonstrated that one can assume that all the biases are set at 0 without reducing the representative power of the network.

Section 4.2.1: Representation power of polynomial neural networks

Bias-free square neural networks with unitary output weights represent SOS polynomials whose degree increases with the depth of the network while the representation power relies on the width. Using the Cholesky decomposition, we managed to prove that a wide enough square neural network can generate any quadratic and quartic SOS polynomial. Moreover, any SOS polynomial can be represented by a polynomial neural network with 2 hidden layers.

Section 4.2.2: Representation power of ReLU neural networks

In general, ReLU neural networks represent continuous piecewise affine or linear functions. Their structure directly influences the underlying polyhedral partition of the state space. In particular, recent results prove that the depth of a ReLU neural networks only depends on the dimension of the function, and the wider the network is, the more complex the partition of the state space is. Moreover, the width depends on the number of pieces of the piecewise affine/linear function that we want to represent.

5
Approximation of the JSR using ReLU neural networks

THE interaction between Lyapunov theory and learning has motivated emerging research in recent years. For instance, [BTSK17] or [FMP21] employ Lyapunov approaches to certify safety conditions in AI-based systems. Other works (among which the present one), motivated by the great performance of neural networks in many computational technologies, tackle the other direction and use AI techniques in order to learn Lyapunov functions [Pro94, Ser05, CRG19, AAE+21, DQGF21, FLYL22, ZXQF23]. They provide promising proofs of concept for automated, data-driven control solutions that are agnostic of the particular properties of the considered control system. Indeed, a prime advantage of neural networks is that they can be deployed in an unknown setting and on complex systems, such as Cyber-Physical Systems, whereas classical control techniques would require the verification of technical properties (such as Lipschitz continuity, linearity, or passivity) that are out of hand in practice.

One of the main paradigms along these lines is a situation where a neural network is constructed from observed trajectories, and its input-output relation is interpreted as a Lyapunov function for the system. Our work falls within this paradigm. However, a crucial limitation of neural techniques is that they rarely come with guarantees; that is, the obtained network is tailored to satisfy the properties of a Lyapunov function *on the observed trajectories*, but very little is known about its generalisation properties out of the sample set. In safety-critical application, one needs generalisation guarantees, where the observed performance of the Lyapunov function is guaranteed to hold for the whole universe, not only the sampled set. Even more, Lyapunov functions must have specific properties, which might be violated by the obtained neural network (think simply of positive definiteness).

In this work, we are primarily interested in understanding this (potential) gap between high performance on the sampled set, and behaviour on the true system. For this purpose, our strategy is to study this approach on switched systems, that are well-understood systems for which alternative computational techniques have been developed as summarized in Section 2.2. Then one can compare the performance of neural networks, both in terms of computational performance and in terms of accuracy.

As a first theoretical contribution, we provide similar theoretical approximation guarantees for neural network-based Lyapunov functions. For this, we leverage results from convex geometry, and combine them with recent results in Machine Learning, about the representation power of neural networks which have not previously been tailored to Lyapunov functions constructed as neural networks.

Our second contribution is empirical. It is well known that ReLU neural networks compute piecewise linear functions. Such functions are another popular technique to compute the JSR, however with less efficient computational power: the computation of a piecewise linear norm cannot be achieved efficiently since it requires to solve a bilinear program [AJ18b], which cannot be done in polynomial time. ReLU neural networks offer an alternative to compute this type of Lyapunov functions. We thus provide numerical experiments in order to benchmark this alternative way of obtaining piecewise linear Lyapunov functions. We first consider a low-dimensional switched system as proof of concept where we observe that we are competitive in terms of approximation precision. Then we increase the dimension to show that we compete with SDP-based techniques both in terms of computation time and approximation precision. Then, we provide an overview of technical problems that are encountered, and investigate techniques from Machine Learning to alleviate these problems.

In particular, we show through examples that we cannot trust the output of the neural network. Indeed, the network provides a JSR upper bound which is only valid on the sample points, but not necessarily over the whole state space. We draw inspiration from [AAE+21] which proposes a CounterExample-Guided Inductive Synthesis (CEGIS) algorithm where a neural network is trained to represent a Lyapunov function whose validity is soundly checked by a Satisfiability Modulo Theories [BSST21] (SMT for short) solver thereafter. This approach benefits from the flexibility and the representation power of neural networks, and it has already shown promising results. In this chapter, we take a similar approach to handle the approximation of the JSR. Therefore, the use of an SMT solver is required to check the Lyapunov inequalities over the whole domain. The neural network and the SMT solver alternate until a valid approximation is found, or the procedure stops when the maximal number of iterations is reached. In case of failure, we introduce a post processing step which leverages the knowledge of the neural network on the sample points to derive a valid approximation of the JSR despite the failure. Finally, we test our new algorithm across several benchmarks.

This chapter results from a collaboration with Alec Edwards and Alessandro Abate, and this work has been partially published in [DEJA24].

5.1 Bounds on the structure of the network

The computation of a polytopic norm cannot be achieved efficiently since it amounts to solving a bilinear program, as recalled in Chapter 3. Therefore, convinced by the numerical efficiency and the representation power of ReLU neural networks introduced in Chapter 4, we decide to use them to synthesize piecewise linear Lyapunov functions. Moreover, we combine Theorems 3.18 and 4.15 derive bounds on the network structure to achieve a given precision on the JSR approximation.

In particular, Theorem 3.18 provides a relation between the polytopic approximation precision and the complexity of its polytopic sublevel sets while Theorem 4.15 provides bounds on the width and depth of a ReLU network to represent a given function with fixed complexity. Therefore, by combining them, we are able to provide upper bounds on the network's structure to reach a given precision τ on the approximation of the joint spectral radius. Then, the following theorem summarizes this result. To the best of our knowledge, this is the first universal approximation result tailored to neural Lyapunov functions.

Theorem 5.1. *Let $\rho(\mathcal{A})$ be the joint spectral radius of a finite set of matrices $\mathcal{A}$ of dimension n. For any real $\tau > 1$ for which there exists $k_\tau \in \mathbb{N}$ satisfying Equation (3.9), there exists a convex CPWL function represented by a ReLU neural network of depth*

$$\lceil \log_2(n+1) \rceil + 1$$

and width

$$\mathcal{O}\left([8D(n, k_\tau)]^{n+1} \right),$$

where $D(n, k_\tau)$ is defined in Equation (3.10), which approximates $\rho(\mathcal{A})$ with a precision of τ.

Proof. The theorem results from the successive application of Theorems 3.18 and 4.15. □

Recalling Corollary 3.17 where τ is taken arbitrarily close to 1, the following corollary can be stated.

Corollary 5.2. *Let $\rho(\mathcal{A})$ be the joint spectral radius of a finite set of matrices $\mathcal{A}$ of dimension n. For any real $\tau := 1 + \varepsilon > 1$ for which there exists $k_\tau \in \mathbb{N}$ satisfying Equation (3.9), there exists a convex CPWL function represented by a ReLU neural network of depth*

$$\lceil \log_2(n+1) \rceil + 1$$

and width of order

$$\mathcal{O}\left(8^{n+1} n^{\frac{n(n+1)}{4}} \varepsilon^{\frac{1-n^2}{2}} \right)$$

which approximates $\rho(\mathcal{A})$ with a precision of τ.

The contribution of Theorem 5.1 is twofold. First, it provides a proof of concept that neural networks can approximate the JSR up to an a priori fixed guarantee of accuracy. Moreover, Theorem 5.1 provides bounds on the architecture of a ReLU neural network so that it can represent a specific Lyapunov function. However, it gives no indication of how the network should be trained to achieve this function. In addition, all these bounds above are on the worst case; we expect to achieve a given precision with smaller networks than the architecture recommended in Theorem 5.1. The rest of this chapter investigates this question through several examples.

5.2 Experimental evaluation

In this section, we provide an empirical investigation of the practical efficiency of the neural approach. We first describe the configuration of the experimental evaluation, i.e. the choice of the loss function, the sampling method, etc. Moreover, we compare our numerical results with more classical SDP-based approaches recalled in Sections 2.2.1 and 2.2.2

First, we consider a low-dimensional switched system as proof of concept where we observe that we are competitive in terms of approximation precision. Then, we increase the dimension to show that we compete with SDP-based techniques both in terms of computation time and approximation precision. Note that, contrary to the SDP-techniques, the neural-based approach is *data-driven*, which needs to be taken into account when we compare both methods.

5.2.1 Setup

Given a linear switched system $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n\times n}$ in dimension $n \in \mathbb{N}$, we consider a ReLU neural network to instantiate a candidate Lyapunov function $NN : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$. By training it appropriately, we aim to achieve the best neural JSR approximation. By Theorem 2.13, we know that any continuous, positive and homogeneous function can be used to compute an approximation of the JSR. Indeed, given such a function V, we can derive an approximation of the JSR by computing the smallest value $\gamma > 0$ such that the Lyapunov inequalities in Equation (2.2) and recalled hereunder, i.e.

$$\forall i \in \langle M\rangle,\ \forall x \in \mathbb{R}^n :\ V\left(\frac{A_i}{\gamma}x\right) \leq V(x),$$

are satisfied by V, where $\mathcal{A}_\gamma := \{A_i/\gamma : i = 1, \ldots, M\}$ denotes the scaled system by γ.

In what follows, we detail the choices that we have made:

- Structure of the network: Let us first check that a ReLU neural network satisfies these properties. By Proposition 4.18, we can assume (without loss of representation power) that the network is bias-free, such that the represented function is positively homogeneous. Thus, any function represented by such a network is continuous, radially unbounded and positively homogeneous by construction. Moreover, we need a $(n+1)$-width network at least to cover the whole state space, and we structurally enforce during training the positivity of the represented function by taking the absolute value of the output weights.

- Loss function: We want the learning process to improve and refine the JSR approximation provided by the function represented by the network. We therefore choose the loss function to correspond to the sample-based JSR approximation, i.e.

$$Loss(NN, \mathcal{S}) := \max_{i=1,\dots,M} \max_{x \in \mathcal{S}} \frac{NN(A_i x)}{NN(x)}, \tag{5.1}$$

where $NN : \mathbb{R}^n \to \mathbb{R}$ is the function represented by the network and $\mathcal{S}$ is the training sample set. Therefore, by definition of the loss function, the function $NN(\cdot)$ satisfies the Lyapunov inequalities in Equation (2.2) on the sample points for the scaled system $\mathcal{A}_\gamma$, where $\gamma := Loss(NN, \mathcal{S})$. Indeed, $\forall i \in \langle M \rangle$ and $\forall x \in \mathcal{S}$:

$$\begin{aligned} \frac{NN(A_i x)}{NN(x)} &\leq \gamma, \\ \Leftrightarrow \quad NN(A_i x) &\leq \gamma NN(x), \\ \Leftrightarrow \quad NN\left(\frac{A_i}{\gamma} x\right) &\leq NN(x), \end{aligned}$$

where the first equivalence is satisfied by positivity of the function $NN(\cdot)$, and the second equivalence holds by positive homogeneity. However, nothing guarantees that this quantity is a *valid* upper bound on the JSR, i.e. $Loss(NN, \mathcal{S}) \geq \rho(\mathcal{A})$. It would be if $\mathcal{S}$ was replaced by $\mathbb{R}^n$, in Equation (5.1) above. This issue will be partially tackled in Section 5.3 and completely overcome in Section 5.4.

Remark 5.3. Note that, if the function represented by the network $NN(\cdot)$ is convex, and therefore a norm, one can compute the matrix norm induced by $NN(\cdot)$ of each matrix $A_i \in \mathcal{A}$. Moreover, the loss function defined in Equation (5.1) is the sample-based approximation of the maximal matrix norm induced by NN over $\mathcal{A}$. △

- Sampling: Finally, we need to define our sampling policy. By homogeneity of the $NN(\cdot)$ and the matrices $\mathcal{A}$, it is sufficient to satisfy the Lyapunov inequalities in Equation (2.2) on the unit ball. Therefore, we train the network with a sample set $\mathcal{S}$ of data points which are uniformly distributed on the unit sphere.

We can now define the JSR approximation provided by a ReLU neural network as the smallest loss value during the training period.

Definition 5.4 (Neural JSR approximation)**.** Consider a switched linear system $\mathcal{A} := \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n \times n}$ and a ReLU neural network with k hidden layers of width m, which has been trained during L iterations over a sample set $\mathcal{S}$. The *neural approximation* of the JSR, denoted by $\rho_{NN(k,m),\mathcal{S}}(\mathcal{A})$, is defined as the lowest value of the loss function in Equation (5.1) during the entire training campaign, i.e.

$$\rho_{NN(k,m),\mathcal{S}}(\mathcal{A}) := \min_{l=1,\ldots,L} Loss(NN_l, \mathcal{S}),$$

where NN_l denotes the function represented by the neural network at the l-th training iteration.

Remark 5.5. Note that the neural JSR approximation defined in Definition 5.4 depends on the sample set $\mathcal{S}$, and should be properly defined as a random variable. Since we do not want to study the corresponding statistical properties, we will use this abuse of notation for the sake of simplicity.

The outcome of this learning procedure is twofold since it instantiates a candidate Lyapunov function $NN : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and provides a sample-based approximation of the JSR of $\mathcal{A}$, denoted by $\rho_{NN(k,m),\mathcal{S}}(\mathcal{A})$. Let us note once again that the function NN is guaranteed to satisfy the Lyapunov inequalities for the scaled system $\mathcal{A}_\gamma$ for $\gamma = \rho_{NN(k,m),\mathcal{S}}(\mathcal{A})$ only on the sample points $\mathcal{S}$. However, the Lyapunov inequalities might be violated elsewhere.

5.2.2 Numerical results

As reference benchmarks through this chapter, we consider two switched linear systems in dimension 2 and 8 respectively, which are challenging as they are known to lead to poor approximation with classical techniques, see [DDJ23, Section 5] and [AJPR14, Example 5.2]. The first one serves as proof of concept, while the second example shows that in higher dimensions, neural Lyapunov functions are competitive with SOS-based techniques both in terms of precision and (as suggested earlier in Figure 3.1) in computation time. Note that all experiments were run on an Intel i7 laptop with 4 cores and 8GB of RAM. Moreover, the network is trained using the stochastic gradient descent optimizer AdamW, introduced in [LH17].

Example 5.6. We consider the switched linear system $\mathcal{A}_2 := \{A_1, A_2\} \subset \mathbb{R}^{2\times 2}$ defined by the matrices

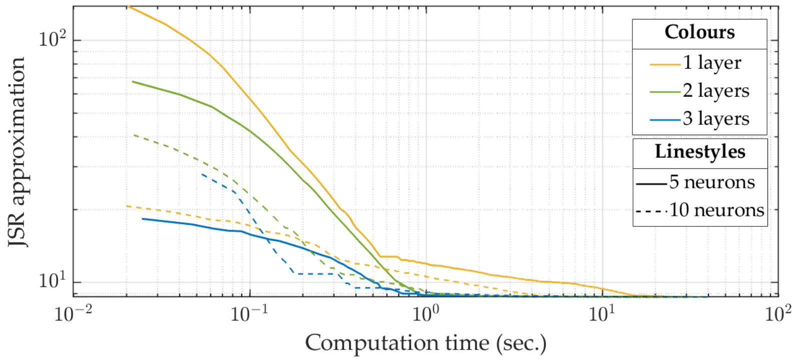
$$A_1 = \begin{bmatrix} 1.5519 & 0.4474 \\ 7.6412 & 7.4716 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} 0.4750 & 9.1755 \\ 1.8955 & 0.1850 \end{bmatrix}. \tag{5.2}$$

Using the JSR toolbox in [VHJ14], we compute that the joint spectral radius of $\mathcal{A}_2$ is 8.6881, the ellipsoidal approximation $\rho_Q(\mathcal{A}_2)$ is 9.5868 and the SOS approximation of degree 4, i.e. $\rho_{SOS,4}(\mathcal{A}_2)$, is 8.7203.

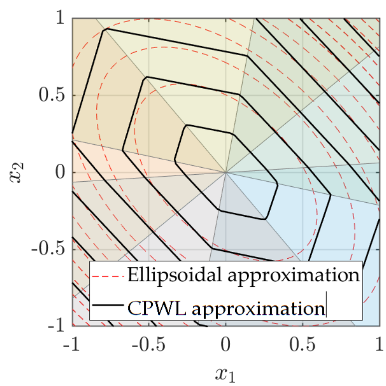
Let us consider a ReLU neural network with different depths and widths that we train with 500 sample points for 20 different seeds. The best (i.e. the smallest) approximation, the mean and the standard deviation are summarized in Table 5.1. Not surprisingly, the more neurons there are, the better the approximation. We also observe that with more neurons, there is significantly less variability with respect to the seed. In terms of approximation precision, the neural approach is more efficient than both the ellipsoidal and the SOS approaches, since the average approximation with 10 neurons is smaller than $\rho_Q(\mathcal{A}_2)$ and $\rho_{SOS,4}(\mathcal{A}_2)$. Figures 5.1b and 5.1c illustrate the partition and the sublevel sets of the Lyapunov function encoded by the best 1-layer network with 5 and 10 neurons respectively, and one can notice the similarity with the ellipsoidal sublevel sets. However, the computation of the ellipsoidal and the SOS approximation are very fast in such low dimension (they clock 0.2816 and 1.5156 seconds respectively), and the

		Neural approximation of the JSR $\rho_{NN(k,m),\mathcal{S}}(\mathcal{A}_2)$		
k	m	Best	Mean	Std.
1 **layer**	5 neurons	8.6977	9.0251	0.8800
	10 neurons	8.6910	8.6969	0.0056
2 **layers**	5 neurons	8.6983	8.9312	0.4645
	10 neurons	8.6944	8.7049	0.0077
3 **layers**	5 neurons	8.6967	9.1984	0.7293
	10 neurons	8.6946	8.7130	0.0175

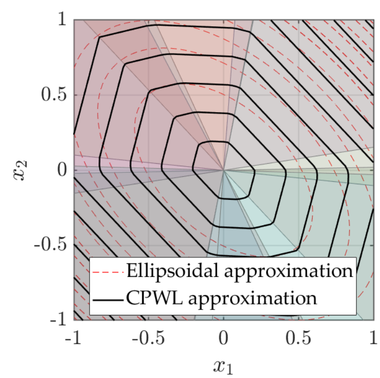
Table 5.1 Best and mean/std. (over 20 seeds) approximation of the JSR of system (5.2) provided by a ReLU neural networks with different architectures.



(a) Evolution with the computation time of the loss function of a ReLU neural network with different widths and depths. For each configuration, only the best seed (which provides the lowest approximation) is illustrated.



(b) Sublevel sets of the best CPWL approximation with 1 hidden layer and 5 neurons, and the partition induced by the network. The sublevel sets of the ellipsoidal approximation have been added for comparison.



(c) Sublevel sets of the best CPWL approximation with 1 hidden layer and 10 neurons, and the partition induced by the network. The sublevel sets of the ellipsoidal approximation have been added for comparison.

Fig. 5.1 Approximation of the JSR of system (5.2) using a ReLU neural network with 500 sample points, alongside sublevel set of the best approximation from two networks. With 5 neurons (5.1b) , the CPWL approximation appears geometrically similar to the ellipsoidal approximation. As more neurons are added (5.1c), the geometry of the sublevel sets diverges from the ellipsoidal approximation, corresponding to an improved approximation of the JSR (Table 5.1).

novel neural network approach cannot compete with them: for example, Figure 5.1a shows that in the best case, we need a few seconds, but in the worst cases, we need almost one minute. △

Let us now consider a 8-dimensional switched system, such that the computation time of the ellipsoidal and SOS approximation are larger and the neural approach starts to be competitive.

Example 5.7 (High dimension). We consider a switched system in dimension 8 with 8 different modes, defined by the matrices $\mathcal{A}_8 := \{A_i : i \in \langle 8 \rangle\} \subset \{0,1\}^{8\times 8}$ such that for $i = 2, \ldots, 8$,

$$A_i(k,l) := \begin{cases} -1 & \text{if } k = l = i, \\ 1 & \text{if } l = i \text{ and } k \neq i, \\ 0 & \text{otherwise}, \end{cases} \tag{5.3}$$

while the matrix $A_1 = \mathbf{1}e_1^\top$. One can prove that the joint spectral radius of this finite set of matrices is 1, i.e. $\rho(\mathcal{A}_8) = 1$. Regarding the approximation of the JSR using classical techniques, the ellipsoidal approximation generates $\rho_{\mathcal{Q}}(\mathcal{A}_8) = 2.4286$ but the computation is relatively fast ($\approx$ 2.5 seconds), while the SOS approximation is better since $\rho_{SOS,4}(\mathcal{A}_8) = 1.0006$ but it requires much more computation time ($\approx$ 258 seconds).

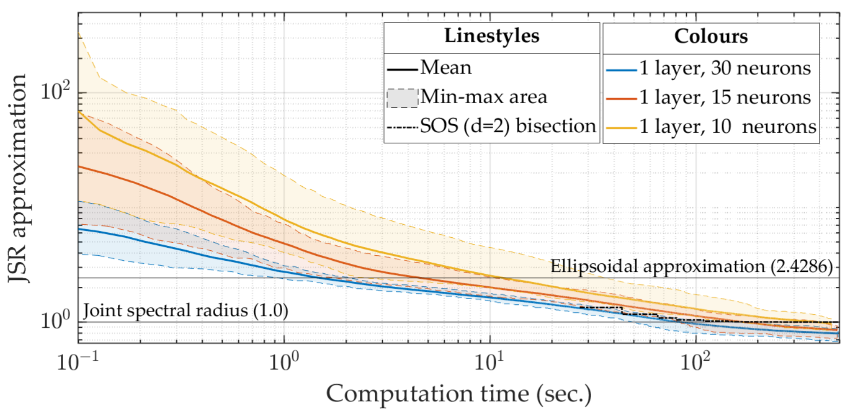


Fig. 5.2 Evolution of the JSR approximation for the 8-dimensional system (5.3) provided by a ReLU neural network with different number of hidden layers and different numbers of neurons. The value of the JSR, the ellipsoidal approximation and the SOS degree-4 (in black) bisection have been added for comparison.

For the neural approximation of the JSR, we use a ReLU neural network with a single hidden layer and different numbers of neurons: 10, 15 or 30. We train it using 500 sample points. Figure 5.2 shows the evolution of the mean and the min-max area of the JSR approximation with the computation time for each configuration. One can see that in dimension 8, the neural approach is almost as fast as the SOS ($d = 2$) method. However, the network overfits the data and then provides an approximation which is smaller than the true JSR value, except with 10 neurons where half of the seeds provide an approximation larger than 1. One way to prevent this behaviour is to consider more sample points. However, the computation time increases as the sample set grows and we are no longer competitive with SDP-based techniques. △

5.3 Improvement techniques

As we have seen, the numerical experiments suffer from classical issues in Machine Learning such as overfitting, large number of sample points needed, etc. In this section, we consider some methods to mitigate these behaviours. In particular, one way to overcome this problem is the *regularization*, which refers to a technique used to prevent overfitting and improve the generalisation performance of a model. In practice, it amounts to adding a penalty term in the loss function during the training. The objective is to discourage the network from reaching too complex functions. In addition, we will consider the concept of *incremental learning* which takes advantage of the computation speed when we consider few sample points at the beginning and progressively add new well-chosen sample points during the training.

5.3.1 Regularisation

As can be seen in Figure 5.2, we have shown that neural networks tend to overfit the data resulting in an incorrect estimate of the JSR. One classical method to prevent overfitting is regularization, i.e. for the problem at hand we penalize the network by adding a new term in the loss function to prevent it from learning overly-complex functions. Depending on the penalty term, the goal of the regularization might be slightly different. We review a few of the most commonly used regularization methods in this section.

First, we introduce the L_1 and L_2 regularization methods. The idea is to add a penalty term to the loss function which only depends on the complexity of the network. The objective is to reduce the weights matrices by assuming that a neural network with less weights makes simpler models. Here we consider the two most classical methods:

- The L_1-*regularisation*, also called *Lasso regularization*, adds as penalty term the L_1-norm of the weights, i.e. the sum of the absolute value of all the coefficients of the weight matrices. In this case, the loss function is defined as

  $$Loss_{L_1}(NN, \mathcal{S}) \;:=\; Loss(NN, \mathcal{S}) \;+\; \lambda \sum |w_i|.$$

  This method will drive some coefficients to zero, and therefore provides sparse solutions. This method is suitable for high-dimensional datasets or when there is a need for feature selection and interpretability.

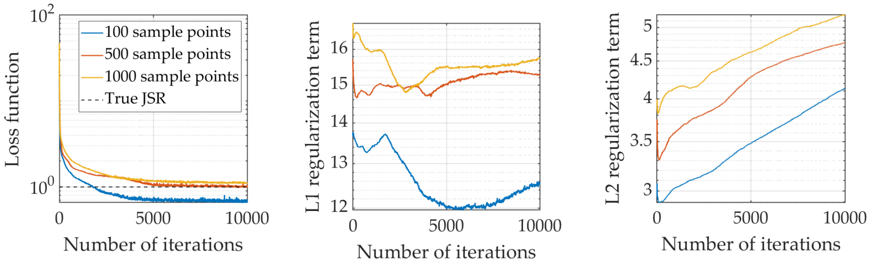
- The L_2-*regularisation*, also called *Ridge regularization*, adds as penalty terms the L_2-norm (i.e. the Euclidean norm) of the weights, i.e. the sum of the square of the coefficients of the weight matrices. The loss function is therefore given by

  $$Loss_{L_2}(NN, \mathcal{S}) \;:=\; Loss(NN, \mathcal{S}) \;+\; \lambda \sum {w_i}^2.$$

  This method will encourage smaller coefficients such that there is no domination from one feature (it reduces the individual impact but it allows all features to contribute to the model's predictions), and therefore promotes overall weight shrinkage. Then, it is effective when dealing with strong feature correlations or when there is no specific need for feature selection.

The *Elastic Net regularization* is a technique that combines both L_1 and L_2-regularization to achieve a balance between feature selection and weight shrinkage. During model training, it incorporates both the L_1 and L_2 regularization terms in the loss function. The values of the corresponding regularization parameters λ_1 and λ_2 control the balance between sparsity and weight shrinkage.

Let us consider an example to see how regularisation influences the approximation of the JSR provided by a ReLU neural network.



(a) Evolution with the training iterations of the loss function.

(b) Evolution with the training iterations of the L_1-norm of the weights.

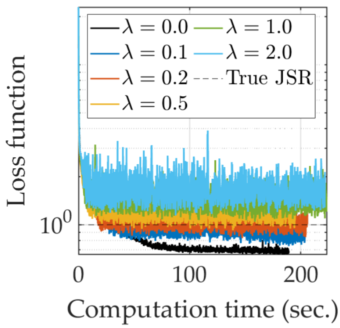
(c) Evolution with the training iterations of the L_2-norm of the weights.

Fig. 5.3 JSR approximation of system (5.3) using a ReLU neural network with 1 layer, 10 neurons in each hidden layer trained with 100 (in blue), 500 (in red) and 1000 (in yellow) sample points.

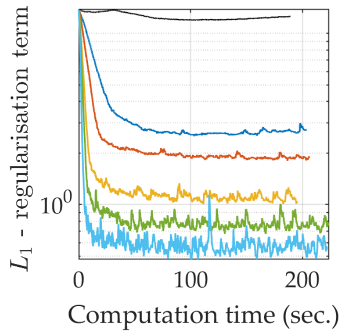
Example 5.8. We consider the 8-dimensional switched system in (5.3). We use a ReLU neural network with a single hidden layer of 10 neurons that we have trained with 100, 500 or 1000 sample points. We show in Figure 5.3 the evolution with the iterations of the loss function (Figure 5.3a) and the L_1 and L_2 norms (Figures 5.3b and 5.3c). In all cases, the L_2-norm of the weights increases while the L_1-norm first varies and then remains more or less constant with 500 and 1000 sample points. With 100 sample points, the L_1-norm first decreases and then increases slightly. Therefore, forcing the L_1-norm or the L_2-norm to decrease by adding a regularization term to the loss function might be useful to prevent overfitting. We regularize the network with different values for the regularization parameter λ with respect to the L_1 and L_2 norms. The results are illustrated in Figure 5.4 and summarized in Table 5.2.

	Value of the regularization parameter λ							
	0.0	0.1	0.2	0.5	1.0	2.0	5.0	10.0
L_1-norm	0.6502	0.7356	0.8262	0.9229	1.0462	1.1442	×	×
L_2-norm	0.6502	0.6998	0.6770	0.6953	0.7366	0.8091	0.8770	0.9245

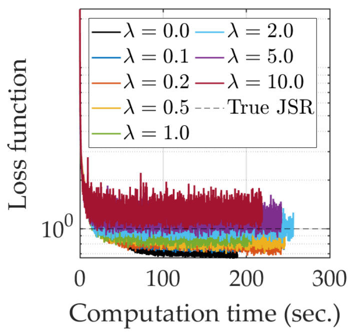
Table 5.2 Neural JSR approximation of system (5.3) provided by a ReLU neural network with 1 hidden layer of 10 neurons using 100 sample points. The invalid upper bounds are highlighted in color.



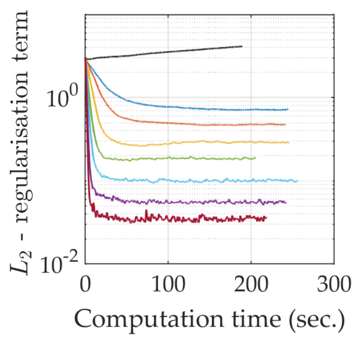
(a) Evolution with the computation time of the loss function with a L_1-regularization term.



(b) Evolution with the computation time of the L_1-regularization term.



(c) Evolution with the computation time of the loss function with a L_2-regularization term.



(d) Evolution with the computation time of the L_2-regularization term.

Fig. 5.4 JSR approximation of system (5.3) using a ReLU neural network with 1 hidden layer of 10 neurons trained with 100 sample points. We use L_1 and L_2 regularization techniques to prevent the network from overfitting with different regularization parameter values.

First, we observe that the regularisation works as expected since the L_1-norm and the L_2-norm reduces as the parameter λ increases in Figures 5.4b and 5.4d respectively. Moreover, as λ increases, the approximation of the JSR reaches larger and larger values and then starts to stop overfitting, as illustrated in Figures 5.4a and 5.4c. This behaviour is more pronounced and effective with the L_1-regularisation, as reflected in Table 5.2. $\triangle$

The regularization techniques are used to prevent the neural network from reaching complex functions. Then these functions might be Lipschitz continuous with a large Lipschitz constant, such that the function might vary a lot and adapt to the sample points. However, the smaller the Lipschitz constant is, the more the function is prevented from large variations. Therefore, we could expect that this constant increases when the network overfits the data, and it might be useful to add it to the loss function to minimize it. The so-called *Lipschitz* or *spectral regularization* has already been addressed in [YM17, GFPC18, MKKY18, VS18].

Exactly computing the Lipschitz constant of a 2-layer ReLU neural network is NP-hard [VS18]. However, since the ReLU activation function is Lipschitz continuous with constant 1, we can upperbound the Lipschitz constant of the network by

$$L_{NN} \le \prod_{l=1}^{h+1} \left\| W^{(l)} \right\|,$$

where $\|W^{(l)}\|$ is the matrix norm of W_h. By equivalence of the norms in finite dimension, optimizing the Lipschitz bound under a particular choice of norms will effectively optimize the bound measured by the other norms. However, the computation cost of these norms differ; for instance the L_2-norm , where $\|W^{(l)}\|$ refers to the largest singular value of the matrix $W^{(l)}$, is more expensive to compute than the other norms.

Rather than adding the product of the spectral norms to the loss function, which might be done without efficiently reducing the Lipschitz constant of the network, let us consider the sum of these norms as penalty term, i.e.

$$Loss_{\text{Lip}}(NN, \mathcal{S}) := Loss(NN, \mathcal{S}) + \lambda \sum_{i=1}^{h} \sigma(W_i). \tag{5.4}$$

The objective is to minimize the spectral norm of each weight matrix, such that the product decreases and so does the Lipschitz constant.

Example 5.9. We consider the switched system (5.3) in dimension 8 whose JSR is 1. We use a ReLU neural network with 1 hidden layer of 10 neurons that we train during 10000 iterations. We add the regularisation term in equation (5.4) with different values for the regularization parameter λ, and we train the network with 100 sample points. As reflected in Table 5.3, the network overfits less and less as λ increases. △

Value of the regularization parameter λ								
0.0	0.1	0.2	0.5	1.0	2.0	5.0	10.0	15.0
0.6502	0.6912	0.7176	0.7691	0.7941	0.8464	1.0471	1.1266	1.1646

Table 5.3 Approximation of the JSR of system (5.3) using a ReLU neural network with 1 layer of 10 neurons trained during 10000 iterations with 100 sample points using regularization in Equation (5.4) with different values for the parameter λ. The invalid upper bounds are highlighted in color.

5.3.2 Incremental learning

The number of sample points is a key parameter since it determines a trade-off between computation time and the risk of overfitting. However, it seems that the network does not need many points at the beginning, but progressively requires more and more points during training.

In our case, we propose to sample several new sample points, then compute the ratio

$$\max_{i=1,\dots,M} \frac{NN(A_i x)}{NN(x)}$$

for each sample points. If the maximum over all the new sample points is below the current JSR approximation of the network, we keep on training the network. Otherwise, we add the subset of the worst points, i.e. the points which provide the highest JSR approximation.

Example 5.10. We consider the switched system (5.3) in dimension 8. We use a ReLU neural network with a single hidden layer of 10 neurons that we train with 100 sample points at the beginning. In Figure 5.5, we compare two different situations. For the first one, we have trained the network during 10000 iterations without stopping it, while for the second case, we have stopped the network every 2000 iterations to possibly add new sample points.

We can clearly observe that without incremental learning, the network overfits and reaches a wrong JSR approximation. When we add new sample points during the training, we stay above the true JSR value in the long run even if sometimes the network ends up overfitting. However, at the next step, we will probably add new points with a larger JSR approximation. Moreover, we can see that the new JSR approximation when we add new sample points (the red circles in the figure) decreases which means

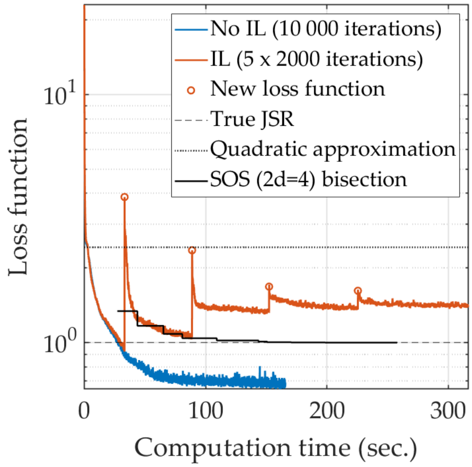


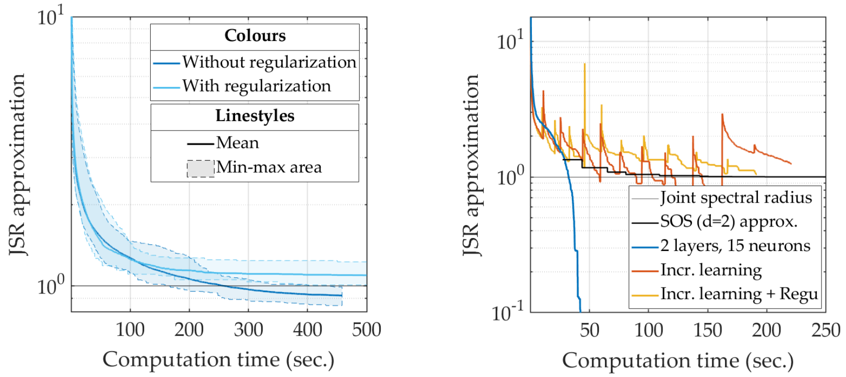
Fig. 5.5 Evolution of the JSR approximation provided by a ReLU neural network with 1 hidden layer of 10 neurons trained initially with 100 sample points during 10000 iterations with an without incremental learning for system (5.3). For the incremental learning, we possibly add the worst 100 sample points over 10000 every 2000 iterations.

that with the new well-chosen points, the network overfits less and is more able to generalize. However, the computation time increases as long as we add new sample points. △

5.3.3 Experimental evaluation

In this section, we apply the two techniques developed in previous sections to avoid overfitting in high dimension, namely the *regularization* and the *incremental learning*.

Example 5.11. We consider the 8-dimensional linear switched system in Equation (5.3) whose JSR is 1. In comparison with Figure 5.2, Figure 5.6a shows promising results using L_1-regularization. Indeed, when the network is regularized, the neural JSR approximation almost never reaches wrong approximation but requires a few more computation time. Moreover, Figure 5.6b illustrates the evolution of the neural JSR approximation provided by a 2-hidden-layer neural network with 15 neurons, where we now add successively new sample points to the training sample set. In this example, after combining the incremental learning with the regularization, the resulting network did not reach a wrong over-approximation value. △



(a) Comparison of the evolution with the computation time of the JSR approximation provided by a ReLU network with 1 hidden layer and 15 neurons with and without L1-*regularization*.

(b) Comparison of the evolution with computation time of the JSR approximation provided by a ReLU network with 1 hidden layer of 15 neurons with and without *incremental learning*.

Fig. 5.6 Illustration of two techniques to avoid overfitting in higher dimensions, namely the *regularization* and the *incremental learning*.

5.4 Computation of valid neural-based JSR approximations

In our previous numerical experiments, we have seen that the network rapidly suffers from overfitting the data, especially when the dimension increases. Therefore, the training may end up with an *invalid* JSR approximation, i.e. a smaller value than the JSR. As a result, we cannot really trust the neural JSR approximation. In this section, we are looking at two different strategies to overcome this issue.

At first, we propose a CounterExample-Guided Inductive Synthesis (CEGIS for short) implementation, that is an inductive loop between two main components: the *learner* which seeks a candidate Lyapunov function with an optimised sample-based estimate of the JSR, and the *verifier* which checks the validity of the Lyapunov inequalities over the whole domain for the candidate function. Section 5.4.1 describes in details this CEGIS architecture. In addition, we introduce a post-processing step that leverages the information acquired by the network over the sample points, then generalizes it to the entire state space. It eventually generates a valid upper approximation on the JSR, even if the CEGIS loop fails. Section 5.4.2 covers the derivation of this valid upper bound.

5.4.1 CEGIS approach

In this section, we introduce a new formal method to approximate the JSR of a finite set of matrices using machine learning algorithms. We consider a *CEGIS architecture* as implemented in the FOSSIL tool developed in [AAE+21, EPA24]. Figure 5.7 illustrates the architecture in its entirety.

This synthesis architecture involves the alternating interaction of two main components, namely a *learner* and a *verifier*. The learner seeks to submit a candidate Lyapunov function and a sample-based JSR approximation to the verifier. However, the corresponding Lyapunov inequalities must be satisfied over the whole state space to be able to derive a valid upper bound on the JSR. Therefore, the verifier will either validate or disprove the candidate stability certificate - in the latter case, the training procedure is refined at next CEGIS iteration by adding the counterexamples to the training sample set. Otherwise, the verifier confirms that there is no counterexample, which provides a valid upper bound on the JSR. As a result, the CEGIS loop stops. The communication between the two components is facilitated by two other elements, namely the *translator*, which translates the neural network into symbolic variables, and the *consolidator*, which leverages the counterexamples provided by the verifier and augments the training set of the learner. Let us look at each component one by one.

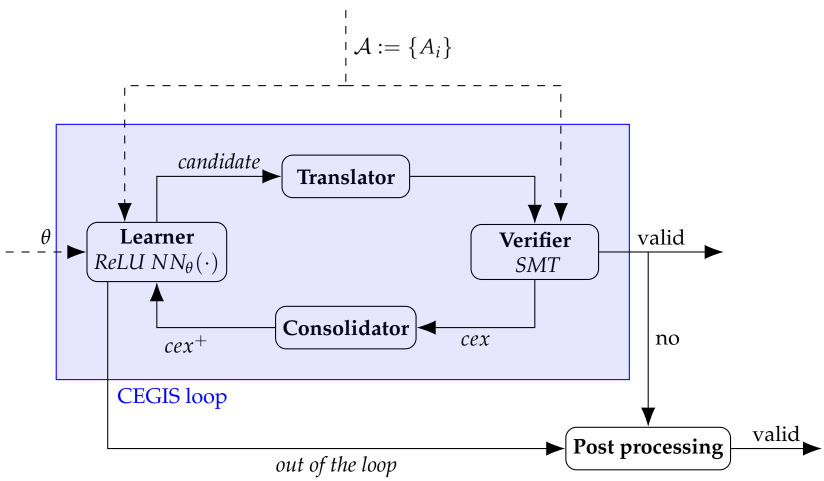


Fig. 5.7 Illustration of the CEGIS architecture of our method to provide sound upper approximation of the JSR of a finite set of matrices $\mathcal{A} \subset \mathbb{R}^{n\times n}$.

Learner: The learning component of our CEGIS architecture is a ReLU neural network which is trained following the procedure described in Section 5.2.1. After a fixed number of training iterations, the training stops and the network provides both a candidate Lyapunov function $NN : \mathbb{R}^n \to \mathbb{R}$ and the corresponding sample-based JSR approximation, defined as the loss function in Equation (5.1) at last iteration. By construction, the candidate Lyapunov function NN satisfies the Lyapunov inequalities with respect to the scaled system by the JSR approximation on the sample points.

Verifier: We use a *Satisfiability Modulo Theories* [BSST21, dMB08] (SMT for short) solver to soundly verify the Lyapunov inequalities over the whole state space, and therefore provide a valid JSR approximation.

An **SMT solver** decides the satisfiability of first-order logic formulae, by combining combinatorial and symbolic algorithms. Unlike classical numerical solvers, SMT solvers are sound and provide therefore formal guarantees. In our case, deciding whether the network provides a valid JSR approximation amounts to deciding the formula

$$\forall x \in \mathbb{R}^n : (x \neq 0) \Rightarrow (\forall i \in \langle M \rangle, NN(A_i x) \leq \gamma NN(x)),$$

where $\gamma = Loss(NN, \mathcal{S})$ in Equation (5.1). Note that we omit the condition $V(0) = 0$ by selecting in advance biases, as explained in Section 4.2.2. Moreover, by linearity of the dynamics and homogeneity of the Lyapunov function, we must only check the Lyapunov inequalities on the unit ball. For computational purpose, we use the unit ball of the infinity norm since its set is described by a linear expression, which speeds up the computation time of the SMT solver with respect to the 2-norm (which requires a nonlinear expression). If the SMT solver answers that the Lyapunov inequalities are not satisfied, it provides a counterexample, meaning a point where at least one of the Lyapunov inequalities is violated. Then this point (along with a few other points supplied by the consolidator, as we discuss below) is added to the training sample set of the neural network for the next CEGIS iteration.

In practice, the SMT solver Z3 [dMB08] is suitable because we only consider linear switched systems and piecewise linear Lyapunov functions. For more general systems, which involves non polynomial terms such as trigonometric or exponential expressions for instance, the SMT solver *dReal* could be used.

Translator and consolidator: In [AAE+21], the authors provide two additional components, namely the *translator* and the *consolidator*, which we also benefit from.

First, the translator is a practical component which ensures the transition between the neural network and the SMT solver. In practice, it translates the numerical network values into symbolic variables.

The consolidator can be seen as an optimization of the counterexample provided by the SMT solver. Indeed, the addition of a new lone sample point might not be powerful enough to influence the training of the network at the next CEGIS iteration. Rather than asking the SMT solver to provide another counterexample, which might be computationally expensive, the authors in [AAE+21] propose the following heuristic approach, based on the assumption that *samples around a counterexample are equally likely to invalidate the certificate conditions* by continuity. Then, they randomly sample new points around the counterexample provided by the verifier, and aggregate them with the training sample set for the next CEGIS round.

5.4.2 Post-processing

The verification step described in the previous section can be quite challenging for different reasons. First, we usually consider a small training sample set to hasten the computation time of the neural network. The drawback of this, however, is that the network tends to overfit such that the Lyapunov inequalities are only satisfied for the scaled system on the sample points while they are strongly violated elsewhere. Moreover, we are constrained by the practical limitations of the SMT solvers, as they suffer poor scalability. As a result, in some cases, the SMT solver might return a counterexample at each iteration and the CEGIS loop might end up without valid approximation on the JSR.

These limitations have motivated the design of a norm induced by $NN(\cdot)$ for which we can easily compute the corresponding JSR approximation.

Definition 5.12 (Polytopic norm induced by a ReLU neural network)**.** Let $NN : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ denote the function represented by a ReLU neural network trained on the sample set $\mathcal{S} := \{x_k\}$. We define the *norm induced by* NN as the unique homogeneous function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ whose 1-sublevel set is given by

$$\overline{\mathcal{B}_{V(\cdot)}} := \operatorname{conv}\left(\left\{v_i := \frac{x_i}{NN(x_i)} \;\middle|\; x_i \in \mathcal{S}\right\}\right).$$

In this manner, the functions $NN(\cdot)$ and $V(\cdot)$ coincide on the sample points, i.e. $NN(x_k) = V(x_k)$ for any point x_k in the sample set $\mathcal{S}$. Elsewhere, computing the norm $V(x)$ of any point $x \in \mathbb{R}^n$ amounts to solving the following Linear Program:

$$\begin{aligned} \lambda^* = \max \quad & \lambda \\ \text{s.t.} \quad & \lambda x \in \overline{\mathcal{B}_{V(\cdot)}} \end{aligned}$$

and $V(x) = 1/\lambda^*$. In turn, the JSR approximation provided by V can be computed as the maximum induced matrix norm

$$\hat{\rho}_V(\mathcal{A}) := \max_{A \in \mathcal{A}} V(A),$$

where $V(A)$ is the maximal norm of Av_k over all the vertices v_k of the unit ball $\overline{\mathcal{B}_{V(\cdot)}}$.

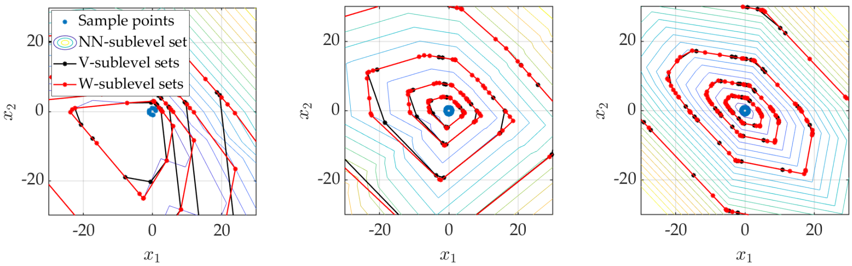
The training not only optimizes the values $NN(x_k)$, but also the image of the samples points, i.e. $NN(A_i x_k)$ for $i = 1, \ldots, M$. Thus, it seems meaningful to also include these points in the definition of the convex hull. We suggest an alternative to the post-processed norm in Definition 3.15, which is expected to lead to better JSR approximations but potentially at the cost of longer computing times.

Definition 5.13 (Extension of the polytopic norm induced by a ReLU neural network)**.** Let $NN : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ denote the function represented by a ReLU neural network trained on the sample set $\mathcal{S} := \{x_k\}$. We define the *extension of norm induced by NN* as the unique homogeneous function $W : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ whose 1-sublevel set is given by

$$\overline{\mathcal{B}_{W(\cdot)}} := \operatorname{conv}\left(\left\{ w_{i,j} := \frac{A_j x_i}{NN(A_j x_i)} \;\middle|\; x_i \in \mathcal{S}, j = 1, \ldots, M \right\} \cup \mathcal{B}_{V(\cdot)} \right).$$

In this case, the functions $NN(\cdot)$ and $W(\cdot)$ have equal value on the sample points and on the images of the sample points. The computation of the norm $W(x)$ of $x \in \mathbb{R}^n$ and $W(A)$ where $A \in \mathbb{R}^{n \times n}$ is similar to the previous case. We expect that the function W provides better upper approximations of the JSR than V, but it requires in return more computation time since it involves more vertices.

Example 5.14. We consider the linear switched system $\mathcal{A}_2 := \{A_1, A_2\} \subset \mathbb{R}^{2\times 2}$ defined in (5.2). As a reminder, $\rho(\mathcal{A}) = 8.6881$, the ellipsoidal ap-



(a) 20 sample points. **(b)** 50 sample points. **(c)** 100 sample points.

Fig. 5.8 Comparison of the sublevel sets of $NN(\cdot)$, $V(\cdot)$ (in black) and $W(\cdot)$ (in red) obtained from a ReLU neural network with 2 layers, 5 neurons and different sample sizes in Example 5.14.

proximation $\rho_Q(\mathcal{A}_2)$ is 9.5868 and the SOS approximation of degree 4, i.e. $\rho_{SOS,4}(\mathcal{A}_2)$, is 8.7203. We consider different sizes for the sample set: 20, 50 or 100 sample points. For each sample, we keep the last approximation of the network NN, and the polytopic norms induced $V(\cdot)$ and $W(\cdot)$.

We consider a ReLU neural network with 2 hidden layers of 5 or 10 neurons which is trained during 2000 iterations. The results are summarized in Table 5.4 and Figure 5.8. One can easily notice that when we consider few sample points (20 for instance), the network ends up overfitting and therefore provides an invalid upper approximation of the JSR. The V and W-

	2 layers of 5 neurons			**2 layers of 10 neurons**		
	Number of sample points			**Number of sample points**		
	20	50	100	20	50	100
$NN(\cdot)$	8.2240	8.9546	8.7347	4.5380	8.6441	8.7200
$V(\cdot)$	10.0729 (0.04 sec.)	9.1380 (0.09 sec.)	9.0226 (0.19 sec.)	25.3548 (0.02 sec.)	11.4275 (0.06 sec.)	9.0827 (0.12 sec.)
$W(\cdot)$	10.0038 (0.08 sec.)	9.5777 (0.25 sec.)	8.7584 (0.95 sec.)	82.0506 (0.07 sec.)	10.7237 (0.21 sec.)	8.8562 (0.58 sec.)

Table 5.4 Approximation of the JSR of system (8.23) using a ReLU neural network with 2 layers, 5 or 10 neurons and different sample sizes and the polytopic norms induced $V(\cdot)$ and $W(\cdot)$. The invalid upper bounds are highlighted in color.

approximations provide valid results, but much larger. See for instance the experiment with 2 hidden layers of 10 neurons and only 20 sample points. The network provides an invalid upper approximation of 4.5380, while the true JSR is 8.6881. Accordingly, the $V(\cdot)$ and $W(\cdot)$ approximations both explode. This can also be observed in Figure 5.8; with only 20 sample points, the network overfits and encodes a function with sublevel sets which are not convex, but not surprisingly, when we increase the number of sample points, the network overfits less and the sublevel sets of $NN(\cdot)$, $V(\cdot)$ and $W(\cdot)$ are becoming increasingly similar. While with only 20 sample points, one can clearly see that they differ from each other. △

Note that in these experiments, the W-approximation is almost always better (i.e. smaller but still valid) than the V-approximation, at the cost of more computation time.

Now, let us consider a switched system in higher dimension for which the computation of the JSR approximation provided by the induced polytopic norms will take more time.

Example 5.15. We consider the switched system $\mathcal{A}_8$ in dimension 8 with 8 different modes, defined by the matrices (5.3). As a reminder, one can prove that $\rho(\mathcal{A}_8) = 1$, the ellipsoidal approximation is 2.4286 (in 2.5 seconds), while the SOS approximation is 1.0006 provided in 258 seconds.

We use a ReLU neural network with a single layer of 10 or 15 neurons that we train during 2000 iterations. We use different sample sizes, and for

	1 layer of 10 neurons			**1 layer of 15 neurons**		
	Number of sample points			**Number of sample points**		
	100	250	500	100	250	500
$NN(\cdot)$	0.9354	1.2016	1.3682	0.8385	0.9829	1.1283
$V(\cdot)$	6.0421 (0.96 sec.)	4.3086 (3.96 sec.)	3.5660 (14.1 sec.)	8.2080 (1.02 sec.)	4.4535 (4.12 sec.)	3.1007 (14.6 sec.)
$W(\cdot)$	2.1311 (31.8 sec.)	1.8909 (179 sec.)	1.5960 (809 sec.)	2.3200 (35.5 sec.)	2.3965 (199 sec.)	1.2176 (1083 sec.)

Table 5.5 Approximation of the JSR of system (5.3) using a ReLU neural network with 1 layer, 10 or 15 neurons and different sample sizes and the polytopic norms induced $V(\cdot)$ and $W(\cdot)$. The invalid upper bounds are highlighted in color.

each configuration, we keep the last approximations of the network $NN(\cdot)$ and we compute both approximations provided by the induced polytopic norms $V(\cdot)$ and $W(\cdot)$. The results are summarized in Table 5.5. The observations are similar to Example 5.14 in dimension 2, but the computation time is much larger. △

According to the results in Tables 5.4 and 5.5, it seems that the approximations provided by the induced polytopic norms increase when the network overfits. This observation leads us to ask the following question: can we use those post-processed approximations to detect overfitting? The following example investigates this question.

Example 5.16. In order to answer this question, we first consider a ReLU neural network with 2 layers of 10 neurons in each layer, 20 or 100 sample points, that we train during 2000 iterations to estimate the JSR of system (5.2). Figure 5.9 illustrates the evolution of the JSR approximations provided by the network and the polytopic norm induced V. One can clearly observe two distinct behaviours: with 100 sample points, the network does not overfit, and both approximations follow the same trend (the maximal difference between the two approximations is around 0.4) while

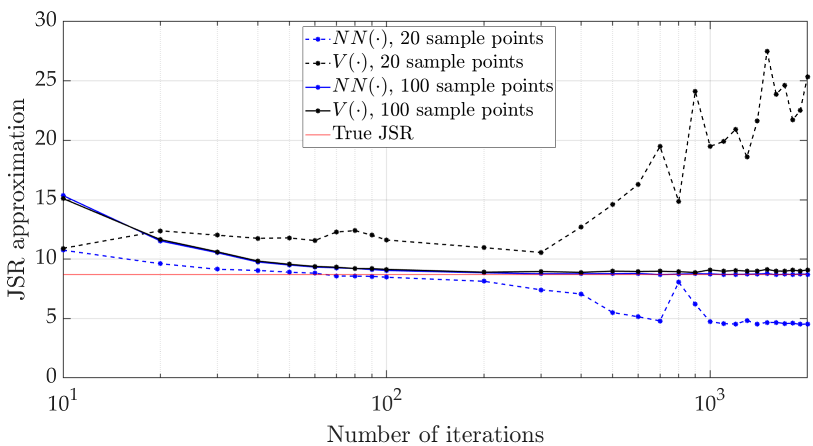


Fig. 5.9 Evolution through the learning iterations of the JSR approximation of system (5.2) provided by a ReLU neural network $NN(\cdot)$ (in blue) with 2 hidden layers and 10 neurons in each layer and 20 (the dashed line) or 100 (the full line) sample points. For comparison, after every 100 iterations, the post-processed approximation provided by $V(\cdot)$ (in black) is computed.

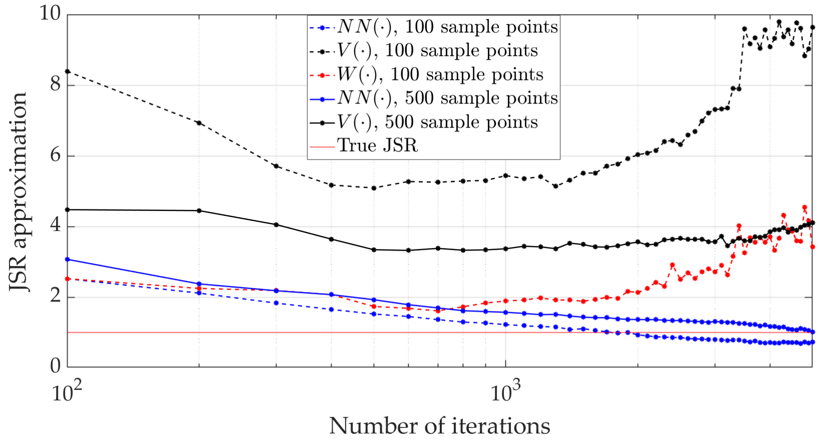


Fig. 5.10 Evolution through the learning iterations of the JSR approximation of system (5.3) provided by a ReLU neural network $NN(\cdot)$ (in blue) with 1 layer and 10 neurons in each layer and 100 (the dashed line) or 500 (the full line) sample points. For comparison, after every 100 iterations, the post-processed approximations provided by $V(\cdot)$ (in black) and $W(\cdot)$ (in red) are computed.

with only 20 sample points, both approximations start to differ after 20 iterations, and they will almost never get closer.

Similarly, we consider a ReLU neural network with 1 layer of 10 neurons with 100 or 500 sample points to provide JSR approximations of system (5.3). After every 100 iterations, we keep the last value of the loss function, and we compute the approximations provided by the induced polytopic norms $V(\cdot)$ and $W(\cdot)$ (only for 100 sample points). Figure 5.10 shows the evolution of these quantities. In this case, the "elbow shape" is not so obvious as in Figure 5.9. However, one can still observe that for 100 sample points, both curves for V and W start to decrease, then become almost constant and then increase, which would indicate that the network overfits. The same situation occurs for 500 sample points. △

Although the computation of these induced polytopic norms might help detect overfitting, this would require too much computation time, especially in high dimensions.

Crucially, the computation of the induced norms V and W allows us to provide correct upper bounds on the JSR without the usage of SMT-solving. SMT-problems are in general NP-hard, so we provide a fail-safe in case of a timeout failure in the verification step.

5.4.3 Experimental evaluation

In this section, we test and compare both methods introduced in Sections 5.4.1 and 5.4.2 on a variety of switched systems of varying complexity, by increasing the dimension and the number of matrices.

We consider the switched system defined in Equation (5.2) and denoted hereafter by $\mathcal{A}_1$, the systems $\mathcal{A}_2$ introduced in [DJ22b, Example 3], $\mathcal{A}_3$ in [GWZ05, Example 6.4] $\mathcal{A}_4$ in [AJ19, Example 2] and $\mathcal{A}_5$ in [GZ08, Section 7]. For each of them, we use the CEGIS architecture described in Section 5.4.1 and we compute the post-processed approximations developed in Section 5.4.2. The numerical experiments are summarized in Table 5.6. For each switched system, we provide two different results: the first line outlines the scheme with the most accurate approximation of the JSR, while the second line provides the sample execution with a good precision-computation time trade off.

Various trends can be identified from this numerical experience, in line with theoretical results. Overall, the results show that for nearly all examples, our neural approach finds a better approximation than the usual quadratic approach. However, the structure of the network and the parameters differ for each system. For instance, the structure of the network grows with the dimension to achieve a similar precision. In particular, the higher the dimension of the system, the wider the neural network, as expected by Theorem 5.1. Moreover, the sampling size increases with the dimension and the number of dynamics as well. As a result, the computation time also increases as reflected in Table 5.6.

In practice, we have noticed that, as the dimension increases, the CEGIS loop never ends with a valid JSR approximation. We expect this to be explained by the number of points required to cover the unit sphere with a fixed percentage. We therefore believe that, as the dimension increases, more and more areas of the state space escape the network. Then, the Lyapunov inequalities are not satisfied over the whole state space, and the verifier fails. To circumvent this problem, we have "relaxed" the Lyapunov inequalities by increasing and/or rounding up the JSR approximation provided by the network. In this case, the CEGIS loop is more likely to stop on a valid JSR approximation.

Regarding the approximations provided by the post-processed norms, W almost always provides a slightly better approximation than V although one must bear in mind that W is more computational demanding than V.

System					Neural approximation						Induced norms	
$\mathcal{A}$	n	M	$\rho(\mathcal{A})$	$\rho_Q(\mathcal{A})$	$\hat{\rho}_{NN(\theta),\mathcal{S}}(\mathcal{A})$	θ	Iters	$\lvert\mathcal{S}\rvert$	Loops	Time	$\hat{\rho}_V(\mathcal{A})$	$\hat{\rho}_W(\mathcal{A})$
$\mathcal{A}_1$	2	2	8.6881	9.5868	8.7090	[5]	400	100	10	19 [16]	8.7141	8.7107
					8.7298	[15]	400	300	0	7 [2]	8.7346	8.7298
$\mathcal{A}_2$	2	3	4.5340	4.7794	4.5367	[8]	200	100	5	12 [6]	4.5402	4.5439
					4.5463	[10]	200	500	1	8 [2]	4.5471	4.5478
$\mathcal{A}_3$	2	4	1.000	1.000	1.0014	[7]	300	500	10	23 [20]	1.0007	1.0007
					1.0016	[8]	300	500	0	3 [2]	1.0004	1.0004
$\mathcal{A}_4$	3	2	0.9506	1.0171	$0.99^\star$	[10]	500	800	6	46 [17]	1.0063	1.0049
					$1.01^\star$	[9]	400	900	2	17 [6]	1.0471	0.9989
$\mathcal{A}_5$	3	3	0.9194	0.9411	$0.94^\star$	[10]	450	1000	4	223 [15]	0.9526	0.9394
					$0.95^\star$	[10]	200	1000	3	168 [5]	0.9508	0.9438

Table 5.6 Evaluation of the numerical benchmarks of Section 5.4.3. For each system, the first line (white background) outlines the scheme with the best approximation of the JSR while the second line (gray background) outlines a sample execution with a good precision-computation time trade off. n is the dimension of the system, $\theta := [x_1, \ldots, x_k]$ defines the structure of the network, meaning that there is x_i neurons in the ith layer, Iters is the number of iterations during the training of the network for each CEGIS iteration, $|\mathcal{S}|$ is the initial number of sample points, Loop is the number of CEGIS iterations and Time is the computation time for the whole CEGIS algorithm with the total learning time in brackets. The symbol $\star$ means that the JSR approximation of the network has been rounded up to 0.01 accuracy and increased by 0.01 before the validation check by the SMT solver.

However, both approximations are similar to the approximation provided by the network. This promotes the use of post processing in case of failure of the CEGIS loop.

5.5 Summary and further research directions

Motivated by recent developments in neural Lyapunov techniques, we have introduced for it a benchmark application and a theoretical framework, for the particular case of switched systems.

Summary of Chapter 5

This chapter introduces an automatic and sound algorithm to study the stability of linear switched systems by approximating the joint spectral radius of the corresponding set of matrices.

Section 5.1: Bounds on the width and depth of the neural network
We have shown that one can determine theoretical bounds for the accuracy of neural Lyapunov functions as a function of the parameters of the network. The depth of the network only depends on the dimension, while the width is a function of the dimension and the precision.

Section 5.2: Experimental evaluation
From the empirical point of view, we have shown that in practice as well, the approach is competitive with SDP-based techniques while our neural networks were trained on simple personal computers, which leaves an important room for improvement.

Section 5.3: Improvement techniques
We have emphasized the problem of overfitting, and proposed avenues for mitigating it. Namely, we have considered the regularization and the incremental learning.

Section 5.4: Computation of valid neural-based JSR approximations
The CEGIS architecture relies on two elements: a ReLU neural network, and an SMT solver. We therefore benefit from the advantages of these two components: notably the flexibility of neural

networks and the soundness of SMT solvers. We also suffer from their disadvantages, like the poor scalability of SMT solvers. However, we introduce post-processed norms V and W to address this problem, and ensure a valid approximation of the JSR. Our algorithm has shown promising results on several examples, nearly always beating the usual quadratic approximation, but further comparison is required with more advanced methods, and in higher dimensions.

We see much possible further work, among which confirming the experiments at a more general level (different network architectures, different benchmark examples), potentially improving the learning algorithm (in particular, the loss function), and pushing further the approaches for mitigating overfitting.

In Section 5.4, we address in particular the problems of verification of the output of the neural network. So far, we have followed the method which was initially used in FOSSIL, that is the use of SMT solvers. Nevertheless, they suffer poor scalability and many alternative methods could be used. [BBLJ23] provides a comparison of the state-of-the-art neural network verifications tools. According to this competition, the current best methods are *linear bound propagation methods*, and in particular the α, β-*CROWN* algorithms used in [WIZ+24, YDS+24].

PART III
Template-dependent comparison of Path-Complete Lyapunov functions

6

Path-complete Lyapunov functions and their comparison

ONE of the possible ways to assess the stability of switched systems is to use *Lyapunov theory*, and *common Lyapunov functions* (CLFs) in particular, as developed in previous sections. This approach consists in finding a single positive definite function that decreases along any dynamics of the system. The *template*, i.e. the set in which the candidate CLF is searched, has evolved over time and became more and more complex. One popular approach considering quadratic functions has been generalized, for example, by considering sum-of-square polynomials [AJ18a], polyhedral Lyapunov functions [BM99] and then the max-min of quadratics [GTHL06]. Although the existence of a CLF is a necessary and sufficient condition for stability (as recalled in Theorem 1.12), it is largely offset by the computing complexity required by the "search" for this Lyapunov function. See for instance, the discussions provided in [AJ18a].

Therefore, the *Multiple Lyapunov functions* approach stands out as a promising alternative, as introduced in [Bra98], [GHT06] and [Lib03] for instance. This approach aims to find (rather than a single function) a *set of Lyapunov functions* whose *joint decrease behaviour* provides a stability certificate. More recently (for example in [AJPR14, PAAJ19, Pep19, AGMG22, DPA22, CGPS21]), multiple Lyapunov framework has been extended to the case in which inequalities involving the candidate Lyapunov functions are encoded in labeled and directed graphs. For studying stability in the arbitrary switching signals case, the graph defining the structure of the inequalities has to recognize (in an automata theory sense) every possible switching sequence, in which case it is usually called a *path-complete graph*. A connection between this graph framework (also called *path-complete Lyapunov functions (PCLF) setting*) and the general multiple Lyapunov functions approach is provided in [JAPR17]: a set of inequalities involving multiple Lyapunov functions is a valid certificate for stability if and only if the corresponding graph describing the inequalities is path-complete. Formally, the PCLF framework involves both combinatorial and algebraic components: first, a path-complete graph that describes the set of Lyapunov inequalities, and then a set of candidate Lyapunov functions, called a *template*, among which a solution is sought. More recently, path-complete techniques have been proven to be effective not only for the stability problem for switched systems, but also in different settings, as for example constrained switched systems in [PEDJ15], stabilization of switched systems in [LDH20], and continuous-time switched systems in [DPA22].

The *path-complete Lyapunov functions* framework provides new guidelines for constructing stability certificates but it opens new questions and challenges, both from a theoretical and computational point of view. Indeed, the theory allows to use different graphs and different templates of functions, and thus provides a wide range of possibilities. However, it is not well understood yet why one of these algorithms provides less conservative stability certificates than another, which has led to the problem of *comparing* different path-complete graphs. More precisely, a graph is said to be “better” than another one when its decay rate approximation capabilities surpass those of the other graph (in a sense that we will clarify in Definition 6.21 below).

After a brief introduction on multiple Lyapunov functions, we review and summarize in this chapter current knowledge on the path-complete Lyapunov formalism and the comparison of complete-path graphs. In particular, we formally define three levels of of ordering of graphs, namely the *template and dynamics-dependent* ordering relation, the *template-dependent* ordering relation, and the *general* ordering relation which holds regardless of the dynamics and the template. For the latter ordering relation, we recall the simulation-based characterization in [PJ19], as well as its proof that will serve as the basis for the template-dependent order characterizations in Chapter 8.

6.1 Introduction to multiple Lyapunov functions

It is clear from Section 1.1.2 that Lyapunov theory is a powerful methodology to provide stability certificates for switched systems. In particular Theorem 1.12 states that the existence of a *common* Lyapunov function is a necessary and sufficient condition for stability. Nevertheless, the search for such a function in practice is limited to specific templates for which numerical algorithms have been developed, as illustrated in Section 2.2. Whatever the template used, it provides conservative Lyapunov stability certificates. The template of quadratic Lyapunov functions is a good illustration. Indeed, there exist several examples [LM99, LA09] of stable linear switched systems for which there does not exist a *common quadratic* Lyapunov function. This has forced researchers to consider more and more complex candidate Lyapunov functions. As example, [PJ08, Example 2.8] has motivated the introduction of SOS polynomials as template. However, no matter how complex the template, it is often possible to build a stable switched system which does not admit a common Lyapunov function within this template. For instance, [AJ14, Theorem 1] reveals that, for any $d \in \mathbb{N}$, there exists a switched system which does not admit a polynomial Lyapunov function of degree d, or a polytopic Lyapunov function with d facets, or a piecewise quadratic Lyapunov function with d pieces.

In order to alleviate this conservativeness, researchers have started to rely on more structured sets of Lyapunov inequalities which involve several candidate Lyapunov pieces [Bra98, GHT06, Lib03], rather than looking for a single, increasingly complex Lyapunov function. In this case, the decrease along trajectories with respect to any switching signal is guaranteed by the *joint behaviour* of the Lyapunov pieces and not by their individual behaviour. Therefore, the so-called *multiple Lyapunov functions* (MLFs for short) turned out to be a promising alternative to increasingly complex template. Moreover, with the multiple Lyapunov formalism, we hope to be able to moderate the numerical cost by searching for more functions but in simpler templates, i.e. templates that require less numerical effort.

To start with, let us consider the example of a multiple Lyapunov function for a switched system with 2 modes which has been first introduced in [DB01], and further studied in [PAAJ19]. The authors provide a specific switched system for which the quadratic template fails to produce a *common* Lyapunov function, but provides a *multiple* Lyapunov function.

Example 6.1. Consider any n-dimensional linear switched system $\mathcal{A} := \{A_1, A_2\} \subseteq \mathbb{R}^{n\times n}$ with $n \in \mathbb{N}$. The multiple Lyapunov stability certificate proposed in [DB01] involves two candidate quadratic Lyapunov functions $V_a, V_b : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ such that the following Lyapunov inequalities are satisfied, namely $\forall x \in \mathbb{R}^n$:

$$\begin{cases} V_a(A_1x) \leq V_a(x), & (6.1a) \\ V_b(A_1x) \leq V_a(x), & (6.1b) \\ V_a(A_2x) \leq V_b(x), & (6.1c) \\ V_b(A_2x) \leq V_b(x). & (6.1d) \end{cases}$$

In this case, the multiple Lyapunov function is time-dependent and can be defined as

$$V(x,k) := \begin{cases} V_a(x) & \text{if } \sigma(k) = 1, \\ V_b(x) & \text{if } \sigma(k) = 2. \end{cases} \tag{6.2}$$

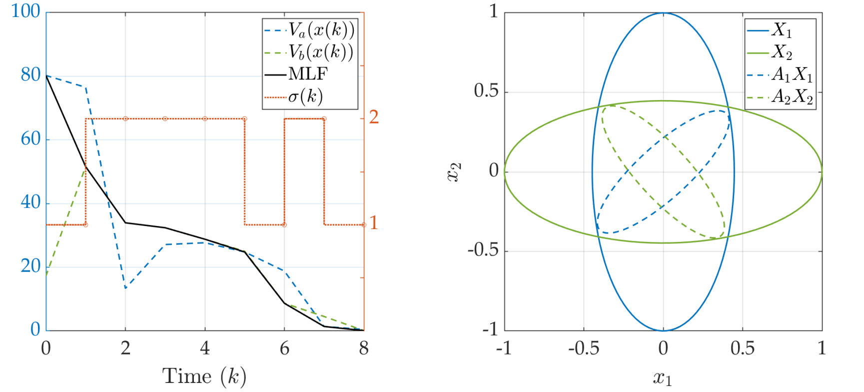
Moreover, [PAAJ19, Example III.4] provides the linear switched system in dimension 2 defined by the matrices

$$A_1 := \alpha \begin{bmatrix} 1.3 & 0 \\ 1 & 0.3 \end{bmatrix} \text{ and } A_2 := \alpha \begin{bmatrix} -0.3 & 1 \\ 0 & -1.3 \end{bmatrix}, \tag{6.3}$$

with $\alpha = (1.4)^{-1}$. This system does not admit a common quadratic Lyapunov function but there exist two quadratic Lyapunov functions V_a and V_b satisfying the multiple Lyapunov stability criterion defined by Equation (6.1). In particular,

$$\begin{cases} V_a(x_1, x_2) := 5{x_1}^2 + {x_2}^2, \\ V_b(x_1, x_2) := {x_1}^2 + 5{x_2}^2, \end{cases} \tag{6.4}$$

satisfy the Lyapunov inequalities. Figure 6.1 shows the evolution of each piece V_a and V_b along the trajectory starting at $x(0) := [4, -1/2]^\top$ and following the switching sequence $\sigma := 122221212$. As expected, one can observe that the functions V_a and V_b do not decrease at each time step. However, the joint behaviour defined by the multiple Lyapunov function in Equation (6.2) decreases at each time along the trajectory, as illustrated in Figure 6.1a. We can formally verify that the function decreases along the trajectory. Indeed,



(a) Evolution with time k of the switching signal σ (on the right axis), Lyapunov pieces V_a and V_b in (6.4), and the multiple Lyapunov function in (6.2) (on the left axis) along the trajectory starting at $x(0) = [4, -1/2]^\top$ and following the switching sequence $\sigma := 122221212$.

(b) Illustration of the 1-sublevel sets of V_a and V_b respectively denoted by X_1 and X_2, and the images A_1X_1 and A_2X_2. Equation (6.1a) implies that $A_1X_1 \subseteq X_1$. Similarly, Equation (6.1d) implies that $A_2X_2 \subseteq X_2$.

Fig. 6.1 Illustration of the multiple Lyapunov function in (6.4) for system (6.3) in Example 6.1.

$$
\begin{aligned}
V(x(8),8) &\overset{(6.2)}{:=} V_b(x(8)) \overset{\sigma(7)=1}{=} V_b(A_1x(7)),\\
&\overset{(6.1b)}{\leq} V_a(x(7)) \overset{(6.2)}{:=} V(x(7),7) \overset{\sigma(6)=2}{=} V_a(A_2x(6)),\\
&\overset{(6.1c)}{\leq} V_b(x(6)) \overset{(6.2)}{:=} V(x(6),6) \overset{\sigma(5)=1}{=} V_b(A_1x(5)),\\
&\overset{(6.1b)}{\leq} V_a(x(5)) \overset{(6.2)}{:=} V(x(5),5) \overset{\sigma(4)=2}{=} V_a(A_2x(4)),\\
&\overset{(6.1c)}{\leq} V_b(x(4)) \overset{(6.2)}{:=} V(x(4),4) \overset{\sigma(3)=2}{=} V_b(A_2x(3)),\\
&\overset{(6.1d)}{\leq} V_b(x(3)) \overset{(6.2)}{:=} V(x(3),3) \overset{\sigma(2)=2}{=} V_b(A_2x(2)),\\
&\overset{(6.1d)}{\leq} V_b(x(2)) \overset{(6.2)}{:=} V(x(2),2) \overset{\sigma(1)=2}{=} V_b(A_2x(1)),\\
&\overset{(6.1d)}{\leq} V_b(x(1)) \overset{(6.2)}{:=} V(x(1),1) \overset{\sigma(0)=1}{=} V_b(A_1x(0)),\\
&\overset{(6.1b)}{\leq} V_a(x(0)) \overset{(6.2)}{:=} V(x(0),0),
\end{aligned}
$$

where the definition of the MLF in Equation (6.1) and the Lyapunov inequalities in Equation (6.1) are used when appropriate. △

Just as for common Lyapunov functions (see Section 2.2), the choice of the template for multiple Lyapunov functions is crucial and partially affects the conservatism of the corresponding stability certificate. In particular, we will see in the following chapters that the *closure properties* of a template, i.e. the operations which preserve the template regardless of the dimension, play a key role in the conservatism-based comparison of graph-based stability certificates.

6.2 Graph-based Lyapunov functions

In this section, we formally introduce the notion of *path-complete Lyapunov function* which generalizes the multiple Lyapunov framework by encoding the Lyapunov inequalities with specific directed and labeled graphs. This new formalism relies on two structural components, namely the *template* already introduced in Section 1.3 and the *path-complete graph*. The next section addresses the path-completeness while the second one gathers both parameters and formally defines the concept of path-complete Lyapunov function. Although this formalism is equally suitable for linear or nonlinear switched systems, the last section focuses on switched linear systems. We consider in particular the JSR approximation using the path-complete Lyapunov formalism.

6.2.1 Path-complete graphs

The path-complete Lyapunov framework mainly relies on the notion of *path-completeness* of a graph. The principle of this framework is to use a finite graph to encode all the possible switching sequences. A convenient way to do it is to consider a directed and labeled graph.

Definition 6.2 (Directed and labeled graph)**.** Given $M \in \mathbb{N}$, a *directed and labeled graph* on the alphabet $\langle M \rangle$ is a couple $\mathcal{G} = (S, E)$ where

- S is the finite set of *nodes* of the graph, and
- $E \subseteq S \times S \times \langle M \rangle^+$ is the set of directed and labeled *edges*.

A *directed and labeled edge* is a tuple $(s, d, w) \in E$ where s is called the source or the *starting node*, d is the destination or the *ending node* and the word $w := i_1 \dots i_k$ is the *label* of the edge.

Definition 6.3 (Dual graph)**.** Given a directed and labeled graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, the *dual graph* of $\mathcal{G}$, denoted by $\mathcal{G}^\top = (S, E^\top)$ is obtained by reversing the direction of each edge and the order of the word, i.e. $(s, d, w) \in E^\top$ if and only if $(d, s, w^\top) \in E$.

Given a directed and labeled graph $\mathcal{G} = (S, E)$, we can define a *path* in $\mathcal{G}$ as a sequence of consecutive edges, i.e.

$$p \ := \ \{(s_i, s_{i+1}, w_i) \in E \mid i = 1, 2, \dots\}.$$

The *length* of the path p is defined as the number of edges that it contains. Given an integer $k \in \mathbb{N}$, we denote by $\mathscr{P}^k(\mathcal{G})$ the set of all the paths in $\mathcal{G}$ of length k. By extension, we define $\mathscr{P}(\mathcal{G})$ as the set of all the paths in $\mathcal{G}$, i.e.

$$\mathscr{P}(\mathcal{G}) \ := \ \bigcup_{k=1}^{\infty} \mathscr{P}^k(\mathcal{G}).$$

Any path p also generates a *word*, denoted by $w(p)$, which is the concatenation of the labels on its edges, i.e. $w(p) = w_1 w_2 \cdots \in \langle M \rangle^+$.

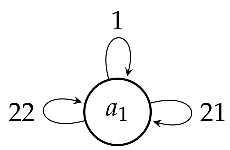
Definition 6.4 (Path-complete graph)**.** Given a directed and labeled graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, with $M \in \mathbb{N}$, the graph $\mathcal{G}$ is *path-complete* if for any finite word $\omega := i_1 i_2 \dots i_k \in \langle M \rangle^+$ of any length $k \in \mathbb{N}$, there exists a directed path $p \in \mathscr{P}^k(\mathcal{G})$ in the graph whose word $w(p)$ contains ω as a subword.

Note that the notion of path-completeness is known as *universality* in automata theory. Some existing algorithms can check whether an automata can generate any finite word but this problem is known to be PSPACE-complete, see [JAPR17].

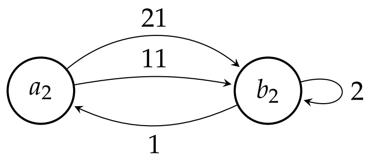
The most trivial path-complete graph is probably the *common Lyapunov function graph*, denoted by $\mathcal{G}_0$, with one node and as many loops as the number of modes M, i.e. $\mathcal{G}_0 := (\{a\}, \{(a, a, i) \mid i \in \langle M \rangle\})$. There exist many other possible graphs, as illustrated in the following example.

Example 6.5 (Example of path-complete graphs). Consider the alphabet $\langle 2 \rangle$ and the two graphs $\mathcal{G}_1 = (S_1, E_1)$ and $\mathcal{G}_2 = (S_2, E_2)$ in Figure 6.2. One can prove that $\mathcal{G}_1$ is path-complete over the alphabet while $\mathcal{G}_2$ is not.

Since the graph $\mathcal{G}_1$ admits a single node, the reasoning is quite intuitive. Given any finite word $\omega := i_1 \dots i_k$ on the alphabet $\langle 2 \rangle$, we proceed in



(a) $\mathcal{G}_1 = (S_1, E_1)$, an example of a path-complete graph on the alphabet $\langle 2 \rangle$ with labels of different lengths.



(b) $\mathcal{G}_2 = (S_2, E_2)$, an example of a graph which is *not* path-complete on the alphabet $\langle 2 \rangle$.

Fig. 6.2 Examples of two graphs $\mathcal{G}_1$ ans $\mathcal{G}_2$, one which is path-complete and one which is not.

an iterative way: if the first character i_1 is 1, we take the loop $(a_1, a_1, 1)$. Otherwise, if i_1 is 2, we either take $(a_1, a_1, 22)$ or $(a_1, a_1, 21)$ according to the second following symbol. Following this procedure, we will always be able to generate any finite word with $\mathcal{G}_1$.

Regarding $\mathcal{G}_2$, let us consider the word $\omega := 221122$. Indeed, one can easily see that the first sequence of 2's can only be achieved by the loop $(b_2, b_2, 2)$. From there, we can only leave the node through the edge $(b_2, a_2, 1)$. However, none of the outgoing edges from a_2 can be used to complete the sequence w. This argumentation can be generalized to any sequence with an even number of consecutive 2's followed by an even number consecutive of 1's, itself followed by an even number of consecutive 2's and so forth. △

Moreover, it is easy to see [AJPR14, Theorem 3.2] that the duality preserves the path-completeness.

Proposition 6.6. *Consider a directed and labeled graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$. The graph $\mathcal{G}$ is path-complete if and only if the dual graph $\mathcal{G}^\top$ is path-complete.*

Proof. Consider a path-complete graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$ with $M \in \mathbb{N}$. We consider a finite word $w := w_1 \dots w_k \in \langle M \rangle^k$ of length $k \in \mathbb{N}$. By definition of the path-completeness, there exists a path

$$p := \{(s_i, s_{i+1}, w_{k+1-i}) \in E \mid i = 1, \dots, k\}$$

in $\mathcal{G}$ whose word is the reverse of w, i.e $w^\top := w_k \dots w_1$. By duality in Definition 6.3, the reversed path $p' := \{(s_{i+1}, s_i, w_i) \in E \mid i = 1, \dots, k\} \subseteq E^\top$ and $w(p') = w$. This completes the proof. □

In Definition 6.2 and in Definition 6.4 consequently, we assume that the labels belong to the Kleene closure of the alphabet and can therefore have different lengths. However, it is always possible to derive a directed and labeled graph with labels of length 1 via the *expanded graph* introduced in [AJPR14, Definition 2.1].

Definition 6.7 (Expanded graph)**.** Given a directed and labeled graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$ with label of length larger than 1, the *expanded graph*, denoted by $\mathcal{G}^e = (S^e, E^e)$ is defined as the outcome of the following procedure. For every edge $e = (s, d, w) \in E$ such that the label $w := i_1 i_2 \dots i_k$ is of length $k > 1$, we add $k - 1$ intermediate nodes $s_1, \dots, s_{k-1}$. We replace the initial edge with label of multiple length with k new edges (s_j, s_{j+1}, i_{j+1}) of label of length 1, where $s_0 := s$.

As example, Figure 6.3 illustrates the expanded form of $\mathcal{G}_1$ in Figure 6.2a. Using the expanded graph, we assume that we will only consider path-complete graphs with labels of length one in the rest of this report. The conservatism of this assumption will be discussed in Chapter 7. Indeed, Proposition 7.70 provides sufficient conditions for one graph to be equivalent to its expanded form. For instance, a graph and its expanded form are equivalent for linear switched systems and Lyapunov norms.

Assumption 6.8. *The path-complete graphs considered herein only have labels of length one.*

Let us now introduce a property of graphs which implies the path-completeness.

Definition 6.9 (Complete and co-complete graphs)**.** A directed and labeled graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$ is *complete* if for all $a \in S$, for all $i \in \langle M \rangle$, there exists at least one node $b \in S$ such that the edge $(a, b, i) \in E$. The graph is *co-complete* if for all $b \in S$, for all $i \in \langle M \rangle$, there exists at least one node $a \in S$ such that the edge $(a, b, i) \in E$.

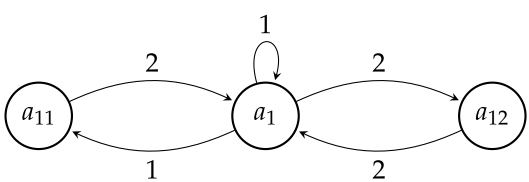


Fig. 6.3 $\mathcal{G}_1{}^e = (S_1{}^e, E_1{}^e)$, the path-complete graph $\mathcal{G}_1$ in expanded form.

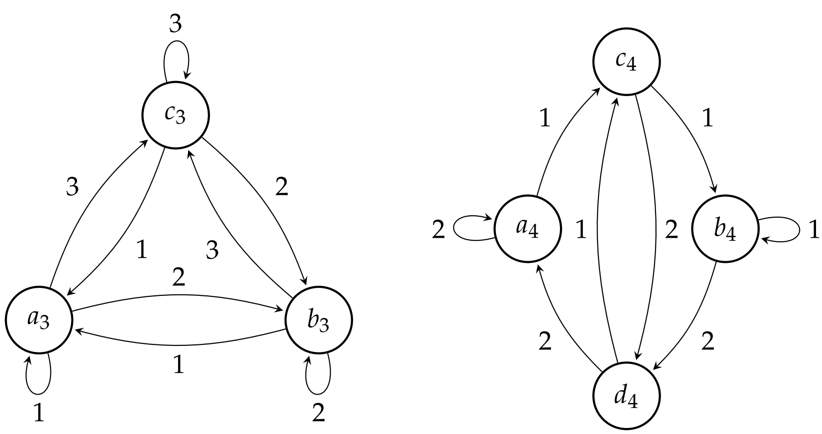
It follows that the dual of a complete graph is co-complete and reversely. Moreover, any complete or co-complete graph is path-complete, see [AJPR14, Proposition 3.3].

Regarding complete graphs, the class of *De Bruijn graphs* is a notable example. They were introduced in [de 46] and were initially used to represent common past of finite words of an alphabet. Nowadays they are used in various fields, such as grid network or bioinformatics [CLJ$^+$15, CLM16]. Let us formally define them.

Definition 6.10 (De Bruijn Graphs)**.** Given $M, K \in \mathbb{N}$, the *(primal) De Bruijn graph of order* $K-1$ *(on the alphabet* $\langle M \rangle$*)* denoted by $\mathcal{G}_{db}^{K} = (S, E)$ is defined as follows: $S := \langle M \rangle^{K-1}$ and, given any node $a = (i_1, \ldots, i_{K-1}) \in S$, we have $(a, b, j) \in E$ for every b of the form $b = (i_2, \ldots, i_{K-1}, j)$, for any $j \in \langle M \rangle$. The *dual De Bruijn graph of order* $K-1$ *(on the alphabet* $\langle M \rangle$*)* is the dual of $\mathcal{G}_{db}^{K}$.

Figure 6.4 provides two examples of De Bruijn graphs of different orders.

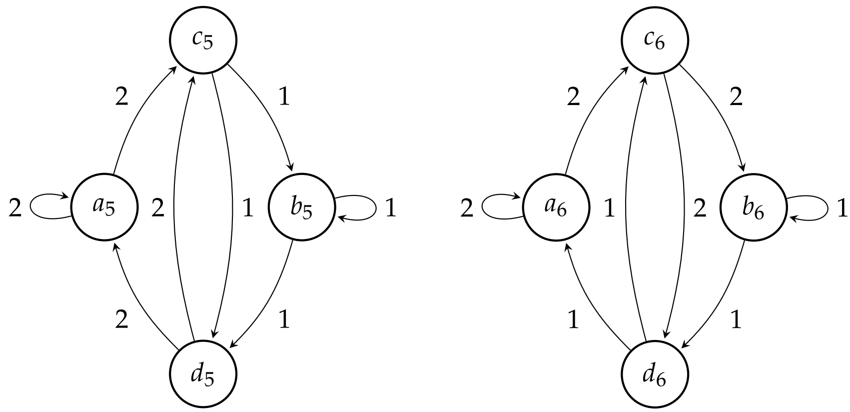
Recently, the notion of De Bruijn graph has been generalized in [DJ23, Jun24] by mixing common past and future of finite words. Even if they are not complete anymore, those graphs have been proved [DJ23, Proposition 4] to be path-complete.



(a) The De Bruijn graph $\mathcal{G}_{db}^{2}$ of order 1 on the alphabet $\langle 3 \rangle$.

(b) The De Bruijn graph $\mathcal{G}_{db}^{3}$ of order 2 on the alphabet $\langle 2 \rangle$.

Fig. 6.4 Examples of De Bruijn graphs.



(a) $\mathcal{G}_{db}^{3,1}$, the generalized De Bruijn graph of order 2 and memory 1.

(b) $\mathcal{G}_{db}^{3,0}$, the generalized De Bruijn graph of order 2 and memory 0.

Fig. 6.5 Examples of *generalized* De Bruijn graphs of order 2 on the alphabet $\langle 2 \rangle$.

Definition 6.11 (Generalized De Bruijn graphs)**.** Given M, K and $k \in \mathbb{N}$ such that $0 \leq k \leq K$, the *generalized (primal) De Bruijn graph of order* $K-1$ *and memory* k *(on the alpahbet* $\langle M \rangle$*)*, denoted by $\mathcal{G}_{db}^{K,k} := (S, E)$ is defined as follows: $S := \langle M \rangle^{K-1}$ and

$$(a, b, h) \in E \Leftrightarrow \begin{cases} a_{2:K-1} = b_{1:K-2} \wedge b_{K-1} = h, & \text{if } k = K-1, \\ a_{2:K-1} = b_{1:K-2} \wedge a_{k+1} = h, & \text{otherwise.} \end{cases}$$

Note that the generalized De Bruijn graph of order $K-1$ of full memory, i.e. $k = K-1$, coincides with the classical De Bruijn graph of order $K-1$. Figure 6.5 illustrates the generalized De Bruijn graphs of order 2 and memory 1 and 0.

6.2.2 Path-complete Lyapunov functions

As already mentioned, several multiple Lyapunov certificates have been introduced over the years. For the purposes of unification and generalization, Ahmadi et al. introduced in [AJPR14] the *path-complete Lyapunov framework*. In short, the main idea consists in using a path-complete graph to encode the Lyapunov constraints on a set of Lyapunov functions. Formally, the nodes of the graph represent the Lyapunov pieces while the edges describe Lyapunov inequalities. Then, the path-completeness of the graph is required to guarantee that the corresponding set of Lyapunov inequalities is a stability certificate.

Definition 6.12 (Path-complete Lyapunov function)**.** Given a switched system $F = \{f_i : i \in \langle M \rangle\} \subset \mathcal{C}^0(\mathbb{R}^n, \mathbb{R}^n)$ of dimension $n \in \mathbb{N}$ and a template $\mathcal{V}$ of candidate Lyapunov functions, a *path-complete Lyapunov function* (PCLF in short) for F in $\mathcal{V}$ is a pair $(\mathcal{G} = (S, E), V_S)$ where $\mathcal{G}$ is a path-complete graph on $\langle M \rangle$, and $V_S := \{V_s : s \in S\} \in \mathcal{V}^S$ is a set of candidate Lyapunov functions in the template such that the following inequalities are satisfied:

$$\forall\, (a, b, i) \in E,\ \forall x \in \mathbb{R}^n : \ V_b(f_i(x)) \ \leq \ V_a(x). \tag{6.5}$$

If this is the case, we say that V_S is *admissible for* $\mathcal{G}$ *and* F, and we denote it by $V_S \in PCLF(\mathcal{G}, F)$.

Remark 6.13. Note that if we consider a graph with an edge $e := (s, d, w)$ of label $w := i_1 \dots i_k$ of length $k > 1$, the corresponding Lyapunov inequality is given by

$$V_d(f_{i_k} \circ \cdots \circ f_{i_1}(x)) \ \leq \ V_s(x),$$

for all $x \in \mathbb{R}^n$. △

The following theorem states that a path-complete Lyapunov function is a sufficient condition for stability.

Theorem 6.14 ([AJPR14, Phi17])**.** *Consider a switched system $F := \{f_i : i \in \langle M \rangle\}$ with $M \in \mathbb{N}$ continuous dynamics on $\mathbb{R}^n$. If there exists a path-complete Lyapunov function $(\mathcal{G} = (S, E), V_S := \{V_s : s \in S\})$ for F, then the switched system is stable under arbitrary switching.*

Not only the path-completeness is a sufficient condition for stability, but it has been proved that it is also necessary. Therefore, the path-complete Lyapunov framework fully characterizes the Lyapunov certificates for switched systems.

Theorem 6.15 (Theorem 3 in [JAPR17])**.** *A set of Lyapunov inequalities is a stability certificate under arbitrary switching if and only if the corresponding graph is path-complete.*

Remark 6.16. On the one hand, the path-completeness characterizes the Lyapunov certificates for the stability under arbitrary switching, as stated in Theorem 6.15. On the other hand, we can use the graph to encode constraints on the switching sequences such as dwell-time restrictions [AJ18b].

In general, this formalism allows to prove the stability on the *language* of the graph, i.e. on the set of possible finite sequences that we can generate with the graph. This has been extensively studied in [Phi17]. △

6.2.3 Approximation of the JSR using PCLFs

Let us consider in particular a linear switched system of the form

$$x(k+1) = A_{\sigma(k)}x(k) \tag{6.6}$$

where $A_{\sigma(k)} \in \mathcal{A} := \{A_1, \dots, A_M\} \subset \mathbb{R}^{n\times n}$ for every time step $k \in \mathbb{N}$. We have seen that we can prove the stability of this kind of systems using the path-complete Lyapunov framework, but we can also use it to derive upper bounds on the JSR of $\mathcal{A}$. Due to the homogeneity of the JSR (see Proposition 2.2), if we manage to find an admissible set of Lyapunov functions in the given template for a path-complete graph $\mathcal{G}$ and the scaled set of matrices

$$\mathcal{A}_\gamma := \{A_i/\gamma : i = 1, \dots, M\}$$

with $\gamma > 0$, then $\rho(\mathcal{A}) \leq \gamma$. Therefore, we can define the approximation of the JSR provided by a path-complete graph and a template as the smallest value $\gamma^\star$ which satisfies this inequality.

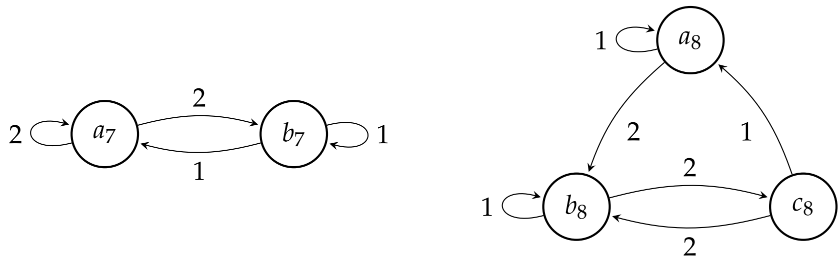
Definition 6.17 (JSR approximation provided by a path-complete graph and a template)**.** Given a linear switched system with dynamics $\mathcal{A} := \{A_1, \dots, A_M\} \subset \mathbb{R}^{n\times n}$, a path-complete graph $\mathcal{G}$ on $\langle M\rangle$ and a template of candidate Lyapunov functions $\mathcal{V}$. The *approximation of the JSR of $\mathcal{A}$ provided by $\mathcal{G}$ and $\mathcal{V}$*, denoted by $\rho_{\mathcal{G},\mathcal{V}}(\mathcal{A})$, is defined as the smallest value $\gamma > 0$ such that the lifted system $\mathcal{A}_\gamma$ admits a path-complete Lyapunov function $(\mathcal{G}, V_\gamma)$ in the template, i.e.

$$\rho_{\mathcal{G},\mathcal{V}}(\mathcal{A}) := \min_{\gamma>0} \left\{\exists V_\gamma \in \mathcal{V}^S \mid V_\gamma \in PCLF(\mathcal{G}, \mathcal{A}_\gamma)\right\}. \tag{6.7}$$

Let us take a few examples.

Example 6.18. We consider a linear switched system with two modes $\mathcal{A}_1 := \{A_1, A_2\} \subset \mathbb{R}^{2\times 2}$ in dimension 2, where the matrices are defined by

$$A_1 := \begin{bmatrix} 0.9 & 0.3 \\ 0.9 & 0.7 \end{bmatrix} \text{ and } A_2 := \begin{bmatrix} 0.6 & 0.9 \\ 0.6 & 0.3 \end{bmatrix}.$$



(a) $\mathcal{G}_7 = (S_7, E_7)$, a path-complete graph on $\langle 2 \rangle$ with 2 nodes and 4 edges.

(b) $\mathcal{G}_8 = (S_8, E_8)$, a path-complete graph on $\langle 2 \rangle$ with 3 nodes and 6 edges.

Fig. 6.6 Comparison of the approximation of the JSR provided by the two path-complete graphs $\mathcal{G}_7$ and $\mathcal{G}_8$ with different templates of candidate Lyapunov functions.

We use two different path-complete graphs $\mathcal{G}_7$ and $\mathcal{G}_8$, illustrated in Figure 6.6, in addition to the common Lyapunov function graph $\mathcal{G}_0$ and the different (generalized) De Bruijn graphs $\mathcal{G}_{db}^3$, $\mathcal{G}_{db}^{3,1}$ and $\mathcal{G}_{db}^{3,1}$. Moreover, we use different templates of Lyapunov functions, namely the primal copositive norms which have been formally introduced in Section 2.2.3 and the quadratic Lyapunov functions. For each possible graph-template couple, we compute the corresponding approximation of the JSR. The results are summarized in Table 6.1.

We can see that the approximation varies according to the graph and the template. In general, the template of primal copositive norms provides poorer results while the quadratic template reaches the JSR value, that is 1.3534 (computed using the JSR toolbox [VHJ14]), with four graphs. However, the graph $\mathcal{G}_{[0,2]}$ allows to reach an approximation of 1.3545 using the

Template	**Path-complete graph**					
	$\mathcal{G}_0$	$\mathcal{G}_{db}^3$	$\mathcal{G}_{[1,1]}$	$\mathcal{G}_{[0,2]}$	$\mathcal{G}_7$	$\mathcal{G}_8$
Copositive norms	1.5487	1.5487	1.3807	1.3545	1.3807	1.5487
Quadratics	1.3736	1.3534	1.3534	1.3534	1.3534	1.3638

Table 6.1 $\rho_{\mathcal{G},\mathcal{V}}(\mathcal{A}_1)$, the approximation of the JSR of system $\mathcal{A}_1$ in Example 6.18 with different path-complete graphs $\mathcal{G}$ and different templates $\mathcal{V}$. The JSR of $\mathcal{A}_1$, i.e. 1.3534, is achieved by the quadratic template with different graphs.

primal copositive norms. Therefore, this example shows that a graph can generate both "good" and "bad" results, depending on the template, and some templates which may seem less effective, sometimes turn out to be really effective with particular graphs. The next chapters will partially explain these fluctuations. △

As illustrated by this example, this quantity depends both on the graph and on the template that we consider. In fact, we will use it as a performance index to compare path-complete graphs in the next section.

In addition, some hierarchies of graphs are known to converge to the JSR value. For instance, [AJPR14] proved that the class of De Bruijn graphs with increasing order leads to the JSR if we use the quadratic template.

Theorem 6.19 (Theorem 6.1 in [AJPR14]). *Given $\mathcal{A} \subset \mathbb{R}^{n\times n}$ a linear switched system of M matrices of dimension $n \in \mathbb{N}$ and any integer $l \in \mathbb{N}$, we consider the dual De Bruijn graph of order $l-1$ on the alphabet $\langle M \rangle$ denoted by $\mathcal{G}^l_{db^\top}$, and the quadratic template denoted by $\mathcal{Q}$. The following inequalities are satisfied:*

$$\frac{1}{\sqrt[2l]{n}}\,\rho_{\mathcal{G}^l_{db^\top},\mathcal{Q}}(\mathcal{A}) \;\leq\; \rho(\mathcal{A}) \;\leq\; \rho_{\mathcal{G}^l_{db^\top},\mathcal{Q}}(\mathcal{A}). \tag{6.8}$$

This theorem leads to a hierarchy of semidefinite programs to approximate the JSR with a given precision.

Example 6.20. Let us consider the linear switched system of 3 modes $\mathcal{A}_2 := \{A_1, A_2, A_3\} \subset \mathbb{R}^{3\times 3}$ of dimension 3 defined by

$$A_1 := \begin{bmatrix} -4.6740 & 1.6918 & -0.3927 \\ 0.6120 & -3.0957 & 4.8164 \\ 3.8187 & -1.3108 & -3.4360 \end{bmatrix},\; A_2 := \begin{bmatrix} 3.5552 & -3.0908 & -3.7939 \\ 1.4476 & -0.7175 & 0.8951 \\ -1.2373 & -0.1798 & -2.7381 \end{bmatrix}$$

$$\text{and } A_3 := \begin{bmatrix} -1.1538 & -2.0956 & 3.2438 \\ 0.8299 & 1.1709 & 4.8266 \\ -2.4819 & -2.3472 & 2.3025 \end{bmatrix}.$$

Using the JSR toolbox [VHJ14], we find that the JSR of $\mathcal{A}_2$ is 5.8353.

For comparison, we use the quadratic De Bruijn hierarchy in Theorem 6.19 to approximate the JSR of system $\mathcal{A}_2$. Table 6.2 summarizes the JSR approximations and the computation time of the first iterations of this hierarchy. One can observe that the JSR approximation provided by $\mathcal{G}^{K+1}_{db}$ decreases as the order K increases. After 8 iterations, the JSR is approximated with an error order of 10^{-3}. In parallel, the computation time in-

	De Bruijn graph $\mathcal{G}_{db}^{K+1}$ of order K							
	2	3	4	5	6	7	8	9
$\rho_{\mathcal{G},\mathcal{Q}}(\mathcal{A}_2)$	5.9345	5.8477	5.8466	5.8445	5.8425	5.8419	5.8402	5.8360
Time (sec.)	0.5797	1.3560	2.8771	8.9788	26.659	95.981	378.72	2044.5

Table 6.2 Approximation of the JSR of system $\mathcal{A}_2$ using the quadratic De Bruijn hierarchy in Theorem 6.19. For comparison, $\rho(\mathcal{A}_2) = 5.8353$. The computation time (in seconds) of each iteration is also included.

creases as well since the number of nodes (i.e. SDP variables) $\langle M \rangle^K$ and the number of edges (i.e. SDP constraints) $\langle M \rangle^{K+1}$ both grow exponentially with respect to the order K. △

6.3 Comparison of PCLFs

The path-complete Lyapunov function framework generates a wide range of Lyapunov stability certificates since it provides two degrees of freedom: the path-complete graph $\mathcal{G}$ and the template $\mathcal{V}$. Example 6.18 has highlighted that the conservatism of these stability certificates differs according to the template and the path-complete graph used, i.e. all certificates do not provide the same JSR approximation.

6.3.1 Definition of the comparison

Similarly to Definition 1.30, we define a family of systems $\mathcal{F}$ as a family of countably many sets of systems of fixed dimension, i.e.

$$\mathcal{F} := \bigcup_{n \in \mathbb{N}} \mathcal{F}_n,$$

where $\mathcal{F}_n \subseteq \mathcal{C}^0(\mathbb{R}^n, \mathbb{R}^n)$. In what follows, we introduce ordering relations among the set of path-complete graphs, formalizing the idea that one graph "produces less conservative stability conditions" with respect to another. This notion has been initially introduced in [AJPR14, Section 4.2] and further developed in [Phi17, PJ19].

Definition 6.21 (Ordering relations between graphs)**.** Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\tilde{S}, \tilde{E})$ on the alphabet $\langle M \rangle$, a template of candidate Lyapunov functions $\mathcal{V}$ and a family $\mathcal{F}$ of systems.

(a) We say that $\widetilde{\mathcal{G}}$ *is better than* $\mathcal{G}$ *with respect to the template* $\mathcal{V}$ *and the family* $\mathcal{F}$, denoted by

$$\mathcal{G} \ \leq_{\mathcal{V},\mathcal{F}} \ \widetilde{\mathcal{G}}, \tag{6.9}$$

if, for any $F \in \mathcal{F}^{\langle M \rangle}$,

$$\Big[\exists V \in \mathcal{V}^S \text{ s.t. } V \in PCLF(\mathcal{G}, F)\Big] \Rightarrow \Big[\exists W \in \mathcal{V}^{\tilde{S}} \text{ s.t. } W \in PCLF(\widetilde{\mathcal{G}}, F)\Big]. \tag{6.10}$$

(b) We say that $\widetilde{\mathcal{G}}$ *is better than* $\mathcal{G}$ *with respect to the template* $\mathcal{V}$, denoted by

$$\mathcal{G} \ \leq_{\mathcal{V}} \ \widetilde{\mathcal{G}}, \tag{6.11}$$

if the inequality in Equation (6.9) is satisfied for $\mathcal{F} = \bigcup_{n \in \mathbb{N}} \mathcal{C}^0(\mathbb{R}^n, \mathbb{R}^n)$.

(c) We say that $\widetilde{\mathcal{G}}$ *is better than* $\mathcal{G}$, denoted by

$$\mathcal{G} \ \leq \ \widetilde{\mathcal{G}}, \tag{6.12}$$

if for any template $\mathcal{V}$, the inequality in Equation (6.11) is satisfied.

Figure 6.7 provides an illustration of the ordering in Definition 6.21.

The following lemma proves that these ordering relations are well-defined in particular in the context of linear switched systems since they can be translated in terms of comparison of the JSR approximations.

Proposition 6.22. *Consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ on the same alphabet, a template of candidate Lyapunov functions $\mathcal{V}$ and the family $\mathcal{L}$ of linear switched systems. If $\mathcal{G} \ \leq_{\mathcal{V},\mathcal{F}} \ \widetilde{\mathcal{G}}$, then*

$$\rho_{\mathcal{G},\mathcal{V}}(\mathcal{A}) \ \geq \ \rho_{\widetilde{\mathcal{G}},\mathcal{V}}(\mathcal{A}) \tag{6.13}$$

for any system $\mathcal{A} \in \mathcal{L}^{\langle M \rangle}$.

Proof. Consider an arbitrary linear switched system $\mathcal{A} := \{A_1, \ldots, A_M\}$ with M modes of dimension $n \in \mathbb{N}$, and a template $\mathcal{V}$ of candidate Lyapunov functions. Take any $\gamma > 0$, and assume that there exists a PCLF

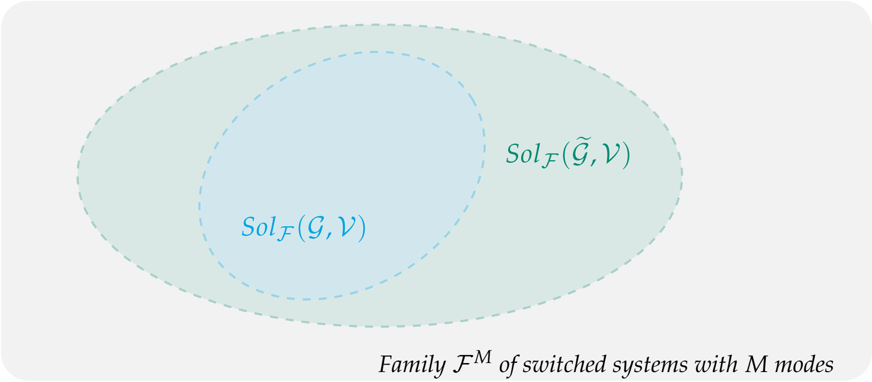


Fig. 6.7 Illustration of ordering of graphs in Definition 6.21. Given a template $\mathcal{V}$, a path-complete graph $\mathcal{G} = (S, E)$ and a family $\mathcal{F}$ of systems, $Sol_{\mathcal{F}}(\mathcal{G}, \mathcal{V})$ denotes the set of switched systems with dynamics in $\mathcal{F}$ for which there exists an admissible solution in the template $\mathcal{V}$, i.e. $Sol_{\mathcal{F}}(\mathcal{G}, \mathcal{V}) := \{F = \{f_1, \ldots, f_M\} \in \mathcal{F}^{\langle M \rangle} \mid \exists V \in \mathcal{V}^S : V \in PCLF(\mathcal{G}, F)\}$. Then, $\mathcal{G} \leq_{\mathcal{V},\mathcal{F}} \widetilde{\mathcal{G}}$ if and only if $Sol_{\mathcal{F}}(\mathcal{G}, \mathcal{V}) \subseteq Sol_{\mathcal{F}}(\widetilde{\mathcal{G}}, \mathcal{V})$.

$(\mathcal{G}, V_S)$ with $V_S \in \mathcal{V}^S$, admissible for $\mathcal{A}_\gamma$, i.e.

$$\forall (s, d, \sigma) \in E,\ \forall x \in \mathbb{R}^n :\ V_d\left(\frac{A_\sigma}{\gamma} x\right) \leq V_s(x). \tag{6.14}$$

By assumption, $\mathcal{G} \leq_{\mathcal{V},\mathcal{F}} \widetilde{\mathcal{G}}$, therefore there exists a PCLF $(\widetilde{\mathcal{G}}, W_{\tilde{S}})$ with $W_{\tilde{S}} \in \mathcal{V}^{\tilde{S}}$ admissible for $\mathcal{A}_\gamma$, i.e.

$$\forall (\tilde{s}, \tilde{d}, \sigma) \in \widetilde{E},\ \forall x \in \mathbb{R}^n :\ V_{\tilde{d}}\left(\frac{A_\sigma}{\gamma} x\right) \leq V_{\tilde{s}}(x). \tag{6.15}$$

By Definition 6.17 and since Equation (6.15) holds for any γ such that Equation (6.14) holds, we can conclude that for any linear switched system $\mathcal{A}$, the inequality in Equation (6.13) holds. □

Note that the relations in Equations (6.9), (6.11) and (6.12) are actually *preorder relations* and not *order relations* because these relations are not antisymmetric. However, they satisfy the transitivity property: for any path-complete graphs $\mathcal{G}$, $\mathcal{G}'$ and $\mathcal{G}''$,

$$\left[(\mathcal{G} \leq \mathcal{G}') \wedge (\mathcal{G}' \leq \mathcal{G}'')\right] \Rightarrow (\mathcal{G} \leq \mathcal{G}''), \tag{6.16}$$

and the same property remains true considering the relations $\leq_{\mathcal{V}}$ and $\leq_{\mathcal{V},\mathcal{F}}$, for any template $\mathcal{V}$ and any family of systems $\mathcal{F}$. Moreover, since Equation (6.12) implies Equations (6.9) and (6.11) for any template $\mathcal{V}$ and any family $\mathcal{F}$, the following implications hold:

$$\left[(\mathcal{G} \leq \mathcal{G}') \wedge (\mathcal{G}' \leq_{\mathcal{V}} \mathcal{G}'')\right] \Rightarrow (\mathcal{G} \leq_{\mathcal{V}} \mathcal{G}''), \tag{6.17}$$

and

$$\left[(\mathcal{G} \leq \mathcal{G}') \wedge (\mathcal{G}' \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}'')\right] \Rightarrow (\mathcal{G} \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}''). \tag{6.18}$$

Moreover, the three relations in Equations (6.9), (6.11) and (6.12) satisfy the two following properties.

Proposition 6.23. *Consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ on the same alphabet, a template of candidate Lyapunov functions $\mathcal{V}$ and a family of switched systems $\mathcal{F}$ such that $\mathcal{G} \ \leq_{\mathcal{V},\mathcal{F}} \ \widetilde{\mathcal{G}}$. Then, for any path-complete component $\mathcal{G}'$ of $\widetilde{\mathcal{G}}$,*

$$\mathcal{G} \ \leq_{\mathcal{V},\mathcal{F}} \ \mathcal{G}'.$$

The same result holds for the relations in Equations (6.11) and (6.12).

The proof of Proposition 6.23 is straightforward, since by definition the set of inequalities encoded by $\mathcal{G}'$ is a subset of the inequalities encoded by $\widetilde{\mathcal{G}}$. A second property follows directly from Definition 6.21 and involves the common Lyapunov function graph $\mathcal{G}_0$.

Proposition 6.24. *Consider a path-complete graph $\mathcal{G}$ and the common Lyapunov function graph $\mathcal{G}_0$ on the same alphabet. Then, for any template of candidate Lyapunov functions $\mathcal{V}$ and any family of switched systems $\mathcal{F}$,*

$$\mathcal{G}_0 \ \leq_{\mathcal{V},\mathcal{F}} \ \mathcal{G}.$$

The same result holds for the relations in Equations (6.11) and (6.12).

In other words, this means that, for any switched system, if there exists a common Lyapunov function in a given template then there exists an admissible solution for any multiple Lyapunov stability certificate.

6.3.2 Comparison and duality

In this section, we leverage the duality of norms described in Section 1.2 to build a duality theory for the template-dependent ordering of path-complete graphs in the context of linear switched systems.

First, we prove that the dual norms of a given solution $V \in PCLF(\mathcal{G}, \mathcal{A})$ is admissible for the transposed matrices $\mathcal{A}^\top$ and the dual graph $\mathcal{G}^\top$.

Lemma 6.25. *Given a path-complete graph $\mathcal{G} = (S, E)$, a linear switched system $\mathcal{A}$ and a template $\mathcal{V}$ of norms, the following equivalence holds:*

$$\begin{aligned} V \; &:= \; \{V_s \in \mathcal{V} : s \in S\} \in PCLF(\mathcal{G}, \mathcal{A}) \\ \Leftrightarrow \; V^* \; &:= \; \{{V_s}^* \in \mathcal{V}^* : s \in S\} \in PCLF(\mathcal{G}^\top, \mathcal{A}^\top). \end{aligned}$$

Proof. Consider $V := \{V_s : s \in S\} \in \mathcal{V}^S$ an admissible solution for $\mathcal{G}$ and $\mathcal{A}$. By definition, it means that all the Lyapunov inequalities encoded by the edges of $\mathcal{G}$ are satisfied, i.e.

$$\forall (s, d, i) \in E : \; \forall x \in \mathbb{R}^n, \; V_d(A_i x) \; \leq \; V_s(x).$$

By Lemma 1.28, this is equivalent to

$$\begin{aligned} &\forall (s, d, i) \in E : \; \forall x \in \mathbb{R}^n, \; {V_s}^*(A_i^\top x) \; \leq \; {V_d}^*(x), \\ \Leftrightarrow \quad &\forall (d, s, i) \in E^\top : \; \forall x \in \mathbb{R}^n, \; {V_s}^*(A_i^\top x) \; \leq \; {V_d}^*(x), \end{aligned}$$

which means that $V^* := \{{V_s}^* : s \in S\} \in {\mathcal{V}^*}^S$ is admissible for the dual graph $\mathcal{G}^\top$ and the transpose modes $\mathcal{A}^\top$. □

We can directly derive from Lemma 6.25 the following result in terms of JSR approximation.

Lemma 6.26. *Consider a path-complete graph $\mathcal{G}$ on the alphabet $\langle M \rangle$, a template $\mathcal{V}$ of norms and a linear switched system $\mathcal{A} \subset \mathbb{R}^{n \times n}$. Then,*

$$\rho_{\mathcal{G}, \mathcal{V}}(\mathcal{A}) \; = \; \rho_{\mathcal{G}^\top, \mathcal{V}^*}(\mathcal{A}^\top).$$

If a template $\mathcal{V}$ is self-dual, Lemma 6.26 states that transposing the matrices has the same effect as dualizing the graph. In particular, this result generalizes [AJPR14, Theorem 5.1] which claims the same statement for the quadratic template in particular.

Finally, the following proposition formally demonstrates that the ordering in Definition 6.21 can as easily be studied in a template as in its dual, by simply transposing the graphs.

Proposition 6.27. *Consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ on the same alphabet, and a template $\mathcal{V}$ of norms. The following equivalence holds:*

$$\mathcal{G} \leq_{\mathcal{V},\mathcal{L}} \widetilde{\mathcal{G}} \Leftrightarrow \mathcal{G}^\top \leq_{\mathcal{V}^*,\mathcal{L}} \widetilde{\mathcal{G}}^\top$$

where $\mathcal{L}$ denotes the family of linear systems.

Proof. Assume that $\mathcal{G} \leq_{\mathcal{V},\mathcal{L}} \widetilde{\mathcal{G}}$. By Definition 6.21 of the graph ordering, the following implication holds for any linear switched system $\mathcal{A} \in \mathcal{L}^{\langle M \rangle}$:

$$\left[\exists V \in \mathcal{V}^S \text{ s.t. } V \in PCLF(\mathcal{G}, \mathcal{A})\right] \Rightarrow \left[\exists W \in \mathcal{V}^{\tilde{S}} \text{ s.t. } W \in PCLF(\widetilde{\mathcal{G}}, \mathcal{A})\right].$$

We can use Lemma 6.25 for both members of this implication. Then,

$$\left[\exists V \in \mathcal{V}^{*S} \text{ s.t. } V \in PCLF(\mathcal{G}^\top, \mathcal{A}^\top)\right] \Rightarrow \left[\exists W \in \mathcal{V}^{*\tilde{S}} \text{ s.t. } W \in PCLF(\widetilde{\mathcal{G}}^\top, \mathcal{A}^\top)\right]$$

holds for any linear switched system $\mathcal{A}$, which ends the proof. □

6.3.3 Characterization by the simulation

So far, ad-hoc techniques were used to establish the ordering relations between graphs. In particular, most of these techniques provide sufficient conditions for ordering. In this section, we summarize the first necessary and sufficient condition for the general ordering relation in Equation (6.12) between graphs developed in [PJ19], which relies on a combinatorial relation between graphs called the *simulation* defined in [PJ19, Definition 3.1].

Definition 6.28 (Simulation)**.** Consider two graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet $\langle M \rangle$. The graph $\mathcal{G}$ *simulates* $\widetilde{\mathcal{G}}$ if and only if there exists a function $R : \widetilde{S} \to S$ such that

$$\forall (a, b, i) \in \widetilde{E} : (R(a), R(b), i) \in E. \tag{6.19}$$

Remark 6.29. Although this notion has been introduced in [PJ19] for the path-complete Lyapunov formalism, it recalls basic notions in automata and graph theory.

In graph theory, the simulation is a generalization of a *graph homomorphism* [HT97, HN04] for directed and labeled graphs. A graph homomorphism is a mapping between the nodes of two graphs which preserves the edges, and generalizes graph colouring problems. From a computational complexity point of view, the problem of deciding whether there exists a homomorphism from a graph to another has been proved to be NP-complete. However, several particular cases can be solved in polynomial time, see [HN08] for a survey on this topic.

In contrast, Definition 6.28 is inspired by the notion of simulation in automata theory [CL10, Section 2.3.5], and is a particular case of [Phi17, Definition 6.4]. In this case, the authors say that a graph $\mathcal{G}_1$ simulates a graph $\mathcal{G}_2$ if there exists a function $R : S_2 \to \mathcal{P}(S_1)$ which associates every node of S_2 to a subset of nodes of S_1 such that for every edge $(s, d, \sigma) \in E_2$, there exists a bijection between $R(s)$ and $R(d)$ using the edges in E_1 of label σ. Then, Definition 6.28 is the special case when $R(s)$ is a singleton. △

This notion defines a relation between the nodes of the two graphs, where the nodes of the simulated graph are sent to one node of the simulating graph. We can translate this relation in terms of candidate Lyapunov functions: we define the function associated to a node $\tilde{s} \in \widetilde{S}$, denoted by $W_{\tilde{s}}$, as equal to the function associated to the corresponding node $R(s)$ in S, itself referred as $V_{R(s)}$. To summarize,

$$\forall \tilde{s} \in \widetilde{S},\ \forall x \in \mathbb{R}^n :\ W_{\tilde{s}}(x) := V_{R(\tilde{s})}(x).$$

One can prove that under this assumption, for any system F and any template $\mathcal{V}$, the set of candidate Lyapunov functions $\{W_{\tilde{s}}\}_{\tilde{s} \in \widetilde{S}}$ in $\mathcal{V}^{\tilde{S}}$ is admissible for $\widetilde{\mathcal{G}}$ and F if the corresponding set $\{V_s\}_{s \in S}$ in $\mathcal{V}^S$ is admissible for $\mathcal{G}$ and F. The following theorem shows that the simulation relation is not only a sufficient condition for ordering, but also a necessary condition.

Theorem 6.30 (Theorem 3.5 in [PJ19])**.** *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet. The following statements are equivalent:*

(1) $\mathcal{G}$ simulates $\widetilde{\mathcal{G}}$.

(2) $\mathcal{G} \leq \widetilde{\mathcal{G}}$ in the sense of Definition 6.21(c).

This theorem provides a combinatorial characterization of the general ordering relation in Equation (6.12) between path-complete graphs. This implies that the conservatism of a path-complete graph mainly relies on its structure. Therefore, the conservatism-based comparison of path-complete graphs can be achieved by comparing their structures and relate them through a simulation relation. Moreover, this relation can be checked in practice.

In the following chapters, we will derive similar theorems for weaker ordering relations whose proof is widely inspired by the proof of Theorem 6.30. To make the proofs of the theorems in Chapter 8 easier to understand, we recall and explain the proof of Theorem 6.30 in detail. The proof mainly relies on the following technical result which states that given any graph, it is possible to build a linear switched system and a set of candidate Lyapunov functions such that a Lyapunov inequality is satisfied if and only if it is encoded by one of the edges of the graph.

Lemma 6.31 (Theorem 3.6 in [PJ19]). *For any graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, there exists an integer $n \in \mathbb{N}$, a system $F := \{f_i : i \in \langle M \rangle\}$ in dimension n and $|S|$ candidate Lyapunov functions $\{V_s : s \in S\}$ such that*

$$\forall e = (s, d, i) \in E,\ \forall x \in \mathbb{R}^n : \ V_d\left(f_i(x)\right) \ \leq \ V_s(x), \tag{6.20}$$

$$\forall e = (s, d, i) \in \overline{E},\ \exists \tilde{x} \in \mathbb{R}^n : \ V_d\left(f_i(\tilde{x})\right) \ > \ V_s(\tilde{x}), \tag{6.21}$$

where $\overline{E} := S \times S \times \langle M \rangle \setminus E$.

Proof. The proof of this theorem is constructive. We first build the system $F := \{f_i : i \in \langle M \rangle\}$ and the set of candidate Lyapunov functions $\{V_s : s \in S\}$, and then, we prove that the inequalities are satisfied.

We build a linear switched system $F := \{A_i \in \mathbb{R}^{n \times n} : i \in \langle M \rangle\}$ where $n = 2|\overline{E}|$. These matrices are defined block-wise with $|\overline{E}|$ blocks of dimension 2×2 on the diagonal, and zero everywhere else. Since there are $|\overline{E}|$ blocks, we will identify each of them with an edge in $\overline{E}$, i.e. $A_i[e]$ is the 2×2 block associated to the edge $e \in \overline{E}$. Then, we define

$$A_i[e] \ := \begin{cases} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} & \text{if } label(e) = i, \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \text{otherwise.} \end{cases}$$

For the template, we consider candidate diagonal quadratic Lyapunov functions of the form

$$V_s(x) \ := \ x^\top diag(v_s) x,$$

where the vectors $v_s \in \mathbb{R}^n_{>0}$ for $s \in S$ are defined block-wise, where each 2-dimensional block $v_s[e]$ is indexed by an edge $e \in \overline{E}$. In this context, we have that for any edge $(s, d, i) \in S \times S \times \langle M \rangle$, the Lyapunov inequality encoded by this edge is satisfied if and only if for all $x \in \mathbb{R}^n$:

$$
\begin{aligned}
& (A_i x)^\top diag(v_d) A_i x \le x^\top diag(v_s) x, \\
\Leftrightarrow \quad & \forall\, e \in \overline{E},\ x[e]^\top A_i[e]^\top diag(v_d[e]) A_i[e] x[e] \le x[e]^\top diag(v_s[e])\, x[e],
\end{aligned}
$$

Then, there are two different cases.

- Either the edge $e \in \overline{E}$ has a label different from σ. Then, the matrix $A_i[e]$ is nul, and the inequality is satisfied for all $x[e]$.
- Or, the edge $e \in \overline{E}$ has i as label. Then, the inequality is given by

$$
\forall\, x[e] \in \mathbb{R}^2,\ x[e]^\top \begin{bmatrix} v_d[e]_2 & 0 \\ 0 & 0 \end{bmatrix} x[e] \le x[e]^\top \begin{bmatrix} v_s[e]_1 & 0 \\ 0 & v_s[e]_2 \end{bmatrix} x[e]. \tag{6.22}
$$

  Since it has to be satisfied for any $x[e]$, this is equivalent to saying that the second component of $v_d[e]$ is smaller than the first component of $v_s[e]$. Moreover, if this condition is not satisfied, i.e. $v_d[e]_2 > v_s[e]_1$, then $x[e] = (1, 0)^\top$ does not satisfy the inequality, since the inequality (6.22) becomes

$$
v_d[e]_2 > v_s[e]_1, \tag{6.23}
$$

  and then highlights the relation between the component of $v_d[e]$ and $v_s[e]$. In general, any vector $x[e] = (\alpha, 0)$ for $\alpha \neq 0$ works, while $x[e] = (0, \alpha)^\top$ satisfy the inequality for instance. Note that these vectors do not depend on anything, and could be chosen for any edge.

Then, we have that, given an edge $(s, d, i) \in S \times S \times \langle M \rangle$, the inequalities associated to unexisting edges with a label different from i are always satisfied and only those with a label i imply conditions on the values of the vectors v_s. So, we have that

$$
\begin{aligned}
& [\forall x \in \mathbb{R}^n : V_d(A_i x) \le V_s(x)] \\
\Leftrightarrow \quad & \left[\forall e \in \overline{E} \text{ s.t. } label(e) = i : v_d[e]_2 \le v_s[e]_1\right].
\end{aligned} \tag{6.24}
$$

Regarding the negation, if there exists $\bar{x}$ such that the Lyapunov inequality encoded by (s, d, i) is not satisfied, it means that there exists at least one edge $e \in \overline{E}$ with label i (otherwise, the inequality is always satisfied) such that $v_d[e]_2 > v_s[e]_1$. Then in particular, the vector $\bar{x}$ defined such that $\bar{x}[e] = (1, 0)^\top$ (and anything everywhere else) does not satisfy the Lyapunov inequality since it will highlight that the equivalent condition on $v_d[e]$ and $v_s[e]$ is not satisfied.

Finally, we just need to define the vectors v_s. Given $s \in S$, the 2-dimensional bloc $v_s[e_2]$ for $e_2 = (s_2, d_2, \sigma)$ is defined by

- if $s = s_2 = d_2$, then $v_s[e_2] = [3,4]^\top$,
- if $s = s_2 \neq d_2$, then $v_s[e_2] = [2,1]^\top$,
- if $s = d_2 \neq s_2$, then $v_s[e_2] = [4,3]^\top$,
- otherwise, $v_s[e_2] = [4,1]^\top$.

We now have to prove that this construction satisfies (6.20) and (6.21). We start by showing that the Lyapunov inequalities (6.20) are satisfied. Let us consider an edge $e_1 = (s_1, d_1, i) \in E$. We want to prove that the corresponding Lyapunov inequality is satisfied, that is, recalling (6.24), for any $e_2 = (s_2, d_2, i) \in \overline{E}$, the second entry of $v_{d_1}[e_2]$ is smaller than the first entry of $v_{s_1}[e_2]$. We need to distinguish two different cases.

- If $s_1 = d_1$, then $v_{s_1} = v_{d_1}$. From our construction, the condition is satisfied in every situation except if $s_1 = s_2 = d_2$, where $v_{s_1}[e_2] = [3,4]^\top$. However, this cannot arise since it would imply that $e_1 = e_2$ while $e_1 \in E$ and $e_2 \in \overline{E}$ by assumption.
- If $s_1 \neq d_1$, there are 6 possible scenarios that could arise. Indeed we have to consider whether or not $s_1 = s_2$, $s_2 = d_2$ and $d_1 = d_2$. This leads to 8 possible scenarios. However, we discard the case when $s_1 = s_2 = d_1 = d_2$ since $s_1 \neq d_1$ by assumption, and the case $s_1 = s_2 \neq d_2 = d_1$ since it implies once again that $e_1 = e_2$, which is impossible by assumption. Table 6.3 summarizes the 6 scenarios and shows that the inequality is satisfied for each of them.

We now focus on expression (6.21). We consider any $e_1 = (s_1, d_1, i_1) \in \overline{E}$. We have to show that there exists $e_2 = (s_2, d_2, i_2) \in \overline{E}$ such that $v_{d_1}[e_2]_2 > v_{s_1}[e_2]_1$. We show that we can pick $e_2 := e_1$ to achieve this. Similarly, we distinguish two different cases.

- If $s_1 = d_1$, then $v_{s_1}[e_2] = v_{d_1}[e_2] := [3,4]^\top$ such that $v_{d_1}[e_2]_2 = 4 > 3 = v_{s_1}[e_2]_1$.
- Otherwise, if $s_1 \neq d_1$, we have $v_{s_1}[e_2] := (2,1)^\top$ and $v_{d_1} := [4,3]^\top$ such that $v_{d_1}[e_2]_2 = 3 > 2 = v_{s_1}[e_2]_1$.

This concludes the proof of Lemma 6.31. □

Scenario	Vector $v_{s_1}^\top[e_2]$	Vector $v_{d_1}^\top[e_2]$	Inequality $v_{d_1}[e_2]_2 \overset{?}{\leq} v_{s_1}[e_2]_1$
$s_1 = s_2 = d_2 \neq d_1$	$[3,4]$	$[4,1]$	$1 \leq 3$
$s_1 = s_2 \neq d_2 \neq d_1$	$[2,1]$	$[4,1]$	$1 \leq 2$
$s_1 \neq s_2 = d_2 = d_1$	$[4,1]$	$[4,1]$	$1 \leq 4$
$s_1 \neq s_2 = d_2 \neq d_1$	$[4,1]$	$[3,4]$	$4 \leq 4$
$s_1 \neq s_2 \neq d_2 = d_1$	$[4,1]$	$[4,3]$	$3 \leq 4$
$s_1 \neq s_2 \neq d_2 \neq d_1$	$[4,1]$	$[4,1]$	$1 \leq 4$

Table 6.3 The 6 possible scenarios when $s_1 \neq d_1$ in the proof of Lemma 6.31.

Let us take an example to show how it works in practice.

Example 6.32. Consider the path-complete graph $\mathcal{G}_7$ in Figure 6.6a with 4 edges, i.e. $E_7 = \{e_1 = (b_7, b_7, 1),\ e_2 = (b_7, a_7, 1), e_3 = (a_7, a_7, 2),\ e_4 = (a_7, b_7, 2)\}$ and 4 unexisting edges, i.e. $\overline{E_7} = \{e_5 = (a_7, a_7, 1), e_6 = (a_7, b_7, 1),$ $e_7 = (b_7, b_7, 2),\ e_8 = (b_7, a_7, 2)\}$. We can define the matrices A_1 and A_2:

$$A_1 = \begin{bmatrix} 0 & 0 & \times & \times & \times & \times & \times & \times \\ 1 & 0 & \times & \times & \times & \times & \times & \times \\ \times & \times & 0 & 0 & \times & \times & \times & \times \\ \times & \times & 1 & 0 & \times & \times & \times & \times \\ \times & \times & \times & \times & 0 & 0 & \times & \times \\ \times & \times & \times & \times & 0 & 0 & \times & \times \\ \times & \times & \times & \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & \times & \times & 0 & 0 \end{bmatrix}$$

$$\text{and } A_2 = \begin{bmatrix} 0 & 0 & \times & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times & \times & \times \\ \times & \times & 0 & 0 & \times & \times & \times & \times \\ \times & \times & 0 & 0 & \times & \times & \times & \times \\ \times & \times & \times & \times & 0 & 0 & \times & \times \\ \times & \times & \times & \times & 1 & 0 & \times & \times \\ \times & \times & \times & \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & \times & \times & 1 & 0 \end{bmatrix}.$$

Edge $e = (s, d, i)$	**Edges** $\tilde{e} = (s_2, d_2, i)$	**Vector** $v_s[\tilde{e}]$	**Vector** $v_d[\tilde{e}]$	**Inequality** $v_d[\tilde{e}]_2 \leq v_s[\tilde{e}]_1$
$e_1 = (b_7, b_7, 1)$	$e_5 = (a_7, a_7, 1)$	$v_{b_7}[e_5] = [4, 1]$	$v_{b_7}[e_5] = [4, 1]$	$1 \leq 4$
	$e_6 = (a_7, b_7, 1)$	$v_{b_7}[e_6] = [4, 3]$	$v_{b_7}[e_6] = [4, 3]$	$3 \leq 4$
$e_2 = (b_7, a_7, 1)$	$e_5 = (a_7, a_7, 1)$	$v_{b_7}[e_5] = [4, 1]$	$v_{a_7}[e_5] = [3, 4]$	$4 \leq 4$
	$e_6 = (a_7, b_7, 1)$	$v_{b_7}[e_6] = [4, 3]$	$v_{a_7}[e_6] = [2, 1]$	$1 \leq 4$
$e_3 = (a_7, a_7, 2)$	$e_7 = (b_7, b_7, 2)$	$v_{a_7}[e_7] = [4, 1]$	$v_{a_7}[e_7] = [4, 1]$	$1 \leq 4$
	$e_8 = (b_7, a_7, 2)$	$v_{a_7}[e_8] = [4, 3]$	$v_{a_7}[e_8] = [4, 3]$	$3 \leq 4$
$e_4 = (a_7, b_7, 2)$	$e_7 = (b_7, b_7, 2)$	$v_{a_7}[e_7] = [4, 1]$	$v_{b_7}[e_7] = [3, 4]$	$4 \leq 4$
	$e_8 = (b_7, a_7, 2)$	$v_{a_7}[e_8] = [4, 3]$	$v_{b_7}[e_8] = [2, 1]$	$1 \leq 4$

Table 6.4 Lyapunov inequalities for the edges in the graph $\mathcal{G}$ for the Example 6.32

According to the rules for the definition of the vectors v_s for $s \in S$, we have that

$$\begin{aligned} v_{a_7} &= [v_{a_7}[e_5],\ v_{a_7}[e_6],\ v_{a_7}[e_7],\ v_{a_7}[e_8]]^\top, \\ &= [3,\ 4,\ 2,\ 1,\ 4,\ 1,\ 4,\ 3]^\top, \\ \text{and }\ v_{b_7} &= [v_{b_7}[e_5],\ v_{b_7}[e_6],\ v_{b_7}[e_7],\ v_{b_7}[e_8]]^\top, \\ &= [4,\ 1,\ 4,\ 3,\ 3,\ 4,\ 2,\ 1]^\top. \end{aligned}$$

Then, we can first check whether each Lyapunov inequalities encoded by the edges of the graph are satisfied. We know that for an edge (s, d, σ), we have to look at the components of $v_d(e_2)$ and $v_s(e_2)$ for each edge $e_2 \in \overline{E_7}$ of label σ. So, for each edge, we have only two inequalities to check. We can see in Table 6.4 that all these inequalities are satisfied.

Now, let's see whether the Lyapunov inequalities encoded by the unexisting edges, i.e. e_5, e_6, e_7 and e_8, are violated. As shown in the proof, this is always the edge itself that leads to the violation of the inequality. For instance, the edge e_5 is not satisfied because of the inequality associated to e_5. The same holds for e_6, e_7 and e_8. Table 6.5 summarizes the inequalities and shows that none of them is satisfied. △

We are now in position to prove the main result in Theorem 6.30.

Proof of Theorem 6.30. (1) $\Rightarrow$ (2): We have to prove that $\mathcal{G} \leq \widetilde{\mathcal{G}}$. By definition, this means that we have to prove that as soon as $\mathcal{G}$ admits a solution

Edge $e=(s,d,i)$	Edges $\tilde{e}=(s_2,d_2,i)$	Vector $v_s[\tilde{e}]$	Vector $v_d[\tilde{e}]$	Inequality $v_d[\tilde{e}]_2 \leq v_s[\tilde{e}]_1$
$e_5=(a_7,a_7,1)$	$e_5=(a_7,a_7,1)$	$v_{a_7}[e_5]=[3,4]$	$v_{a_7}[e_5]=[3,4]$	$4>3$
	$e_6=(a_7,b_7,1)$	$v_{a_7}[e_6]=[2,1]$	$v_{a_7}[e_6]=[2,1]$	$1\leq 2$
$e_6=(a_7,b_7,1)$	$e_5=(a_7,a_7,1)$	$v_{a_7}[e_5]=[3,4]$	$v_{b_7}[e_5]=[4,1]$	$1\leq 3$
	$e_6=(a_7,b_7,1)$	$v_{a_7}[e_6]=[2,1]$	$v_{b_7}[e_6]=[4,3]$	$3>2$
$e_7=(b_7,b_7,2)$	$e_7=(b_7,b_7,2)$	$v_{b_7}[e_7]=[3,4]$	$v_{b_7}[e_7]=[3,4]$	$4>3$
	$e_8=(b_7,a_7,2)$	$v_{b_7}[e_8]=[2,1]$	$v_{b_7}[e_8]=[2,1]$	$1\leq 2$
$e_8=(b_7,a_7,2)$	$e_7=(b_7,b_7,2)$	$v_{b_7}[e_7]=[3,4]$	$v_{a_7}[e_7]=[4,1]$	$1\leq 3$
	$e_8=(b_7,a_7,2)$	$v_{b_7}[e_8]=[2,1]$	$v_{a_7}[e_8]=[4,3]$	$3>2$

Table 6.5 Lyapunov inequalities for the unexisting edges in the graph $\mathcal{G}$ for the Example 6.32

for a system F in a template $\mathcal{V}$, then so does $\widetilde{\mathcal{G}}$. Consider a system F of a family $\mathcal{F}$, and a template $\mathcal{V}$. Assume that there exists an admissible solution $\{V_s : s \in S\} \in \mathcal{V}^S$ for F and $\mathcal{G}$. Since $\mathcal{G}$ simulates $\widetilde{\mathcal{G}}$ through a function $R : \widetilde{S} \to S$ by assumption, let us prove that

$$\{W_{\tilde{s}} := V_{R(\tilde{s})} : \tilde{s} \in \widetilde{S}\}$$

is admissible for $\widetilde{\mathcal{G}}$ and F. Consider any edge $(\tilde{s}, \tilde{d}, i) \in \widetilde{E}$ and the corresponding Lyapunov inequality

$$\begin{aligned} &\forall x \in \mathbb{R}^n : W_{\tilde{d}}(f_i(x)) \leq W_{\tilde{s}}(x) \\ \Leftrightarrow \quad &\forall x \in \mathbb{R}^n : V_{R(\tilde{d})}(f_i(x)) \leq V_{R(\tilde{s})}(x) \end{aligned}$$

by definition of the functions $W_{\tilde{s}}$. By simulation, the edge $(R(\tilde{s}), R(\tilde{d}), i) \in E$ and therefore this Lyapunov inequality is satisfied by assumption.

$(2) \Rightarrow (1)$: First, let us apply Lemma 6.31 to the graph $\mathcal{G}$. We obtain a system $\{f_i : i \in \langle M\rangle\}$ and a set of candidate Lyapunov functions $\{V_s : s \in S\}$ such that the Lyapunov inequalities encoded by the edges of $\mathcal{G}$ are satisfied and none of the non-existing edges of $\mathcal{G}$ are satisfied, i.e.

$$\forall (p_1, q_1, i) \in E,\ \forall x \in \mathbb{R}^n : V_{q_1}(f_i(x)) \leq V_{p_1}(x) \tag{6.25}$$

$$\forall (p_1, q_1, i) \in \overline{E},\ \exists \bar{x} \in \mathbb{R}^n : V_q(f_i(\bar{x})) > V_s(\bar{x}) \tag{6.26}$$

Let us define the family with a single system $\mathcal{F} = \{F\}$ and the template $\mathcal{V} := \{V_s : s \in S\}$. Obviously, there exists a solution for $\mathcal{G}$ and F in $\mathcal{V}$. Then, by assumption, there exists a set of candidate Lyapunov functions which are admissible for F and $\widetilde{\mathcal{G}}$ in the template $\mathcal{V}$, i.e. there exist $\{U_{\tilde{s}} : \tilde{s} \in \widetilde{S}\} \in \mathcal{V}^{\tilde{S}}$ that satisfy the Lyapunov inequalities encoded by $\widetilde{\mathcal{G}}$. Since these functions belong to the template $\mathcal{V}$, we can associate a node of $\mathcal{G}$ to each node of $\widetilde{\mathcal{G}}$, i.e. we can define a function $R : \widetilde{S} \to S$ such that $U_{\tilde{s}} = V_{R(\tilde{s})}$. Finally, we just have to prove that this function R satisfies the definition of simulation, i.e.

$$\forall(\tilde{p}, \tilde{q}, i) \in \widetilde{E},\ (R(\tilde{p}), R(\tilde{q}), i) \in E.$$

Assume by contradiction that there exists $(\tilde{p}, \tilde{q}, i) \in \widetilde{E}$ such that the edge $(R(\tilde{p}), R(\tilde{q}), \sigma) \in \overline{E}$. Using the Lemma 6.31, this means that there exists $\overline{x} \in \mathbb{R}^n$ such that

$$U_{\tilde{q}}(\overline{x}) \ := \ V_{R(\tilde{q})}(f_i(\overline{x})) > V_{R(\tilde{p})}(\overline{x}) \ := \ U_{\tilde{p}}(\overline{x})$$

i.e. the set $\{U_{\tilde{s}} : \tilde{s} \in \widetilde{S}\}$ is not admissible. However the set $\{V_s : s \in S\}$ is admissible by construction. Here is the contradiction. $\square$

The most trivial simulation relation once again concerns the common Lyapunov function graph $\mathcal{G}_0$. One can easily prove that any graph $\mathcal{G} = (S, E)$ is simulated by $\mathcal{G}_0$ by considering the function $R(s) := s_0$ for every node $s \in S$. Then, $\mathcal{G}_0 \leq \mathcal{G}$ for every path-complete graph $\mathcal{G}$, as already stated in Proposition 6.24. Let us now take a more complex example.

Example 6.33. We consider the path-complete graphs $\mathcal{G}_4$ and $\mathcal{G}_7^\top$, the dual of $\mathcal{G}_7$. We define the function $R : S_4 \to S_7$ such that $R(a_4) = R(d_4) := a_7$ and $R(b_4) = R(c_4) := b_7$. We can prove that the function R satisfies the expression (6.19) , and therefore that $\mathcal{G}_7^\top$ simulates $\mathcal{G}_4$. For instance, for the edge $e := (c_4, d_4, 2) \in E_4$, the corresponding edge denoted by $R(e) := (b_7, a_7, 2)$ belongs to $E_7^\top$.

By Theorem 6.30, this simulation relation implies that whatever the system and the template that we use, the JSR approximation provided by $\mathcal{G}_4$ is better than or equal to the JSR approximation provided by $\mathcal{G}_7^\top$. For instance, let us consider the switched system $\mathcal{A}_1$ in Example 6.18. Numerical results match with theoretical results since both approximations coincide with primal copositive norms and quadratic functions, i.e. $\rho_{\mathcal{G}_4,\mathcal{P}}(\mathcal{A}_1) = \rho_{\mathcal{G}_7^\top,\mathcal{P}}(\mathcal{A}_1) := 1.5487$ and $\rho_{\mathcal{G}_4,\mathcal{Q}}(\mathcal{A}_1) = \rho_{\mathcal{G}_7^\top,\mathcal{Q}}(\mathcal{A}_1) := 1.3534$. $\triangle$

6.3.4 Limitations of the characterization

As discussed in [PJ19], Theorem 6.30 states that the general ordering relation (6.12) in Definition 6.21 is associated to a combinatorial property, notably the simulation. However, when it is not possible to establish a simulation relation, i.e. when there exist at least one template $\mathcal{V}$ and one family $\mathcal{F}$ such that the inequality (6.9) is not satisfied, it might still be possible to compare graphs with the relations in Equations (6.9) and (6.11). In practice, this can result in wiser choices of template for the stability analysis in the sense of Definition 6.21.

Example 6.34. Consider the graphs $\mathcal{G}_9 = (S_9, E_9)$ and $\mathcal{G}_7 = (S_7, E_7)$ in Figures 6.6a and 6.8 respectively.

One can easily verify that $\mathcal{G}_9$ does not simulate $\mathcal{G}_7$. Indeed, we cannot define a relation $R : S_7 \to S_9$ with $R(a) \in S_9$ such that the edge $(R(a), R(a), 1) \in E_9$ since $\mathcal{G}_9$ does not have any loop. By Theorem 6.30, it means that $\mathcal{G}_9 \not\leq \mathcal{G}_7$ in the sense of Definition 6.21.

However, one can easily prove that for any template $\mathcal{V}$ closed under addition (as formally defined in Definition 1.33), the inequality

$$\mathcal{G}_9 \leq_{\mathcal{V}} \mathcal{G}_7$$

holds. Indeed, let $\{V_{p_9}, V_{q_9}, V_{r_9}\} \in \mathcal{V}^{S_9}$ be admissible for $\mathcal{G}_9$ and a given switched system $F := \{f_i : i \in \langle 2 \rangle\}$. Define the Lyapunov functions $W_{a_7} := V_{q_9} + V_{r_9}$ and $W_{b_7} := V_{p_9} + V_{q_9}$. One can easily prove that the set $\{W_{a_7}, W_{b_7}\} \in \mathcal{V}^{S_7}$ is admissible for $\mathcal{G}_7$ and F. For example, the Lyapunov inequality

$$\forall x \in \mathbb{R}^n : \underbrace{V_{p_9}(f_1(x)) + V_{q_9}(f_1(x))}_{:= W_{a_7}(f_1(x))} \leq \underbrace{V_{q_9}(x) + V_{r_9}(x)}_{:= W_{b_7}(x)},$$

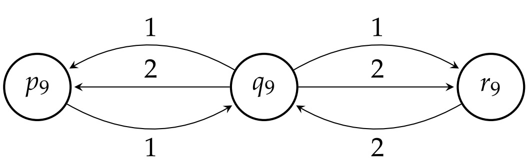


Fig. 6.8 The path-complete graph $\mathcal{G}_9 = (S_9, E_9)$ in Example 6.34. This graph does not simulate $\mathcal{G}_7$ in Figure 6.6a but satisfies that $\mathcal{G}_9 \leq_{\mathcal{V}} \mathcal{G}_7$ for any template $\mathcal{G}$ closed under addition.

encoded by the edge $(b_7, a_7, 1) \in E_7$, holds because the Lyapunov inequalities encoded by the edges $(p_9, q_9, 1)$ and $(q_9, r_9, 1) \in E_9$ are satisfied by the functions $\{V_{p_9}, V_{q_9}, V_{r_9}\}$ by assumption. It implies in particular that the inequality holds for the quadratic Lyapunov functions. $\triangle$

6.4 Summary

The purpose of this chapter is to motivate multipe Lyapunov functions and then introduce the Path-complete Lyapunov formalism. In particular, we have summarized the literature on the comparison of path-complete Lyapunov stablity certificates.

Summary of Chapter 6

This chapter introduces the path-complete Lyapunov stability certificates and motivates their conservatism-based comparison. In order to delve deeper into this question in the following chapters, we summarize the state of the art.

Section 6.1: Introduction to multiple Lyapunov functions

After identifying the limitations of common Lyapunov functions, we use an example to introduce the notion of multiple Lyapunov functions, which involve a finite set of candidate Lyapunov functions whose joint decreasing properties guarantee stability.

Section 6.2: Graph-based Lyapunov functions

We formally introduce the path-complete Lyapunov formalism, that is the generalization of the multiple Lyapunov approach. This framework relies on two components: a path-complete graph and a template.

Section 6.3: Comparison of PCLFs

This section introduces the three levels of comparison of path-complete graphs, and recalls the simulation-based characterization of the general ordering relation. After detailing the demonstration, as it will be useful to us later, we identify the limitations of this characterization result.

The following chapters pursue similar simulation-based characterizations of template-dependent ordering of graphs for a family of template sharing a commun closure property, such as the addition, the pointwise minimum or maximum.

7

Template-dependent lifts and closure properties

FOR the purpose of understanding the relations between different multiple Lyapunov functions structures, *path-complete Lyapunov functions* have been proposed as a unifying and flexible approach. While this framework is now mature enough to provide effective criteria to approximate the joint spectral radius in the linear subsystems case, the following open question naturally arises: *How to systematically compare different path-complete graph structures*? More precisely, given a template $\mathcal{V} \subseteq \mathcal{C}(\mathbb{R}^n, \mathbb{R})$ and any path-complete graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, we want to provide a condition ensuring whether or not the conditions arising from $\mathcal{G}_1$ are less/more conservative than the ones given by $\mathcal{G}_2$ if we restrict our search for a solution in $\mathcal{V}$. Such a result is a crucial challenge in the path-complete stability criteria framework, since it would provide, as side product, formal confidence intervals for the decay rate and, in practice, could guide the user, given a particular system setting, to choose the path-complete structure in a proper and smart way, see [Phi17].

A partial result is provided in Theorem 6.30 stated in [PJ19], without any hypothesis on the candidate Lyapunov functions template $\mathcal{V}$, relying on the notion of simulation of graphs. On the other hand it has been observed, since the introduction of this framework [AJPR14], that the order relations between path-complete stability criteria strongly depend on the chosen set of candidate Lyapunov functions. Although it may be counter-intuitive, some examples have shown that increasing the size of the graph does not improve the stability certificate if we consider a particular family of Lyapunov functions. On another note, sufficient conditions [PEDJ15] have already been provided in the context of *constrained switched systems*, and they rely on combinatorial operations on graphs called *lifts* that maintain the path-completeness. While these previous abstract lifts were introduced in order to reduce the conservatism of the arising stability condition, in this chapter instead we understand how these tools can be used to induce the comparison of path-complete graphs in the sense that all the known comparison relations can be expressed in terms of lift. In this chapter, we introduce formal transformations of graphs, called *template-dependent lifts*, in order to improve the performance of a path-complete criterion (by enlarging the underlying graph) following particular rules which rely on closure properties of $\mathcal{V}$.

Note that this work has been done in collaboration with Matteo Della Rossa, and gathers the results which have been published in [DJ20, DDJ21, DDJ22a].

First, we make the following assumption on path-complete graphs for this chapter.

Assumption 7.1. *The path-complete graphs considered herein have one strongly connected component and are such that if we remove any edge, the graph is not path-complete.*

This in particular implies that all nodes admit at least one incoming edge and one outgoing edge. This assumption is not restrictive since our aim is to compare stability conditions: we suppose that the inequalities of the form (6.5) encoded in the graphs are sufficient conditions for stability (path-completeness) without having redundant/unnecessary inequalities.

7.1 Introduction to the lifts and their validity

We develop in this section several expansions of graphs, called *lifts*. The goal of a (valid) lift is to generate a better graph, in the sense of Definition 6.21.

Definition 7.2 (Lift)**.** Given $M \in N$, we denote with $Graphs_M$ the set of directed and labeled graphs on the alphabet $\langle M \rangle$. A function $L : Graphs_M \to Graphs_M$ is a *lift* if for any path-complete graph $\mathcal{G}$, $L(\mathcal{G})$ is path-complete.

In short, we will say that a lift *preserves the path-completeness of graphs*.

Some examples of lifts have already been introduced [PEDJ15, Definitions 2 & 3] in the path-complete Lyapunov framework with the aim of improving the accuracy of the stability criteria but without exploiting the particular properties of the considered candidate Lyapunov functions template. In our case, instead, we want to use them as tools to provide further insight about the order relations in Definition 6.21, and in particular in Equations (6.9) and (6.11). Thus we have the following definitions.

Definition 7.3 (Valid lift)**.** We say that a lift $L : Graphs_M \to Graphs_M$ is:

(a) *valid with respect to a template $\mathcal{V}$ and a family $\mathcal{F}$* if for any path-complete graph $\mathcal{G}$,

$$\mathcal{G} \ \leq_{\mathcal{V},\mathcal{F}} \ L(\mathcal{G}).$$

(b) *valid with respect to a template $\mathcal{V}$* if for any path-complete graph $\mathcal{G}$,

$$\mathcal{G} \ \leq_{\mathcal{V}} \ L(\mathcal{G}).$$

(c) *valid* if for any path-complete graph $\mathcal{G}$,

$$\mathcal{G} \ \leq \ L(\mathcal{G}).$$

To be consistent with Theorem 6.30 that characterizes the general inequality (6.12), a lift is valid if and only if there exists a simulation relation between $\mathcal{G}$ and $L(\mathcal{G})$. This is the case for both T-product and M-path-dependent lifts defined in [PEDJ15], for instance.

In this work, we are particularly interested in the order relations (6.9) and (6.11) and therefore, we focus our study on lifts that are valid with respect to a template (and a family) as in Definition 7.3(a) and (b). Indeed, quadratic functions are *closed under addition*. By this, we mean that the sum of two quadratic functions of a fixed dimension can also be expressed as a quadratic function. It turns out that this property is key for the relation (6.11). More generally, we will show that such a closure property allows us to define lifts that are valid in a specific setting, even though they are not valid in general (i.e. in the sense of Definition 6.21).

In this chapter, we study the consequences of the closure properties of a given template on the path-complete stability certificates. In what follows, we introduce three *template-dependent lifts*, that are lifts whose validity depends on the template properties.

7.2 Duality of lifts

In this section, we leverage the duality developed in Section 6.3.2 to introduce the notion of *dual lift*. We define this dual operation such that it is valid with respect to the dual template, as stated in Proposition 7.5.

Definition 7.4. Consider any lift $L : Graphs_M \to Graphs_M$. The *dual lift* of L, denoted by $L^\star$, is defined as

$$L^*(\mathcal{G}) := \left[L(\mathcal{G}^\top)\right]^\top, \tag{7.1}$$

for any path-complete graph $\mathcal{G}$ on $\langle M \rangle$.

The duality in Definition 7.4 is *involutory*, i.e. applying twice the duality produces the original graph:

$$(L^*)^*(\mathcal{G}) = \left(L^*(\mathcal{G}^\top)\right)^\top = \left(L(\mathcal{G})^\top\right)^\top = L(\mathcal{G}).$$

The validity of a lift and its dual can be linked using the following result.

Proposition 7.5. *Consider a lift $L : Graphs_M \to Graphs_M$ valid with respect to the family of linear systems and a template $\mathcal{V}$. Then, the dual lift $L^* : Graphs_M \to Graphs_M$ is valid with respect to the family of linear switched systems and the dual template $\mathcal{V}^*$, and reversely. In other words,*

$$[\forall \mathcal{G} = (S, E),\ \mathcal{G} \leq_{\mathcal{V},\mathcal{L}} L(\mathcal{G})] \Leftrightarrow [\forall \mathcal{G} = (S, E),\ \mathcal{G} \leq_{\mathcal{V}^*,\mathcal{L}} L^*(\mathcal{G})].$$

Proof. Consider a path-complete graph $\mathcal{G}$ on the alphabet $\langle M \rangle$. Since the lift L is valid with respect to $\mathcal{V}$ and $\mathcal{L}$, the ordering holds in particular for $\mathcal{G}^\top$, i.e.

$$\mathcal{G}^\top \leq_{\mathcal{V},\mathcal{L}} L(\mathcal{G}^\top).$$

Using Proposition 6.27, we conclude that

$$\mathcal{G} \leq_{\mathcal{V}^\star,\mathcal{L}} \left[L(\mathcal{G}^\top)\right]^\top := L^*(\mathcal{G}).$$

The reverse implication can be derived similarly. □

In particular, Proposition 7.5 implies that if a lift L is valid with respect to any template closed under a binary operation $\star$, the dual lift $L^\star$ is valid with respect to any template closed under the corresponding dual operation of $\star$ (see Lemma 1.27).

7.3 Sum lift

As first and meaningful example of binary operation, we consider the *addition* under which many usual templates are closed, as quadratic functions, convex functions or sum-of-squares polynomials for example. Therefore, in what follows, we define the *T-sum lift*, which explores the existing relations between *sums* of T functions/nodes of the initial graph. The *sum lift*, defined as the union over $\mathbb{N}$ of all the T-sum lifts, in turn exploits the addition property, regardless of the number of terms in the addition.

Let us first start with an example where the sum-closure is needed to be able to compare two path-complete graphs.

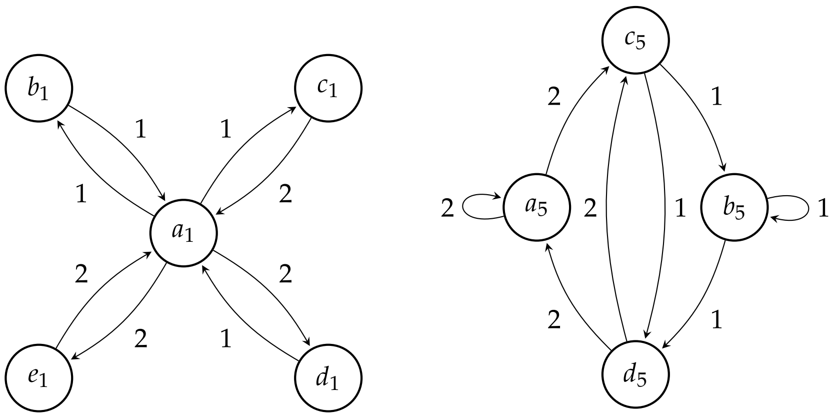
Example 7.6. Consider two path-complete graphs $\mathcal{G}_1 = (S_1, E_1)$ and $\mathcal{G}_{db}^{3,1}$ in Figures 6.5a and 7.1 respectively. We recall the graph $\mathcal{G}_{db}^{3,1} = (S_{db}^{3,1}, E_{db}^{3,1})$ in Figure 7.1b for simplicity.

One can prove that $\mathcal{G}_1$ does not simulate $\mathcal{G}_{db}^{3,1}$: the graph $\mathcal{G}_{db}^{3,1}$ admits the loop $(b_5, b_5, 1)$ but $\mathcal{G}_1$ does not admit any loop. Therefore, it is impossible to associate the node b_5 to a node $R(b_5) \in S_1$ such that the loop $(R(b_5), R(b_5), 1) \in E_1$.

We suppose now that the template $\mathcal{V}$ is closed under addition. Given a switched system F with 2 modes and a solution $V_{S_1} := \{V_s : s \in S_1\} \in \mathcal{V}^{S_1}$ admissible for $\mathcal{G}_1$ and F, one can build a solution $W_{S_{db}^{3,1}} \in \mathcal{V}^{S_{db}^{3,1}}$ admissible for $\mathcal{G}_{db}^{3,1}$ and F by defining

$$W_{S_{db}^{3,1}} := \begin{cases} W_{a_5} & := & V_{a_1} + V_{e_1}, \\ W_{b_5} & := & V_{a_1} + V_{b_1}, \\ W_{c_5} & := & V_{a_1} + V_{d_1}, \\ W_{d_5} & := & V_{a_1} + V_{c_1}. \end{cases} \tag{7.2}$$

Let us take for instance the edge $(d_5, c_5, 2) \in E_{db}^{3,1}$. The inequality encoded



(a) The path-complete graph $\mathcal{G}_1 = (S_1, E_1)$.

(b) $\mathcal{G}_{db}^{3,1}$, the generalized De Bruijn graph of order 2 and memory 1.

Fig. 7.1 The path-complete graphs $\mathcal{G}_1$ and $\mathcal{G}_{db}^{3,1}$ in Example 7.6. Even though the relation $\mathcal{G}_1 \leq \mathcal{G}_{db}^{3,1}$ does not hold, $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_{db}^{3,1}$ for any template $\mathcal{V}$ closed under addition.

by this edge, i.e.

$$\forall x \in \mathbb{R}^n,\ \underbrace{V_{a_1}(f_2(x)) + V_{d_1}(f_2(x))}_{:= W_{c_5}(f_2(x))} \leq \underbrace{V_{a_1}(x) + V_{c_1}(x)}_{:= W_{d_5}(x)},$$

is satisfied since $(a_1, d_1, 2)$ and $(c_1, a_1, 2) \in E_1$. In terms of ordering introduced in Definition 6.21, this means that the graph $\mathcal{G}_{db}^{3,1}$ is $\mathcal{V}$-greater than $\mathcal{G}_1$ for any template $\mathcal{V}$ closed under addition, i.e. $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_{db}^{3,1}$. △

In this section, we first introduce the notion of *multi-set* and a few operations on them in order to define the sum lift (see Section 7.3.1). Then we provide a characterization of the edges of the sum lift in Section 7.3.3 using the notion of *perfect matching* in graph theory. Similarly to Theorem 6.30, we provide a sufficient condition based on both the simulation and the sum lift for the ordering of graphs with respect to the class of templates closed under addition. In Section 7.3.5, we finally discuss the possible values of T for the T-sum lift, and characterizes the simulation by the sum lift as a linear program. Throughout this section, we illustrate most of the results with numerous examples.

7.3.1 Definition and properties

Given a path-complete graph, we want to build the lift which encodes the relations between a sum of a finite number of functions/nodes. To this aim, given a set S and $T \in \mathbb{N}$, we denote by $\mathrm{Multi}^T(S)$ the set of *multi-sets* of cardinality T with elements in S , where a multi-set is defined as a set with possible repetitions, see [Bli88]. Note that equality of multi-sets is independent of the ordering. The number of repetitions of an element s in a multi-set P is called the *multiplicity* of this element, denoted by $\mathrm{m}_P(s)$.

First, we define a few operations on multi-sets.

Definition 7.7 (Operations on multi-sets)**.** Consider two multi-sets A and B of the same universe S, and their multiplicity functions $m_A(\cdot)$ and $m_B(\cdot)$ respectively. We define

(a) The *inclusion*: A is included in B, denoted by $A \subseteq B$ if

$$\forall s \in S,\ m_A(s) \leq m_B(s).$$

(b) The *union*: the union of A and B, denoted by $A \cup B$, is the multi-set C with the multiplicity function $m_C(\cdot)$ defined by

$$\forall s \in S,\ m_C(s) := \max\left\{m_A(s),\, m_B(s)\right\}.$$

(c) The *intersection*: the intersection of A and B, denoted by $A \cap B$, is the multi-set C with the multiplicity function $m_C(\cdot)$ defined by

$$\forall s \in S,\ m_C(s, S) := \min\left\{m_A(s),\, m_B(s)\right\}.$$

(d) The *sum*: the sum of A and B, denoted by $A \oplus B$, is the multi-set C with the multiplicity function $m_C(\cdot)$ defined by

$$\forall s \in S,\ m_C(s) := m_A(s) + m_B(s).$$

(e) The *multiplication by an integer*: given $n \in \mathbb{N}$, the product of n and A, denoted by $n \otimes A$, is the multi-set C with the multiplicity function $m_C(\cdot)$ defined as the sum of n times the multi-set A, i.e.

$$\forall s \in S,\ m_C(s) := n \times m_A(s).$$

It is possible to state the following properties about the distributivity of the multiplication over the addition.

Proposition 7.8. *Consider two multi-set A and B of a universe S, and $\alpha, \beta \in \mathbb{N}$. The following statements hold:*

(a) $(\alpha \otimes A) \oplus (\beta \otimes A) = (\alpha + \beta) \otimes A$,

(b) $\alpha \otimes (A \oplus B) = (\alpha \otimes A) \oplus (\alpha \otimes B)$.

Proof. (a) By definition and by the associativity of the addition in Proposition 7.8, we have

$$\begin{aligned} (\alpha \otimes A) \oplus (\beta \otimes A) &= \left(\underbrace{A \oplus \cdots \oplus A}_{\alpha \text{ times}}\right) \oplus \left(\underbrace{A \oplus \cdots \oplus A}_{\beta \text{ times}}\right) \\ &= (\alpha + \beta) \otimes A. \end{aligned}$$

(b) Similarly, by the definition of the multiplication by an integer and the associativity of the addition, we have

$$\begin{aligned} \alpha \otimes (A \oplus B) &= \underbrace{(A \oplus B) \oplus \cdots \oplus (A \oplus B)}_{\alpha \text{ times}}, \\ &= \left(\underbrace{A \oplus \cdots \oplus A}_{\alpha \text{ times}}\right) \oplus \left(\underbrace{B \oplus \cdots \oplus B}_{\alpha \text{ times}}\right), \\ &= (\alpha \otimes A) \oplus (\alpha \otimes B). \end{aligned}$$

□

Let us now define the *T-sum lift* for any integer value $T \in \mathbb{N}$ and the *sum lift* as the union of all the T-sum lifts.

Definition 7.9. Consider $T \in \mathbb{N}$ and a graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$.

(a) The *T-sum lift* of $\mathcal{G}$, denoted by $\mathcal{G}^{\oplus T} = (S^{\oplus T}, E^{\oplus T})$, is defined as follows :

(1) The set of nodes $S^{\oplus T}$ is defined by

$$S^{\oplus T} := \mathrm{Multi}^T(S).$$

(2) For each $i \in \langle M \rangle$ and each multi-set of edges of E of the form $D := \{(a_1, b_1, i), \ldots, (a_T, b_T, i)\}$ such that the multi-sets $A := \{a_1, \ldots, a_T\}$ and $B := \{b_1, \ldots, b_T\} \in S^{\oplus T}$, the edge $(A, B, i) \in E^{\oplus T}$.

(b) The *sum lift* of $\mathcal{G}$, denoted by $\mathcal{G}^{\oplus} = (S^{\oplus}, E^{\oplus})$, is defined as the infinite disjoint union of the T-sum lifts, i.e.

$$\mathcal{G}^{\oplus} := \bigcup_{T \in \mathbb{N}} \mathcal{G}^{\oplus T}. \tag{7.3}$$

Note that each T-sum lift is finite but the sum lift is an infinite graph, i.e. there is an infinitely countable number of nodes and edges. Moreover, in practice we will only consider the multi-sets of cardinality T of nodes for which we can reach and leave each node with the same label: this ensures that all the nodes of $\mathcal{G}^{\oplus T}$ have at least one incoming and one outgoing edge. On the other hand, observe that $\mathcal{G}^{\oplus T}$ might not satisfy Assumption 7.1, even if $\mathcal{G}$ does, i.e. $\mathcal{G}^{\oplus T}$ is possibly composed by more than one strongly connected and path-complete component, as illustrated in the subsequent Example 7.12. In practice, we will consider each of these components independently, recall Proposition 6.23.

First, let us demonstrate that both the sum lift and the T-sum lifts satisfy the definition of a lift. The first requirement is the preservation of the path-completeness of graphs.

Proposition 7.10. *For any $T \in \mathbb{N}$, the sum lift and the T-sum lift preserve the path-completeness of graphs.*

Proof. Given $T \in \mathbb{N}$, the path-completeness of $\mathcal{G}^{\oplus T}$ is direct since $\mathcal{G}$ is a strongly connected component of $\mathcal{G}^{\oplus T}$. Indeed, each node $a \in S$ admits an outgoing and an incoming edge thanks to the Assumption 7.1. This implies that the node $A := \{a, \ldots, a\} \in S^{\oplus T}$ with $m_A(a) = T$, and then for every edge $(a, b, i) \in E$, the edge $(\{a, \ldots, a\}, \{b, \ldots, b\}, i) \in E^{\oplus T}$.

The path-completeness of $\mathcal{G}^{\oplus}$ directly follows since each of the strongly connected components is path-complete. □

We now have to discuss the validity of the *sum lift*.

Proposition 7.11. *Consider $T \in \mathbb{N}$ and the family of binary operations $\{+_n\}_{n \in \mathbb{N}}$ such that $\forall n \in \mathbb{N}$, $+_n$ corresponds to the addition. The T-sum lift is valid in the sense of Definition 7.3 with respect to any template closed under $\{+_n\}_{n \in \mathbb{N}}$.*

Proof. Consider a path-complete graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, a template $\mathcal{V}$ of candidate Lyapunov functions closed under addition and any family $\mathcal{F} := \{f_i : i \in \langle M \rangle\}$. Suppose that there exists a set of functions $\{V_s : s \in S\} \in \mathcal{V}^S$ admissible for $\mathcal{G}$ and $\mathcal{F}$, and for any $A = \{a_1, \dots, a_T\} \in S^{\oplus T}$ define

$$W_A \; := \; V_{a_1} + \; \dots \; + V_{a_T} \in \mathcal{V}. \tag{7.4}$$

The Lyapunov inequalities in Equation (6.5) of $\mathcal{G}^{\oplus T}$ are satisfied because, for every edge $(A, B, i) \in E^{\oplus T}$, we have

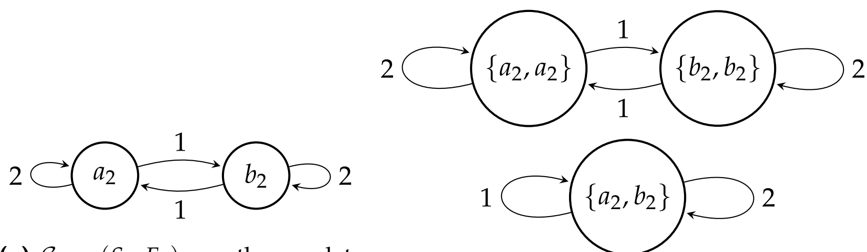
$$\begin{aligned} W_B\left(f_i(x)\right) &= \left(V_{b_1}(f_i(x)) + \dots + V_{b_T}(f_i(x))\right), \\ &\leq \left(V_{a_1}(x) + \dots + V_{a_T}(x)\right) \; := \; W_A(x) \end{aligned}$$

for all $x \in \mathbb{R}^n$ since $(a_1, b_1, i), \dots, (a_T, b_T, i) \in E$ by Definition 7.9 (possibly after a re-ordering of A and B). $\square$

Example 7.12. Consider the path-complete graph $\mathcal{G}_2 = (S_2, E_2)$ on the alphabet $\langle M \rangle := \{1, 2\}$ in Figure 7.2a, and apply the 2-sum lift in Definition 7.9 to $\mathcal{G}_2$. The outcome is provided in Figure 7.2b. As expected, the lifted graph $(\mathcal{G}_2)^{\oplus 2}$ admits three nodes, one for each multi-set of cardinality 2 of the initial set of nodes $S_2 = \{a_2, b_2\}$, i.e. $\text{Multi}^2(S_2) = \{\{a_2, a_2\}, \{a_2, b_2\}, \{b_2, b_2\}\}$. By Proposition 7.11, we know that for any template $\mathcal{V}$ closed under addition, the inequality

$$\mathcal{G}_2 \; \leq_{\mathcal{V}} \; (\mathcal{G}_2)^{\oplus 2}$$

holds. By Proposition 6.23, this inequality is also verified for the two path-complete and strongly connected components of $(\mathcal{G}_2)^{\oplus 2}$. As reported in



(a) $\mathcal{G}_2 = (S_2, E_2)$, a path-complete graph over the alphabet $\langle 2 \rangle$.

(b) $(\mathcal{G}_2)^{\oplus 2}$, the 2-sum lift of $\mathcal{G}_2$.

Fig. 7.2 Example of the 2-sum lifted graph of a path-complete graph.

the proof of Proposition 7.10, one of the components induced by the nodes $\{a_2, a_2\}$ and $\{b_2, b_2\}$ is isomorphic to the graph $\mathcal{G}_2$ itself. The second one induced by the node $\{a_2, b_2\}$ is isomorphic to the common Lyapunov function graph $\mathcal{G}_0$ since the node associated to $\{a_2, b_2\}$ admits one loop for each mode. So, Propositions 6.23 and 7.11 imply together that

$$\mathcal{G}_2 \ \leq_{\mathcal{V}} \ \mathcal{G}_0$$

for any template $\mathcal{V}$ closed under addition. Moreover, by Proposition 6.24, we know that the reverse inequality holds for any template and any switched system. In particular,

$$\mathcal{G}_0 \ \leq_{\mathcal{V}} \ \mathcal{G}_2$$

for any template $\mathcal{V}$ closed under addition. We have thus proved that the graphs $\mathcal{G}_2$ and $\mathcal{G}_0$ are equivalent in the sense of the order relation (6.11) for any template closed under addition. In practice, it means that given such a template $\mathcal{V}$ and a switched system F, either both graphs $\mathcal{G}_2$ and $\mathcal{G}_0$ admit a solution admissible for $\mathcal{V}$ and F, or none of them. That is, the inequalities encoded in $\mathcal{G}_2$ are as conservative as the ones encoded in $\mathcal{G}_0$. △

7.3.2 Duality

Let us finally discuss the duality of the sum lift. The following proposition states that the sum lift and each of its finite components that are the T-sum lifts are self-dual.

Proposition 7.13. *For any $T \in \mathbb{N}$, the T-sum lift is* self-dual, *i.e. the dual lift of the T-sum lift is the T-sum lift itself. Then, for any graph $\mathcal{G}$,*

$$\mathcal{G}^{\oplus T} \ = \ \left(\left(\mathcal{G}^{\top} \right)^{\oplus T} \right)^{\top}.$$

As a consequence, the sum lift is self-dual too.

Proof. Given an integer value $T \in \mathbb{N}$ and a graph $\mathcal{G}$, let us take an edge $e = (P, Q, j) \in E^{\oplus T}$. By Definition 7.9, this means that there exists $p_1, \ldots, p_T$ and $q_1, \ldots, q_T \in S$ such that $(p_k, q_k, j) \in E$ for $k = 1, \ldots, T$ with $j \in \langle M \rangle$. By duality, the edges $(q_k, p_k, j) E^{\top}$. Then, by definition of the T-sum lift, the edge $(Q, P, j) \in \left(E^{\top}\right)^{\oplus T}$. Equivalently, the dual edge $(P, Q, j) \in \left(\left(E^{\top}\right)^{\oplus T}\right)^{\top}$, which ends the proof. □

Using Proposition 7.5, this means that the same construction not only exploits the closure operation of addition by construction but also the dual operation, that is the inverse summation.

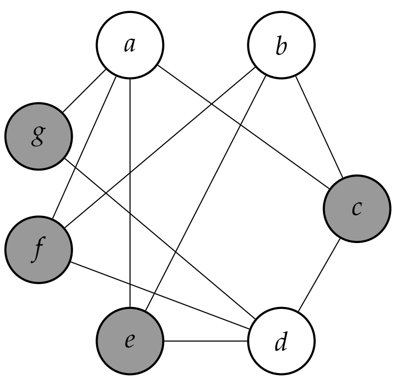
7.3.3 Characterization of the edges of the sum lift using graph theory

The initial definition of the edges of the T-sum lift in Definition 7.9 is constructive and involves the existence of T edges (which might be similar or not) in the graph. This condition might be impractical to check in practice because it requires to verify all the possible combinations of T edges. In this section, we provide a characterization in Proposition 7.16 for an edge to belong to the T-sum lift using graph theory. In particular, this result relies on the notion of *perfect matching* in a *bipartite graph*.

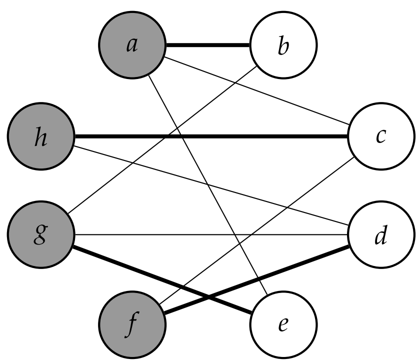
To begin with, let us take a formal look at these two concepts. Intuitively, a graph is bipartite if you are able to split the set of nodes into two groups without internal edges.

Definition 7.14. A *bipartite graph* $\mathcal{G} = (S, E)$ is an (unlabeled and undirected) graph whose nodes S can be divided into two disjoint sets U and V such that each edge admits an extremity in U and the other in V. One often writes $\mathcal{G} = (U, V, E)$ to underline the partition of the set of nodes.

Figures 7.3a and 7.3b provide two examples of bipartite graphs.



(a) Example of a bipartite graph where the two disjoint sets U and V are respectively the nodes in white and gray.



(b) Example of a perfect matching (in bold) in a bipartite graph where the two disjoint sets U and V are respectively in white and gray.

Fig. 7.3 Illustration of **(a)** a bipartite graph and **(b)** a perfect matching in a bipartite graph.

Definition 7.15. Given a bipartite graph $\mathcal{G} = (X, Y, E)$, an *X-perfect matching* is a set of edges without common vertices which covers every node of X. A *perfect matching* of $\mathcal{G}$ is an X-perfect matching and a Y-perfect matching (then $|X| = |Y|$).

Deciding whether there exists a perfect matching and finding it is a well-known problem in graph theory. It is known that it can be solved in polynomial time, and any maximum cardinality matching algorithm can be used. Figure 7.3b provides an example of a perfect matching in a bipartite graph.

We can now derive the characterization of an edge in the T-sum lift.

Proposition 7.16. *Consider a path-complete graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, an integer $T \in \mathbb{N}$, two multi-sets P and Q of S of cardinality T and $i \in \langle M \rangle$. The edge (P, Q, i) is an element of $E^{\oplus T}$ if and only if there exists a perfect matching in the bipartite graph $(P, Q, E_i(P, Q))$ where $E_i(P, Q)$ refers to the restriction of E to the edges of label i starting in P and ending in Q.*

Note that in Proposition 7.16 we are committing an abuse of notation since P and Q are multi-sets by definition. Formally, we should consider the bipartite graph $(\tilde{P}, \tilde{Q}, E_i(\tilde{P}, \tilde{Q}))$ where $\tilde{P}$ and $\tilde{Q}$ refer to the conversion of P and Q into sets (by distinguishing the repeated elements).

Proof. Consider a path-complete graph $\mathcal{G} = (S, E)$ over the alphabet $\langle M \rangle$, $T \in \mathbb{N}$, $i \in \langle M \rangle$ and two multi-sets P and Q of S of cardinality T. Recalling Definition 7.9, the edge $(P, Q, i) \in E^{\oplus T}$ if and only if there exists a multi-set of edges of E, denoted by $D := \{(a_1, b_1, i), \ldots, (a_T, b_T, i)\} \subseteq E$ such that $P := \{a_1, \ldots, a_T\}$ and $Q := \{b_1, \ldots, b_T\}$. Thus, D provides a perfect matching between P and Q composed by edges of $\mathcal{G}$ labeled by i. $\square$

Let us consider an example to see how Proposition 7.16 works in practice.

Example 7.17. Consider the graph $\mathcal{G}_5 = (S_5, E_5)$ in Figure 7.4. One can prove that the 8-sum lift of this graph admits a component isomorphic to the common Lyapunov function graph with 2 modes $\mathcal{G}_0$. And therefore, ${\mathcal{G}_5}^{\oplus 8}$ simulates $\mathcal{G}_0$. Indeed, if we consider the node

$$\tilde{S} := \{a_5,\ a_5,\ b_5,\ c_5,\ d_5,\ d_5,\ e_5,\ e_5\} \in {E_5}^{\oplus 8},$$

we can prove that both loops $(\tilde{S}, \tilde{S}, 1) \in {E_5}^{\oplus 8}$ and $(\tilde{S}, \tilde{S}, 2) \in {E_5}^{\oplus 8}$. As a result, the relation $R : S_0 \to {S_5}^{\oplus 8}$ where $R(s_0) := \tilde{S}$ is a simulation relation.

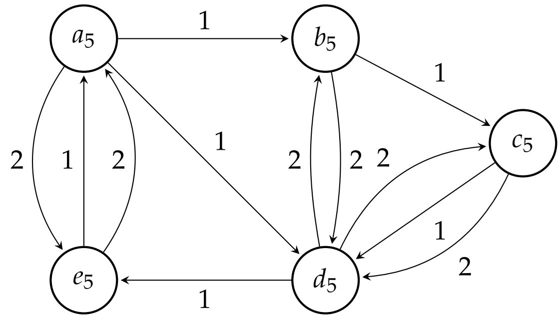
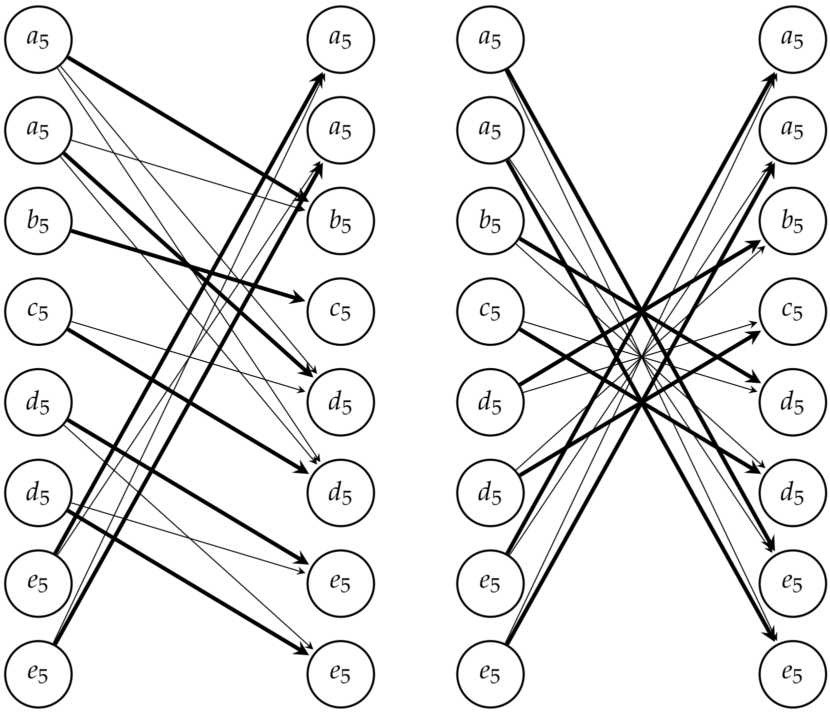


Fig. 7.4 $\mathcal{G}_5 = (S_5, E_5)$, a path-complete graph over the alphabet $\langle 2 \rangle$ which can be compared with the common Lyapunov functions graph $\mathcal{G}_0$ with a multi-set of cardinality 8.



(a) Bipartite graph $(\tilde{S}, \tilde{S}, (E_5)_1 (\tilde{S}, \tilde{S})$ and perfect matching for the edge $(\tilde{S}, \tilde{S}, 1)$.

(b) Bipartite graph $(\tilde{S}, \tilde{S}, (E_5)_2 (\tilde{S}, \tilde{S})$ and perfect matching for the edge $(\tilde{S}, \tilde{S}, 2)$.

Fig. 7.5 Illustration of the perfect matching in Proposition 7.16 for the edges $(\tilde{S}, \tilde{S}, 1)$ in **(a)** and $(\tilde{S}, \tilde{S}, 2) \in E_5{}^{\oplus 8}$ in **(b)** with $\tilde{S} := \{a_5, a_5, b_5, c_5, d_5, d_5, e_5, e_5\}$ in Example 7.17.

Let us first consider the edge $(\tilde{S}, \tilde{S}, 1)$. Using Definition 7.9, this edge belongs to the 8-sum lift because the multi-set of edges $\{(a_5, b_5, 1),\ (a_5, d_5, 1),\ (b_5, c_5, 1),\ (c_5, d_5, 1),\ (d_5, e_5, 1), (d_5, e_5, 1), (e_5, a_5, 1), (e_5, a_5, 1)\}$ is a subset of E_5. Using the characterization in Proposition 7.16, we can check that there exists a perfect matching in the bipartite graph $(\tilde{S}, \tilde{S}, (E_5)_1\,(\tilde{S}, \tilde{S}))$. This is illustrated in Figure 7.5a. Similarly, we can find a multi-set of 8 edges with label 2 of E_5 to prove that the edge $(\tilde{S}, \tilde{S}, 2) \in E^{\oplus 8}$. The perfect matching in the corresponding bipartite graph is illustrated in Figure 7.5b.

This means that, given a switched system F, if one can find a set of Lyapunov functions $\{V_{a_5}, V_{b_5}, V_{c_5}, V_{d_5}, V_{e_5}\}$ admissible for $\mathcal{G}_5$ and F, then the function

$$W_0 \ := \ 2V_{a_5} + V_{b_5} + V_{c_5} + 2V_{d_5} + 2V_{e_5}$$

is a common Lyapunov function for F. Moreover, if we consider a linear switched system and an addition-closed template, both graphs $\mathcal{G}_5$ and $\mathcal{G}_0$ provide the same JSR approximation, although $\mathcal{G}_5$ may seem more complex and therefore better at first glance. △

This result will prove useful in Chapter 8 to characterize the ordering with respect to addition-closed templates with the simulation.

7.3.4 Simulation-based sufficient condition for the template-dependent ordering of graphs

In this section, we make use of the lift approach to provide a *sufficient* condition for the comparison of path-complete graphs, focusing on templates closed under the binary operation of addition. We show that, similarly to the previously mentioned general case solved in Theorem 6.30, the sufficient condition is given by a simulation relation. However in this setting, the simulation is more involved, as it considers the sum lift of the graph $\mathcal{G}$ (for proving the relation $\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ for any template $\mathcal{V}$ closed under addition), not the graph $\mathcal{G}$ itself. Note that the necessary condition will be proved in Chapter 8.

Theorem 7.18. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet. If $\mathcal{G}^{\oplus}$ simulates $\widetilde{\mathcal{G}}$, then*

$$\mathcal{G} \ \leq_{\mathcal{V}} \ \widetilde{\mathcal{G}}$$

for any template $\mathcal{V}$ closed under addition.

Proof. Consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ on the same alphabet. By Proposition 7.11, the inequality

$$\mathcal{G} \leq_{\mathcal{V}} \mathcal{G}^{\oplus}$$

is satisfied for any template $\mathcal{V}$ closed under addition. By assumption and recalling the simulation-based characterisation Theorem 6.30 in [PJ19, Theorem 3.5],

$$\mathcal{G}^{\oplus} \leq \widetilde{\mathcal{G}}.$$

Then, by transitivity of the ordering (see Equation (6.17) in page 131 for more details) , we have

$$\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$$

for this class of templates. □

Remark 7.19. Theorem 7.18 recalls [Phi17, Theorem 6.6] where the solution of $\widetilde{\mathcal{G}}$ is also expressed as the sum of the pieces of the solution for $\mathcal{G}$. In this case, they use the notion of simulation recalled in Remark 6.29 which involves *subsets* of S_1 rather than *multisets*. However, as it will be discussed in Example 8.14, the use of multi-sets is crucial to be able to capture all the comparison of graphs with respect to the class of templates closed under addition. △

Although Theorem 7.18 provides a simulation-based sufficient condition of the template-ordering (6.11) of graphs for the family of templates closed under addition, the simulation relation involves an infinite graph and then cannot be checked numerically. The following lemma points out that when a path-complete graph $\widetilde{\mathcal{G}}$ is simulated by the sum lift of another path-complete graph $\mathcal{G}$, as it is the case in Theorem 7.18, we can restrict the simulation to a (finite) T-level of the sum lift for which the number of terms in the sum is fixed at $T \in \mathbb{N}$.

Lemma 7.20. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$. The following statements are equivalent:*

(1) $\mathcal{G}^{\oplus}$ simulates $\widetilde{\mathcal{G}}$.

(2) $\exists T \in \mathbb{N}$ such that $\mathcal{G}^{\oplus T}$ simulates $\widetilde{\mathcal{G}}$.

Proof. By Assumption 7.1, we assume that $\widetilde{\mathcal{G}}$ is strongly connected. Otherwise, the argument is valid for each strongly connected component of the graph. More precisely, each strongly connected component $\widetilde{\mathcal{H}} \subseteq \widetilde{\mathcal{G}}$ can be associated to a value of $T(\widetilde{\mathcal{H}})$ for which the simulation holds. Moreover, any integer multiple of $T(\widetilde{\mathcal{H}})$ also satisfies the simulation for $\widetilde{\mathcal{H}}$. Thus, taking the least common multiple of the $T(\widetilde{\mathcal{H}})$, for all the strongly connected components $\widetilde{\mathcal{H}}$ of $\widetilde{\mathcal{G}}$, we can conclude.

The implication (2) $\Rightarrow$ (1) is direct since $S^{\oplus T} \subseteq S^{\oplus}$. For the reverse implication (1) $\Rightarrow$ (2), first we note that the sum lift, introduced in Definition 7.9, is the union of a countable number of strongly connected components, since it is the union of all the T-sum lifts, for any $T \in \mathbb{N}$. We now suppose that $\mathcal{G}^{\oplus}$ simulates $\widetilde{\mathcal{G}}$ via a function $R : \widetilde{S} \to S^{\oplus}$, as given by Definition 6.28. Since by hypothesis $\widetilde{\mathcal{G}}$ is strongly connected, by the properties of simulation, the nodes $\{R(\tilde{s})\}_{\tilde{s}\in\widetilde{S}}$ are strongly connected in $\mathcal{G}^{\oplus}$. Thus, $\{R(\tilde{s})\}_{\tilde{s}\in\widetilde{S}}$ lie in the same strongly connected component of $\mathcal{G}^{\oplus}$ i.e. there exists a $T \in \mathbb{N}$ such that $\{R(\tilde{s})\}_{\tilde{s}\in\widetilde{S}}$ are nodes of $\mathcal{G}^{\oplus T}$. Thus, we have shown that $\mathcal{G}^{\oplus T}$ simulates $\widetilde{\mathcal{G}}$, concluding the proof. □

Example 7.21. Let us consider the two path-complete graphs $\mathcal{G}_1$ and $\mathcal{G}_{db}^{3,1}$ in Example 7.6 where we have shown that $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_{db}^{3,1}$ for any template closed under addition. In fact, one can prove that the Equation (7.2) induces a simulation relation between $\mathcal{G}_{db}^{3,1}$ and $(\mathcal{G}_1)^{\oplus 2}$. In practice, we define the relation $R : S_{db}^{3,1} \to (S_1)^{\oplus 2}$ where $R(a_5) := \{a_1, e_1\}$, $R(b_5) := \{a_1, b_1\}$, $R(c_5) := \{a_1, d_1\}$ and $R(d_5) := \{a_1, c_1\}$. We can prove that R is a simulation relation. Let us consider for instance the edge $(c_5, d_5, 1) \in E_{db}^{3,1}$. We have to prove that the edge $(R(c_5), R(d_5), 1) := (\{a_1, d_1\}, \{a_1, c_1\}, 1)$ belongs to the 2-sum lift of $\mathcal{G}_1$. Since the edges $(a_1, c_1, 1)$ and $(d_1, a_1, 1) \in E_1$, the edge $(\{a_1, d_1\}, \{a_1, c_1\}, 1) \in (E_1)^{\oplus 2}$. A similar argument can be used for all the other edges of $\mathcal{G}_{db}^{3,2}$. △

7.3.5 Numerical characterization of the simulation

Although the simulation by the sum lift in Theorem 7.18 cannot be verified numerically, Lemma 7.20 provides us with a semi-algorithm in order to check the simulation by the sum lift. By iteratively checking condition (2) in Lemma 7.20 above for increasing $T \in N$, we get a sufficient condition for the simulation of $\widetilde{\mathcal{G}}$ by the sum lift. However, it remains unclear whether the algorithm will stop since we do not have a stopping condition to upperbound the value of T in Lemma 7.20. Either there exists $T \in \mathbb{N}$ such that $\mathcal{G}^{\oplus T}$ simulates $\widetilde{\mathcal{G}}$ and the algorithm stops, or $\mathcal{G}^{\oplus}$ does not simulate $\widetilde{\mathcal{G}}$ and the algorithm runs without stopping, increasing the value of T at each iteration. In what follows, we would like to provide an algorithm which numerically checks whether $\mathcal{G}^{\oplus}$ simulates $\widetilde{\mathcal{G}}$ or not.

First, we would like to summarize preliminary results that will help us determine the values of T that can be considered and the ones that should eventually be removed. Our thinking is initially guided by the path-completeness of the graphs: this property necessarily implies the existence of a loop or a cycle in the graph for each label. We will see how these graph structures can influence the possible values of T where there is a simulation relation involving the sum lift. Formally, we consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ on the same alphabet $\langle M \rangle$. We assume that the sum lift of $\mathcal{G}$ simulates $\widetilde{\mathcal{G}}$, or equivalently that there exists $T \in \mathbb{N}$ such that $\mathcal{G}^{\oplus T}$ simulates $\widetilde{\mathcal{G}}$ by Lemma 7.20. We denote by $R : \widetilde{S} \to S^{\oplus T}$ the corresponding simulation relation.

Given a label $m \in \langle M \rangle$, we start with the simplest case, i.e. when there exists a loop $(a_m, a_m, m) \in \widetilde{E}$ of label m in the simulated graph $\widetilde{\mathcal{G}}$. By simulation, $(R(a_m), R(a_m), m) \in E^{\oplus T}$ where we denote by

$$R(a_m) \;:=\; \{r_1, r_2, \dots, r_T\}$$

the corresponding multi-set of S of cardinality T, i.e. $r_i \in S$ for $s = 1, \dots, T$. By the characterization in Proposition 7.16, there exists a perfect matching between $R(a_m)$ and itself, which corresponds to a permutation of $R(a_m)$. Therefore if we consider $r_1 \in R(a_m)$, there exists $r_{i_1} \in R(a_m)$ such that $(r_1, r_{i_1}, m) \in E$. Similarly, there exists $r_{i_2} \in R(a_m)$ such that $(r_{i_1}, r_{i_2}, m) \in E$, and so on. At some point, we will reach back r_1 (the number of r_j needed is related to the order of the permutation, and is at least smaller than T). This means that if there exists a loop of label m in $\widetilde{\mathcal{G}}$, there must exist a cycle of label m in $\mathcal{G}$. So, in practice, we should look at the length of all the cycles of label m in $\mathcal{G}$ to have insights about the possible values of T.

Formally, let us denote n_m, the number of cycles of label m of $\mathcal{G}$ and $\{T_{m,k}\}_{k \in \langle n_m \rangle}$ the set of lengths of those cycles. Therefore, T must be either a multiple of one of the $T_{m,k}$'s, or a multiple or a combination of them, i.e.

$$T \;\in\; \left\{ \sum_{k=1}^{n_m} \alpha_k T_{m,k} \;\middle|\; \alpha_k \in \mathbb{N}, k \in \langle n_m \rangle \right\}.$$

Let now assume that $\widetilde{\mathcal{G}}$ admits more than one loop. In this case, T must satisfy all the constraints encoded by the loops. Therefore,

$$T \in \bigcap_{(a,a,m)\in\widetilde{E}} \left\{ \sum_{k=1}^{n_m} \alpha_k T_{m,k} \;\middle|\; \alpha_k \in \mathbb{N}, k \in \langle n_m \rangle \right\}.$$

We can finally derive the following necessary condition on the possible values of T.

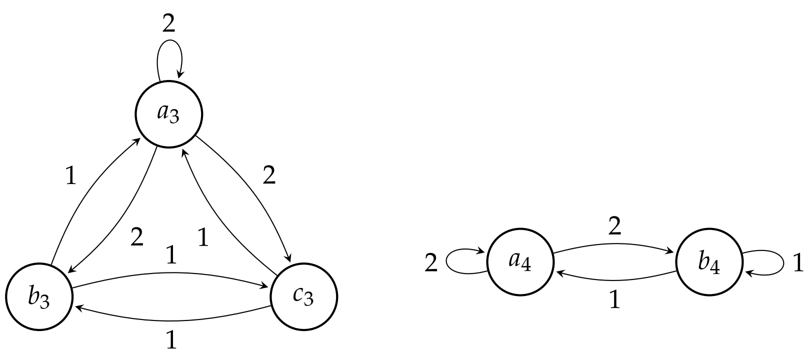
Proposition 7.22. *Consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ over the same alphabet $\langle M \rangle$. Assume that there exists $T \in \mathbb{N}$ such that $\mathcal{G}^{\oplus T}$ simulates $\widetilde{\mathcal{G}}$. Then, the following expression must be satisfied:*

$$T \in \bigcap_{(a,a,m)\in\widetilde{E}} \left\{ T := \sum_{k=1}^{n_m} \alpha_k T_{m,k} \;\middle|\; \alpha_k \in \mathbb{N}, k \in \langle n_m \rangle \right\}, \qquad (7.5)$$

where n_m denotes the number of cycles $\{T_{m,k}\}_{k\in\langle n_m \rangle}$ of label m in $\mathcal{G}$.

Then, the smallest possible value that we should check in practice is the minimal value T in Equation (7.5).

Example 7.23. Consider the graphs $\mathcal{G}_3$ and $\mathcal{G}_4$ in Figures 7.6a and 7.6b respectively. We know that ${\mathcal{G}_3}^{\oplus 2}$ simulates $\mathcal{G}_4$ by the relation $R(a_4) := \{a_3, a_3\}$ and $R(b_4) := \{b_3, c_3\}$. The graph $\mathcal{G}_4$ admits two loops, one for each label. Let us apply Proposition 7.22 to identify the potential values of T.



(a) $\mathcal{G}_3 = (S_3, E_3)$, a path-complete graph over the alphabet $\langle 2 \rangle$.

(b) $\mathcal{G}_4 = (S_4, E_4)$, a strongly connected and path-complete component of ${\mathcal{G}_3}^{\oplus 2}$.

Fig. 7.6 Example of two path-complete graphs $\mathcal{G}_3$ and $\mathcal{G}_4$ on the alphabet $\langle 2 \rangle$, such that ${\mathcal{G}_3}^{\oplus 2}$ simulates $\mathcal{G}_4$.

$\boxed{(a_4, a_4, 2) \in E_4}$

There is a single cycle of label 2 in $\mathcal{G}_3$, i.e.

$$(a_3, a_3, 2) \in E_3.$$

Then, $T_{2,1} = 1$, and T must be a multiple of 1 (this does not add any constraints on the value of T).

$\boxed{(b_4, b_4, 1) \in E_4}$

There is a single cycle of label 1 in $\mathcal{G}_3$, i.e.

$$(c_3, b_3, 1) \in E_3,$$
$$(b_3, c_3, 1) \in E_3.$$

Then, $T_{1,1} = 2$, and T can only be a multiple of 2.

Therefore, T must be a multiple of 2, and the minimal value is 2, which is consistent since ${\mathcal{G}_3}^{\oplus 2}$ simulates $\mathcal{G}_4$. △

Example 7.24. Consider the graphs $\mathcal{G}_6 = (S_6, E_6)$ and $\mathcal{G}_4 = (S_4, E_4)$ in Figures 7.6b and 7.7 respectively. We can prove that ${\mathcal{G}_6}^{\oplus 2}$ simulates $\mathcal{G}_4$ by the relation $R(b_4) := \{a_6, b_6\}$ and $R(a_4) := \{b_6, c_6\}$.

Let us use Proposition 7.22 to identify the potential values of T. The graph $\mathcal{G}_4$ admits two loops, one for each label.

$\boxed{(b_4, b_4, 1) \in E_4}$

There is a single cycle of label 2 in $\mathcal{G}_6$, i.e.

$$(b_6, a_6, 1) \in E_6,$$
$$(a_6, b_6, 1) \in E_6.$$

Then, $T_{1,1} = 2$, and T must be a multiple of 2.

$\boxed{(a_4, a_4, 2) \in E_4}$

There is a single cycle of label 1 in $\mathcal{G}_6$, i.e.

$$(b_6, c_6, 2) \in E_8,$$
$$(c_6, b_6, 2) \in E_8.$$

Then, $T_{2,1} = 2$, and T can only be a multiple of 2.

Therefore, T must be a multiple of 2, and the minimal value is 2, which is consistent since ${\mathcal{G}_6}^{\oplus 2}$ simulates $\mathcal{G}_4$. △

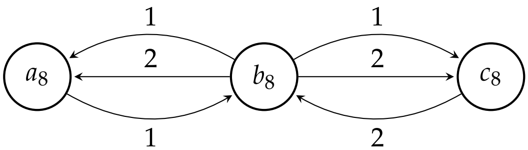


Fig. 7.7 $\mathcal{G}_6 = (S_6, E_6)$, a path-complete graph on the alphabet $\langle 2 \rangle$. This graph is used to illustrate the constraints on the potential value of T for simulation of $\mathcal{G}_4$ by ${\mathcal{G}_6}^{\oplus T}$ using Proposition 7.22.

Let us focus in particular when the simulated graph is the common Lyapunov function graph, i.e. $\widetilde{\mathcal{G}} := \mathcal{G}_0$. This question is useful, because if it appears to be the case, then we know that any other path-complete graph is simulated by the T-sum lifted graph. Since there is no other edge than the self-loops, we will be able to derive a necessary and sufficient condition for the possible values of T.

Theorem 7.25. *Given a graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$ and $\mathcal{G}_0$, the common Lyapunov function graph on M modes. There exists $T \in \mathbb{N}$ such that $\mathcal{G}^{\oplus T}$ simulates $\mathcal{G}_0$ if and only if there exist $T \in \mathcal{N}$ and $\tilde{S} \in Multi^T(S)$ such that for any label $i \in \langle M \rangle$, there exist N_i cycles of label i denoted by $\{\mathcal{C}_k^{(i)}\}_{k=1,\ldots,N_i}$ such that*

$$\bigoplus_{k=1,\cdots,N_i} \left\{ nodes\left(\mathcal{C}_k^{(i)}\right)\right\} = \tilde{S},$$

where $N_i \in \mathbb{N}$ and nodes $(\mathcal{C})$ refers to the nodes involved in a cycle $\mathcal{C}$.

Proof. $\boxed{\Rightarrow}$ Consider $T \in \mathbb{N}$ such that $R : \{s_0\} \to S^{\oplus T}$ is a simulation relation, i.e. $(R(s_0), R(s_0), i) \in E^{\oplus T}$ for every label $i \in \langle M \rangle$. By definition of the T-sum lift, this means that there exist as many permutations (i.e. bijection) as the number of modes $p_i : R(s_0) \to R(s_0)$ defined by the edges of label i. Therefore, both permutations can be decomposed into cycles of label i in the graph $\mathcal{G}$ with all the nodes of $R(s_0)$.

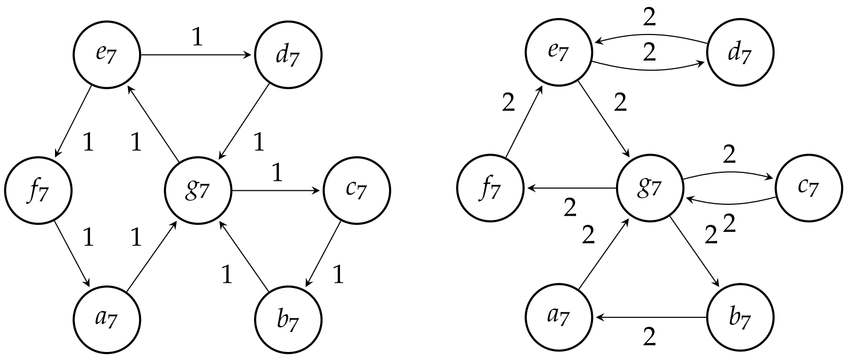
$\boxed{\Leftarrow}$ Consider $\tilde{S} := \{i_1, i_2, \ldots, i_T\} \subseteq S$ a subset of nodes of $\mathcal{G}$ which admits a (group of) cycle(s) of label i for every label $i \in \langle M \rangle$, i.e. there exists $\{j_1^{(i)}, \ldots, j_T^{(i)}\}$ such that $(i_{j_l^{(i)}}, i_{j_{l+1}^{(i)}}, 1) \in E$ for $l = 1, \ldots, T-1$ and $(i_{j_T^{(i)}}, i_{j_1^{(i)}}, 1) \in E$. Therefore, the edges $(\tilde{S}, \tilde{S}, i) \in E^{\oplus T}$ since the functions $p_i : \tilde{S} \to \tilde{S}$ defined by the cycles are permutations. $\square$

Let us take an example to illustrate this theorem.

Example 7.26. Consider the graph $\mathcal{G}_7 = (S_7, E_7)$ on 2 modes in Figure 7.8, where the edges of label 1 and 2 (denoted by $E_7^{(1)}$ and $E_7^{(2)}$) are illustrated in Figures 7.8a and 7.8b respectively. Using Theorem 7.25, the structure of $\mathcal{G}_7^{(1)} := (S_7, E_7^{(1)})$ implies that the admissible values of T_1 for $\mathcal{G}_7{}^{\oplus T_1}$ to admit a loop of label 1 are

$$T_1 \in \{\alpha \times 4 + \beta \times 3 + \gamma \times 3 \mid \alpha, \beta, \gamma \in \mathbb{N} \text{ not all zero}\}$$

because $\mathcal{G}_7^{(1)}$ admits 3 cycles of length 4,3 and 3. In particular the multi-set



(a) $\mathcal{G}_7^{(1)}$, the restriction of $\mathcal{G}_7$ to the edges of label 1.

(b) $\mathcal{G}_7^{(2)}$, the restriction of $\mathcal{G}_7$ to the edges of label 2.

Fig. 7.8 $\mathcal{G}_7 = (S_7, E_7)$, a path-complete graph on $\langle 2 \rangle$ in Example 7.26. By Theorem 7.25, we can find that ${\mathcal{G}_7}^{\oplus 10}$ simulates $\mathcal{G}_0$.

$\tilde{S}^{(1)}$ (i.e. node) associated to this loop has to be of the form

$$
\begin{aligned}
\tilde{S}^{(1)} &= (\alpha \otimes \{a_7, e_7, f_7, g_7\}) \oplus (\beta \otimes \{d_7, e_7, g_7\}) \\
&\quad \oplus (\gamma \otimes \{b_7, c_7, g_7\}), \\
&= (\alpha \otimes (\{a_7, f_7\} \oplus \{e_7\} \oplus \{g_7\})) \oplus (\beta \otimes (\{d_7\} \oplus \{e_7\} \oplus \{g_7\})) \\
&\quad \oplus (\gamma \otimes (\{b_7, c_7\} \oplus \{g_7\})), \\
&= (\alpha \otimes \{a_7, f_7\}) \oplus (\alpha \otimes \{e_7\}) \oplus (\alpha \otimes \{g_7\}) \oplus (\beta \otimes \{d_7\}) \\
&\quad \oplus (\beta \otimes \{e_7\}) \oplus (\beta \otimes \{g_7\}) \oplus (\gamma \otimes \{b_7, c_7\}) \oplus (\gamma \otimes \{g_7\}), \\
&= (\alpha \otimes \{a_7, f_7\}) \oplus (\beta \otimes \{d_7\}) \oplus (\gamma \otimes \{b_7, c_7\}) \\
&\quad \oplus ((\alpha + \beta) \otimes \{e_7\}) \oplus ((\alpha + \beta + \gamma) \otimes \{g_7\}),
\end{aligned}
$$

where the last equalities have been obtained by Proposition 7.8 and where α, β and $\gamma \in \mathbb{N}$ are not all null. Similarly, the admissible values of T_2 such that ${\mathcal{G}_7}^{\oplus T_2}$ admits a loop of label 2 are

$$
T_2 \in \{\delta \times 2 + \varepsilon \times 3 + \zeta \times 2 + \eta \times 3 \mid \delta, \varepsilon, \zeta, \eta \in \mathbb{N} \text{ not all zero}\},
$$

and the multi-set $\tilde{S}^{(2)}$ has to be of the form

$$
\begin{aligned}
\tilde{S}^{(2)} &= (\delta \otimes \{d_7, e_7\}) \oplus (\varepsilon \otimes \{e_7, f_7, g_7\}) \oplus (\zeta \otimes \{c_7, g_7\}) \\
&\quad \oplus (\eta \otimes \{a_7, b_7, g_7\}),
\end{aligned}
$$

$$\begin{aligned} &= (\delta \otimes \{d_7\}) \oplus (\varepsilon \otimes \{f_7\}) \oplus (\zeta \otimes \{c_7\}) \oplus (\eta \otimes \{a_7, b_7\}) \\ &\quad \oplus ((\delta + \varepsilon) \otimes \{e_7\}) \oplus ((\varepsilon + \zeta + \eta) \otimes \{g_7\}), \end{aligned}$$

where δ, ε, ζ and $\eta \in \mathbb{N}$ are not all zero.

Therefore, there exists a value $T \in \mathbb{N}$ such that ${\mathcal{G}_7}^{\oplus T}$ simulates $\mathcal{G}_0$ if and only if there exist some integer values (not all zero) for α, β, γ, δ, ε, ζ and η such that $\tilde{S}^{(1)} = \tilde{S}^{(2)}$, i.e. equivalently

$$\tilde{S}^{(1)} = \tilde{S}^{(2)} \Leftrightarrow \left\{ \begin{array}{lrcl} (\text{for } a_7) & \eta & = & \alpha, \\ (\text{for } b_7) & \eta & = & \beta, \\ (\text{for } c_7) & \zeta & = & \gamma, \\ (\text{for } d_7) & \delta & = & \beta, \\ (\text{for } e_7) & \delta + \varepsilon & = & \alpha + \beta, \\ (\text{for } f_7) & \varepsilon & = & \alpha, \\ (\text{for } g_7) & \varepsilon + \zeta + \eta & = & \alpha + \beta + \gamma, \end{array} \right. \Leftrightarrow \left\{ \begin{array}{rcl} \delta & = & \alpha, \\ \varepsilon & = & \alpha, \\ \zeta & = & \gamma, \\ \eta & = & \alpha, \\ \alpha & = & \beta. \end{array} \right.$$

This means that for any value of (α, β, γ) such that $\alpha = \beta$ (i.e. if (α, β, γ) belongs to the integer-span of $(1, 1, 0)$ and $(0, 0, 1)$), there exists $(\delta, \varepsilon, \zeta, \eta) = (\alpha, \alpha, \gamma, \alpha)$ such that $\tilde{S}^{(1)} = \tilde{S}^{(2)}$. Then, if we consider the multi-set $\tilde{S}$ such that $\tilde{S} = \tilde{S}^{(1)}$ for $(\alpha, \beta, \gamma) = (1, 1, 1)$ (which corresponds to $\tilde{S}^{(2)}$ for $(\delta, \varepsilon, \zeta, \eta) = (1, 1, 1, 1)$), we get

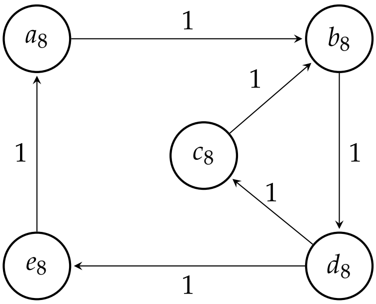
$$\tilde{S} := \{a_7,\ b_7,\ c_7,\ d_7,\ e_7,\ e_7,\ f_7,\ g_7,\ g_7,\ g_7\}$$

of cardinality 10, and this multi-set satisfies that both $(\tilde{S}, \tilde{S}, 1)$ and $(\tilde{S}, \tilde{S}, 2) \in {E_7}^{\oplus 10}$. This means that the 10-sum lift of $\mathcal{G}_7$ simulates $\mathcal{G}_0$. △

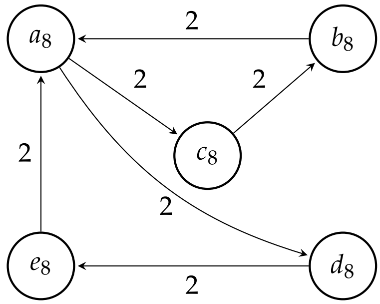
Example 7.27. Let us consider the graph $\mathcal{G}_8$ for which the restrictions to the edges of label 1 and 2, denoted by ${\mathcal{G}_8}^{(1)}$ and ${\mathcal{G}_8}^{(2)}$, are illustrated in Figures 7.9a and 7.9b respectively. We would like to prove that for any $T \in \mathbb{N}$, ${\mathcal{G}_8}^{\oplus T}$ does not simulate $\mathcal{G}_0$. To this aim, we use the characterization in Theorem 7.25. We can find that

$$T_1 \in \{\alpha \times 4 + \beta \times 3 \mid \alpha, \beta \in \mathbb{N} \text{ not all zero}\}$$

because ${\mathcal{G}_8}^{(1)}$ admits two cycles of length 4 and 3. Moreover, the multi-set



(a) $\mathcal{G}_8^{(1)}$, the restriction of $\mathcal{G}_8$ to the edges of label 1.



(b) $\mathcal{G}_8^{(1)}$, the restriction of $\mathcal{G}_8$ to the edges of label 2.

Fig. 7.9 $\mathcal{G}_8 = (S_8, E_8)$, a path-complete graph on $\langle 2 \rangle$. By Theorem 7.25, we can prove that there does not exist any value of $T \in \mathbb{N}$ such that $\mathcal{G}_8{}^{\oplus T}$ simulates $\mathcal{G}_0$.

$\tilde{S}^{(1)}$ must be of the form

$$\begin{aligned}\tilde{S}^{(1)} &= (\alpha \otimes \{a_8, b_8, d_8, e_8\}) \oplus (\beta \otimes \{b_8, c_8, d_8\}),\\ &= (\alpha \otimes \{a_8, e_8\}) \oplus (\beta \otimes \{c_8\}) \oplus ((\alpha+\beta) \otimes \{b_8, d_8\}),\end{aligned}$$

where α and $\beta \in \mathbb{N}$ are not both null. Similarly, for label 2, we get that

$$T_2 \in \{\gamma \times 3 + \delta \times 3 \mid \gamma, \delta \in \mathbb{N} \text{ not all zero }\}$$

because $\mathcal{G}_8{}^{(2)}$ admits two cycles of length 3. Moreover, the multi-set $\tilde{S}^{(2)}$ must be of the form

$$\begin{aligned}\tilde{S}^{(2)} &= (\gamma \otimes \{a_8, d_8, e_8\}) \oplus (\delta \otimes \{a_8, b_8, c_8\}),\\ &= (\gamma \otimes \{d_8, e_8\}) \oplus (\delta \otimes \{b_8, c_8\}) \oplus ((\gamma+\delta) \otimes \{a_8\}),\end{aligned}$$

where γ and $\delta \in \mathbb{N}$ are not both null. Therefore, there exists a value $T \in \mathbb{N}$ such that $\mathcal{G}_8{}^{\oplus T}$ simulates $\mathcal{G}_0$ if and only if there exists some integer values (not all zero) for α, β, γ and δ such that $\tilde{S}^{(1)} = \tilde{S}^{(2)}$, i.e. equivalently

$$\tilde{S}^{(1)} = \tilde{S}^{(2)} \Leftrightarrow \left\{\begin{array}{lrcl} (\text{for } a_8) & \alpha &=& \gamma+\delta, \\ (\text{for } b_8) & \alpha+\beta &=& \delta, \\ (\text{for } c_8) & \beta &=& \delta, \\ (\text{for } d_8) & \alpha+\beta &=& \gamma, \\ (\text{for } e_8) & \alpha &=& \gamma, \end{array}\right. \Leftrightarrow \left\{\begin{array}{rcl} \alpha &=& 0, \\ \beta &=& 0, \\ \gamma &=& 0, \\ \delta &=& 0, \end{array}\right.$$

since b_8 and c_8 imply that $\alpha = 0$, then $\gamma = 0$ by e_8. Then, $\beta = 0$ by d_8 and $\delta = 0$ by c_8. So we have proved that no value of T exists such that ${\mathcal{G}_8}^{\oplus T}$ simulates $\mathcal{G}_0$. $\triangle$

So far this approach only considers the loops, so we miss some edges, i.e. constraints on the possible value of T. We can generalize this reasoning to the cycles of $\widetilde{\mathcal{G}}$. This generalization is all the more useful since there must exist at least one cycle for each label by path-completeness of the graph.

Assume that there exists a cycle of label $m \in \langle M \rangle$ of length L_m in $\widetilde{\mathcal{G}}$:

$$\left.\begin{matrix} (a_1, a_2, m) \\ (a_2, a_3, m) \\ \vdots \\ (a_{L_m}, a_1, m) \end{matrix}\right\} \in \widetilde{E} \quad \longrightarrow \quad \left.\begin{matrix} (R(a_1), R(a_2), m) \\ (R(a_2), R(a_3), m) \\ \vdots \\ (R(a_{L_m}), R(a_1), m) \end{matrix}\right\} \in E^{\oplus T}.$$

Using a similar argument to the loop case, there exists a cycle of label m of length $L_m \times T$ in $\mathcal{G}$. Then, if we compute the length of all the cycles of label m in $\mathcal{G}$ denoted by $\{T_{m,k}\}$, the possible values are

$$\left\{ \sum_k \alpha_k \frac{T_{m,k}}{L_m} \;\middle|\; \frac{T_{m,k}}{L_m} \in \mathbb{N}, \alpha_k \in \mathbb{N} \right\}.$$

Since the graph is path-complete, we know that there exists at least one cycle of label m, but there could be several. We assume then that there exist several cycles of label m of length $\{L_{m,k}\}$. For each of these cycles, we can compute the length of the cycles of label m which are multiple of L_m in $\mathcal{G}$, denoted by $\{T_{m,k,n}\}$.

Proposition 7.28. *Consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ over the same alphabet $\langle M \rangle$. Assume that there exists $T \in \mathbb{N}$ such that $\mathcal{G}^{\oplus T}$ simulates $\widetilde{\mathcal{G}}$. Then, the following expression must be satisfied:*

$$T \in \bigcap_{m \in \langle M \rangle} \bigcap_k \left\{ \sum_n \alpha_n \frac{T_{m,k,n}}{L_{m,k}} \;\middle|\; \alpha_n \in \mathbb{N} \right\},$$

where $\{L_{m,k}\}_k$ denotes the lengths of all the cycles of label m in $\widetilde{\mathcal{G}}$, and for each of them, $\{T_{m,k,n}\}$ denotes the lengths of the cycles of label m in $\mathcal{G}$ which are a multiple of $L_{m,k}$.

Example 7.29. Let us consider the path-complete graphs $\mathcal{G}_2$ in Example 7.12 and recalled in Figure 7.10b, $\mathcal{G}_5 = (S_5, E_5)$ in Figure 7.4 and $\mathcal{G}_9 = (S_9, E_9)$ in Figure 7.10a. We have already shown in Example 7.17 that $\mathcal{G}^{\oplus 5}$ simulates $\mathcal{G}_0$. Let us now prove using Proposition 7.28 that $(\mathcal{G}_5)^{\oplus 3}$ simulates $\mathcal{G}_9$ and $(\mathcal{G}_5)^{\oplus 4}$ simulates $\mathcal{G}_2$.

First, let us find for each label the length of all the different cycles in $\mathcal{G}_5$.

Label 1

There are 2 cycles of label 1 in $\mathcal{G}_5$; one of length 3 i.e.

$$\mathcal{C}_1^1 := \{a_5, d_5, e_5\},$$

and one of length 5, i.e.

$$\mathcal{C}_2^1 := \{a_5, b_5, c_5, d_5, e_5\}.$$

Label 2

There are 3 cycles of label 2 in $\mathcal{G}_5$, all of them of length 2, i.e.

$$\mathcal{C}_1^2 := \{a_5, e_5\},$$

$$\mathcal{C}_2^2 := \{b_5, d_5\},$$

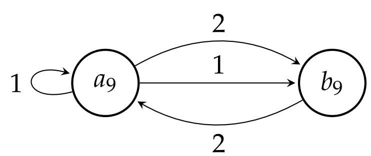
and

$$\mathcal{C}_3^2 := \{c_5, d_5\}.$$

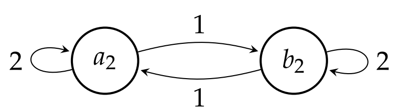
As first example, let us consider the graph $\mathcal{G}_9$. This graph admits one loop of label 1 and one cycle of length 2 for the label 2. Using Proposition 7.28, the possible values $T \in \mathbb{N}$ must satisfy that there exist c_1, c_2 and $c_3 \in \mathbb{N}$ such that

$$\begin{cases} 1 \times T &= 3 \times c_1 + 5 \times c_2, \\ 2 \times T &= 2 \times c_3. \end{cases} \tag{7.6}$$

The first equation guarantees the existence of a perfect matching for the self loop in $\mathcal{G}_9$ by combining the two cycles $\mathcal{C}_1^1$ and $\mathcal{C}_2^1$, while the second equation concerns the cycle of label 2 of length 2 in $\mathcal{G}_9$ and combines the three cycles of length 2 in $\mathcal{G}_5$. Note that even if these equations admit a



(a) $\mathcal{G}_9$, a path-complete graph simulated by $(\mathcal{G}_5)^{\oplus 3}$.



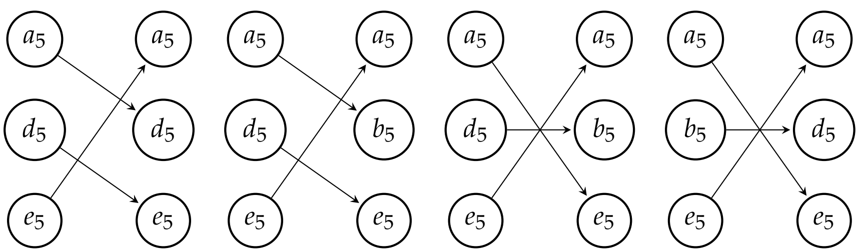
(b) $\mathcal{G}_2$, a path-complete graph simulated by $(\mathcal{G}_5)^{\oplus 4}$.

Fig. 7.10 Examples of two path-complete graphs simulated by the T-sum lift of $\mathcal{G}_5$ for **(a)** $T = 3$ and **(b)** $T = 4$.

solution for a specific T, nothing ensures that there exists a simulation relation between $\mathcal{G}_5{}^{\oplus T}$ and $\mathcal{G}_9$. However, if there exists a value $T \in \mathbb{N}$ such that the simulation relation is satisfied, then Proposition 7.28 ensures that there exists a solution to the equations in Equation (7.6). In our case, equations in (7.6) are satisfied for $T = 3$, $c_1 = 1$, $c_2 = 0$ and $c_3 = 3$. Let us prove that $\mathcal{G}_5{}^{\oplus 3}$ simulates $\mathcal{G}_9$ and let us progressively build the simulation relation $R : \mathcal{G}_9 \to \mathcal{G}_5{}^{\oplus 3}$.

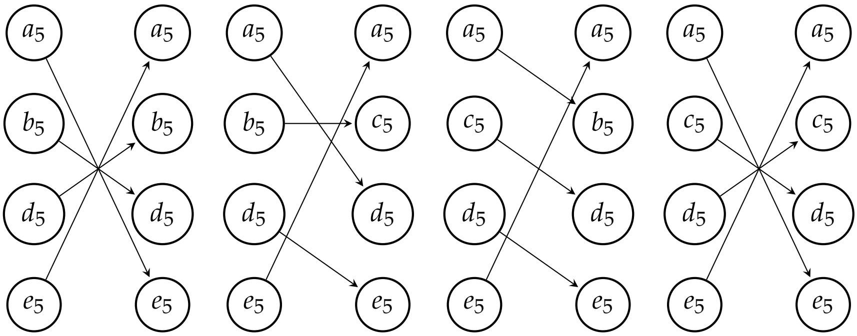
The parameter values $c_1 = 1$ and $c_2 = 0$ forced us to define $R(a_9)$ as the only cycle of length 3, i.e. $\mathcal{C}_1^1 := \{a_5, d_5, e_5\} \in S_5{}^{\oplus 3}$. Recalling Proposition 7.16, we can verify in Figure 7.11a that the edge $(\mathcal{C}_1^1, \mathcal{C}_1^1, 1) \in E_5{}^{\oplus 3}$. Regarding the parameter c_3, its value does not provide any insight on which multi-set we should assign b_9 to. However, since $T = 3$, the cycle of label 2 in $\mathcal{G}_9$ involves 6 nodes. Moreover, the node a_9 belongs to this cycle so $R(a_9)$ is a subset of this set of 6 nodes. Then, we are looking for 3 nodes denoted by $s_1, s_2, s_3 \in S_5$ such that $\mathcal{C}_1^1 \cup \{s_1, s_2, s_3\}$ is the sum (in the sense of Definition 7.7) of three cycles of label 2 and of length 2. Then, $R(b_9)$ is defined as $\{s_1, s_2, s_3\}$. It turn outs that there is a single configuration such that R is a simulation relation, i.e. $(R(a_9), R(b_9), 1), (R(a_9), R(b_9), 2)$ and $(R(b_9), R(a_9), 2) \in E_5{}^{\oplus 3}$. In this case, $R(b_9) := \{a_5, b_5, e_5\}$. Recalling the characterization of the edges of the sum lift in Proposition 7.16, the perfect matchings are illustrated in Figure 7.11 for all the edges of $\mathcal{G}_9$.

We can follow the same logic for $\mathcal{G}_2$ recalled in Figure 7.10b. This graph admits two loops of label 2 and one cycle of label 1 of length 2. Using Proposition 7.28, this implies that the possible values of $T \in \mathbb{N}$ have to



(a) $(a_9, a_9, 1)$. **(b)** $(a_9, b_9, 1)$. **(c)** $(a_9, b_9, 2)$. **(d)** $(b_9, a_9, 2)$.

Fig. 7.11 Illustration of the perfect matchings of the edges $(R(s), R(d), i)$ for the four edges in $\mathcal{G}_9$ with $R(a_9) := \{a_5, d_5, e_5\}$ and $R(b_9) := \{a_5, b_5, e_5\}$.



(a) $(a_2, a_2, 2)$. **(b)** $(a_2, b_2, 1)$. **(c)** $(b_2, a_2, 1)$. **(d)** $(b_2, b_2, 2)$.

Fig. 7.12 Illustration of the perfect matchings of the edges $(R(s), R(d), i)$ for the four edges in $\mathcal{G}_2$ with $R(a_2) := \{a_5, b_5, d_5, e_5\}$ and $R(b_2) := \{a_5, b_5, d_5, e_5\}$.

satisfy the following equations:

$$\begin{cases} 2 \times T &= 3 \times c_1 + 5 \times c_2, \\ 1 \times T &= 2 \times c_3. \end{cases} \tag{7.7}$$

We find that a possible solution is $T = 4$, $c_1 = c_2 = 1$ and $c_3 = 2$. Let us prove that ${\mathcal{G}_5}^{\oplus 4}$ simulates $\mathcal{G}_2$ and build the simulation relation $R : S_2 \to {S_5}^{\oplus 4}$. The first equation in Equation (7.7) implies that

$$R(a_2) \oplus R(b_2) := \mathcal{C}_1^1 \oplus \mathcal{C}_2^1 := \{a_5, a_5, b_5, c_5, d_5, d_5, e_5, e_5\}.$$

Of all the possible configurations, only one satisfies the definition of the simulation relation. In this case, $R(a_2) := \{a_5, b_5, d_5, e_5\}$ and $R(b_2) := \{a_5, b_5, d_5, e_5\}$. For each edge in $\mathcal{G}_2$, the corresponding perfect matching is illustrated in Figure 7.12. △

Although Theorem 7.18 and in particular Theorem 7.25 and Proposition 7.28 highlight the strong connection between the conservatism degree of a graph-based stability certificate and its combinatorial structure, we have not been able to provide an a priori upper-bound on the value of T in Lemma 7.20. However, we will prove that the simulation relation can be checked in polynomial time.

It turns out that the T-sum lift introduced in Definition 7.9 is a generalization of the construction presented in [PAAJ19]. Indeed, the comparison

(6.9) of path-complete graphs in Definition 6.21 is investigated for the particular template of quadratic functions (closed under sum) and the family of linear switched systems. A sufficient condition to the implication (6.10) is provided and consists in checking whether the solution (W) of the second graph can be defined as a conic combination of the solution (V) of the first graph regardless of the switched system. We recall the definition hereafter (see [PAAJ19, Definition IV.2.] for the formal definition).

Definition 7.30. Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet $\langle M \rangle$. We write

$$\mathcal{G} \ \leq_{\sum} \ \widetilde{\mathcal{G}} \tag{7.8}$$

if there is a matrix $C \in \mathbb{R}_{\geq 0}^{|\widetilde{S}| \times |S|}$ satisfying that

$$\forall \tilde{s} \in \widetilde{S} : \ \sum_{s \in S} C_{\tilde{s},s} > 0,$$

such that for any $n \in \mathbb{N}$, any switched system $F := \{f_i : i \in \langle M \rangle\} \subset \mathcal{C}^0(\mathbb{R}^n, \mathbb{R}^n)$ and any admissible Lyapunov function $V_S := \{V_s : s \in S\} \in PCLF(\mathcal{G}, F)$, the candidate Lyapunov function $U_{\widetilde{S}} := \{U_{\tilde{s}} : \tilde{s} \in \widetilde{S}\}$ such that

$$\forall \tilde{s} \in \widetilde{S}, \forall x \in \mathbb{R}^n : U_{\tilde{s}}(x) \ := \sum_{s \in S} C_{\tilde{s},s} V_s(x), \tag{7.9}$$

satisfies $U_{\widetilde{S}} \in PCLF(\widetilde{\mathcal{G}}, F)$. In this case, we say $\mathcal{G} \leq_{\sum} \widetilde{\mathcal{G}}$ through C.

Remark 7.31. Note that the relation between the Lyapunov functions V_S and $U_{\widetilde{S}}$ in Equation (7.9) will be denoted hereafter by abuse of notation by $U_{\widetilde{S}} := C \times V_S$. △

In [PAAJ19, Theorem IV.4.], it is shown that the existence of a matrix C satisfying (7.8) can be checked via a *linear program (LP)* with integer coefficients. Thus, if the problem is feasible, there is at least one solution to this LP with rational elements. Then, an integer-valued solution can be derived by multiplying a rational-valued solution by the least common multiple of the denominators and dividing by the greatest common divisor of the products. We summarize this discussion in the following statement.

Lemma 7.32. *Consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ on the same alphabet. If $\mathcal{G} \leq_{\sum} \widetilde{\mathcal{G}}$, then there exists an integer matrix $C \in \mathbb{N}^{|\widetilde{S}| \times |S|}$ such that $\mathcal{G} \leq_{\sum} \widetilde{\mathcal{G}}$ through C.*

Before stating the main theorem in this section, let us provide a connection between the sum lift, the ordering $\leq_\Sigma$ and the simulation relation.

Lemma 7.33. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet. The following holds:*

(1) If $\mathcal{G}$ simulates $\widetilde{\mathcal{G}}$, then

$$\mathcal{G} \leq_\Sigma \widetilde{\mathcal{G}}$$

through a matrix $C \in \{0, 1\}^{|\widetilde{S}| \times |S|}$.

(2) For any $T \in \mathbb{N}$,

$$\mathcal{G} \leq_\Sigma \mathcal{G}^{\oplus T}$$

through a matrix $C \in \{0, 1, \ldots, T\}^{|S^{\oplus T}| \times |S|}$.

Proof. (1) Consider $R : \widetilde{S} \to S$ a simulation relation. Given a switched system $F := \{f_i : i \in \langle M \rangle\}$ and $V_S := \{V_s : s \in S\}$ an admissible Lyapunov function for $\mathcal{G}$ and F, we know that the indexed set

$$W_{\widetilde{S}} := \{W_s := V_{R(s)} : s \in \widetilde{S}\}$$

is admissible for $\widetilde{\mathcal{G}}$ and F. Therefore, if we define $C \in \{0, 1\}^{|\widetilde{S}| \times |S|}$ such that for all $p \in \widetilde{S}$ and $q \in S$,

$$C_{p,q} = \begin{cases} 1, & \text{if } q = R(s), \\ 0, & \text{otherwise,} \end{cases}$$

we have that CV_S is equal to $W_{\widetilde{S}}$. Then, regardless of the switched system and the solution V_S admissible for $\mathcal{G}$, $C \times V_S$ is admissible for $\widetilde{\mathcal{G}}$.

(2) Consider $s \in S$ and a multi-set $P := \{p_1, \cdots, p_T\} \in S^{\oplus T}$ of cardinality T. We define the matrix $C \in \{0, 1, \ldots, T\}^{|S^{\oplus T}| \times |S|}$ by $C_{P,s} = \mathrm{m}_P(s)$. Let $F := \{f_i : i \in \langle M \rangle\}$ be a switched system and an admissible Lyapunov function V_S for $\mathcal{G}$ and F. If we define $W_{\widetilde{S}} := C \times V_S$,

$$W_P := \sum_{p_i \in P} V_{p_i}.$$

Therefore, $W_{\widetilde{S}}$ is admissible for $\mathcal{G}^{\oplus T}$ by construction of the T-sum lift (see the proof of Proposition 7.11 for more details). $\square$

We now show that the relation $\leq_\Sigma$ is transitive.

Lemma 7.34. *Consider three path-complete graphs $\mathcal{G} = (S, E)$, $\mathcal{G}^\star = (S^\star, E^\star)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet $\langle M \rangle$. The following expression holds:*

$$\left(\mathcal{G} \leq_\Sigma \mathcal{G}^\star \,\wedge\, \mathcal{G}^\star \leq_\Sigma \widetilde{\mathcal{G}}\right) \Rightarrow \left(\mathcal{G} \leq_\Sigma \widetilde{\mathcal{G}}\right).$$

Proof. Consider $n \in \mathbb{N}$, a switched system $F := \{f_i : i \in \langle M \rangle\}$ on M modes and V_S a candidate Lyapunov function. We assume that the graphs $\mathcal{G}$ and $\mathcal{G}^\star$ and the graphs $\mathcal{G}^\star$ and $\widetilde{\mathcal{G}}$ satisfy the inequality (7.8) through the matrices C_1 and C_2 respectively. Then, the following implications hold

$$V_S \in PCLF(\mathcal{G}, F) \Rightarrow C_1 \times V_S \in PCLF(\mathcal{G}^\star, F)$$

$$\Rightarrow C_1 \times C_2 \times V_S \in PCLF(\widetilde{\mathcal{G}}, F).$$

Therefore, $\mathcal{G} \leq_\Sigma \widetilde{\mathcal{G}}$ through the matrix $C := C_1 \times C_2$ by definition. □

We are now able to prove the following characterization theorem between the sum lift simulation and the inequality (7.8) thanks to Theorem 8.6.

Theorem 7.35. *Given two graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet $\langle M \rangle$, the following statements are equivalent.*

(1) $\mathcal{G}^\oplus$ simulates $\widetilde{\mathcal{G}}$.

(2) $\mathcal{G} \leq_\Sigma \widetilde{\mathcal{G}}$.

Proof. (2) $\Rightarrow$ (1) : Suppose that $\mathcal{G} \leq_\Sigma \widetilde{\mathcal{G}}$ and consider any template $\mathcal{V}$ closed under addition. By Lemma 7.32, we can assume without loss of generality that $\mathcal{G} \leq_\Sigma \widetilde{\mathcal{G}}$ through an integer matrix C. By Definition 7.30, for any switched system F and any admissible solution V_S to $\mathcal{G}$ and F, the candidate Lyapunov function $U_{\widetilde{S}} \in \mathcal{V}^{\widetilde{S}}$ defined by equation (7.9) is admissible for $\widetilde{\mathcal{G}}$ and F. Observe that this function is defined as the sum of functions in $\mathcal{V}$ (possibly with repetitions). Thus $\mathcal{G} \leq_\mathcal{V} \widetilde{\mathcal{G}}$ and by Theorem 8.6, this in particular implies that $\mathcal{G}^\oplus$ simulates $\widetilde{\mathcal{G}}$.

(1) $\Rightarrow$ (2) : By Lemma 7.20, statement (1) implies that there exists $T \in \mathbb{N}$ and $R : \widetilde{S} \to S^{\oplus T}$ a simulation relation. By Item (1) of Lemma 7.33, $\mathcal{G}^{\oplus T} \leq_\Sigma \widetilde{\mathcal{G}}$ through a matrix $C_1 \in \{0,1\}^{|\widetilde{S}| \times |S^{\oplus T}|}$. Moreover, by Item (2) of Lemma 7.33, $\mathcal{G} \leq_\Sigma \mathcal{G}^{\oplus T}$ through a matrix $C_2 \in \langle T \rangle^{|S^{\oplus T}| \times |S|}$. By Lemma 7.34, we can conclude (2). □

Thanks to this result, one can use the LP characterization in [PAAJ19, Theorem IV.4.] to check the existence of a simulation relation, and a posteriori identify the appropriate value of $T \in \mathbb{N}$. We illustrate this result in the following example.

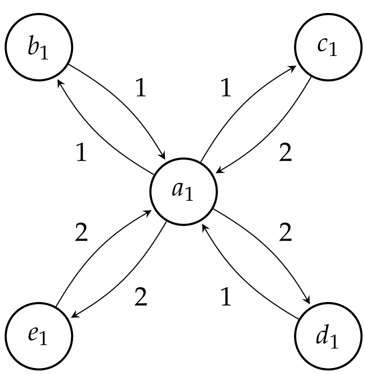
Example 7.36. Consider the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ in Figure 7.1 and repeated hereafter in Figure 7.13. On one hand, we have proven in Example 7.6 that $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_2$ for any template closed under addition. In fact, one can prove that the 2-sum lift of $\mathcal{G}_1$ simulates $\mathcal{G}_2$. From Lemma 7.33, we can derive the matrices $C^1_{sum} \in \{0,1,2\}^{|S_1{}^{\oplus 2}| \times |S_1|}$ and $C^1_{sim} \in \{0,1\}^{|S_2| \times |S_1{}^{\oplus 2}|}$ such that

$$\mathcal{G}_1 \leq_{\Sigma} \mathcal{G}_1{}^{\oplus 2} \text{ and } \mathcal{G}_1{}^{\oplus 2} \leq_{\Sigma} \mathcal{G}_2$$

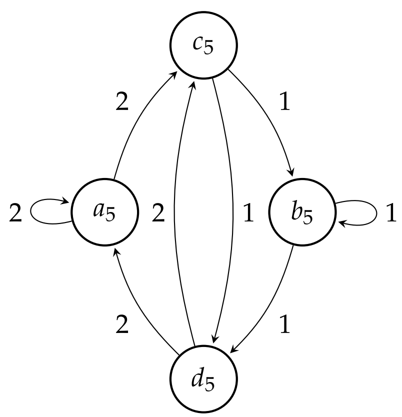
through C^1_{sum} and C^1_{sim} respectively. Therefore, by Lemma 7.34, $\mathcal{G}_1 \leq_{\Sigma} \mathcal{G}_2$ through the matrix

$$C^1 := C^1_{sim} \times C^1_{sum} := \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{7.10}$$

On the other hand, thanks to Theorem 7.35, we can solve the LP criterion to verify the conditions of Definition 7.30. We implemented the LP in MAT-



(a) The path-complete graph $\mathcal{G}_1 = (S_1, E_1)$.



(b) $\mathcal{G}^{3,1}_{db}$, the generalized De Bruijn graph of order 2 and memory 1.

Fig. 7.13 The path-complete graphs $\mathcal{G}_1$ and $\mathcal{G}^{3,1}_{db}$ in Example 7.6. Even though $\mathcal{G}_1 \not\leq \mathcal{G}^{3,1}_{db}$, $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}^{3,1}_{db}$ for any template $\mathcal{V}$ closed under addition. Moreover, $\mathcal{G}_1 \leq_{\Sigma} \mathcal{G}^{3,1}_{db}$ through C^1 in Equation (7.10).

LAB with the toolbox Yalmip [Löf04] and the solver Mosek [MOS19]. Due to the solver, we obtain a floating number solution; to obtain a rational solution, we thus round it and we check a posteriori the feasability, which is indeed still satisfied. Eventually, we get the following matrix

$$C^1_{\text{LP}} := \begin{bmatrix} \frac{1373}{1408} & 0 & \frac{1373}{1408} & 0 & 0 \\ \frac{1373}{1408} & \frac{1373}{1408} & 0 & 0 & 0 \\ \frac{1373}{1408} & 0 & 0 & 0 & \frac{1373}{1408} \\ \frac{1373}{1408} & 0 & 0 & \frac{1373}{1408} & 0 \end{bmatrix}$$

which provides the same simulation relation by multiplying by $\frac{1408}{1373}$. △

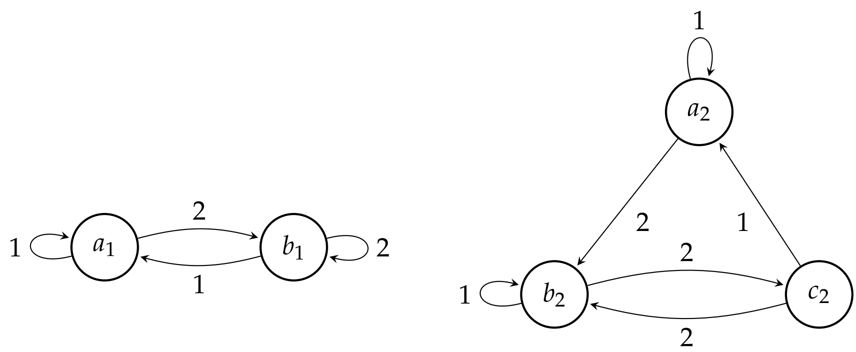
7.4 Min and Max lifts

Even though the binary operation of the sum is natural and many templates of Lyapunov functions are closed under sum, it turns out that our approach generalizes to less straightforward binary operations. For instance, the templates of piecewise $\mathcal{C}^1$ functions [DGTZ21] and polyhedral functions [AJ19], which are commonly used for stability analysis, are closed under pointwise maximum of finitely many functions [DGTZ21, Proposition 3]. These results motivate the introduction of both *min* and *max lifts*.

Let us start with a motivating example.

Example 7.37. Consider the graphs $\mathcal{G}_1 = (S_1, E_1)$ and $\mathcal{G}_2 = (S_2, E_2)$ in Figures 7.14a and 7.14b, and compare them with respect to their conservatism.

First, we prove that $\mathcal{G}_1$ does not simulate $\mathcal{G}_2$. Assume by contradiction that there exists a simulation function $R : S_2 \to S_1$. Because of the self-loops $(a_2, a_2, 1)$ and $(b_2, b_2, 1)$, we must have that $R(a_2) = R(b_2) := a_1$. The node c_2 admits an incoming edge with label 2, then $R(c_2) := b_1$ since the node a_1 only admits incoming edges with label 1. Nevertheless, the edge $(a_2, b_2, 2) \in E_2$ and $(R(a_2), R(b_2), 2) = (a_1, a_1, 2) \notin E_1$, proving that R cannot be a simulation function. A similar approach can be applied to prove that $\mathcal{G}_2$ does not simulate $\mathcal{G}_1$. These two facts, recalling Theorem 6.30, are equivalent to $\mathcal{G}_1 \not\leq \mathcal{G}_2$ and $\mathcal{G}_2 \not\leq \mathcal{G}_1$.



(a) $\mathcal{G}_1 = (S_1, E_1)$, a path-complete graph with 2 nodes and 4 edges.

(b) $\mathcal{G}_2 = (S_2, E_2)$, a path-complete graph with 3 nodes and 6 edges.

Fig. 7.14 Two path-complete graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ over the alphabet $\langle 2 \rangle$ in Example 7.37 such that $\mathcal{G}_1 \not\leq \mathcal{G}_2$ and $\mathcal{G}_2 \not\leq \mathcal{G}_1$ but $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_2$ and $\mathcal{G}_2 \leq_{\mathcal{V}} \mathcal{G}_1$ for any template $\mathcal{V}$ closed under pointwise minimum.

In contrast, we can prove that both graphs are equivalent in the sense of Equation (6.11) for any template closed under pointwise minimum. We consider a template $\mathcal{V}$ which satisfies this closure property, a switched system F with 2 modes and a solution $V_{S_1} := \{V_s : s \in S_1\} \in \mathcal{V}^{S_1}$ admissible for the graph $\mathcal{G}_1$ and the system F. We define $W_{S_2} := \{W_s : s \in S_2\}$ by

$$W_{S_2} := \begin{cases} W_{a_2} & := & V_{a_1}, \\ W_{b_2} & := & \min\{V_{a_1}, V_{b_1}\}, \\ W_{c_2} & := & V_{b_1}. \end{cases} \tag{7.11}$$

We can prove that W_{S_2} is admissible for $\mathcal{G}_2$ and F, meaning that $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_2$ for any template $\mathcal{V}$ closed under pointwise minimum. Let us consider for instance the edge $(a_2, b_2, 2) \in E_2$, and let us prove that the corresponding inequality is satisfied, i.e.

$$\forall x \in \mathbb{R}^n,\ \underbrace{\min\{V_{a_1}(f_2(x)), V_{b_1}(f_2(x))}_{:= W_{b_2}(f_2(x))} \leq \underbrace{V_{a_1}(x)}_{:= W_{a_2}(x)}. \tag{7.12}$$

Since $(a_1, b_1, 2) \in E_1$, the corresponding Lyapunov inequality is satisfied by the functions V_{a_1} and V_{b_1}. As a result, the inequality in Equation (7.12) is automatically satisfied.

Conversely, assume that $W_{S_2} := \{W_s : s \in S_2\} \in \mathcal{V}^{S_2}$ is an admissible solution for $\mathcal{G}_2$ and F. We define $V_{S_1} := \{V_s : s \in S_1\}$ by

$$V_{S_1} := \begin{cases} V_{a_1} & := & \min\{W_{a_2}, W_{b_2}\}, \\ V_{b_1} & := & \min\{W_{b_2}, W_{c_2}\}. \end{cases} \tag{7.13}$$

We can prove that V_{S_1} is admissible for $\mathcal{G}_1$ and F. Let us take for instance the edge $(a_1, b_1, 2) \in E_1$. The corresponding Lyapunov inequality, i.e.

$$\forall x \in \mathbb{R}^n,\ \underbrace{\min\{W_{b_2}(f_2(x)), W_{c_2}(f_2(x))}_{:= V_{b_1}(f_2(x))} \leq \underbrace{\min\{W_{a_2}(x), W_{b_2}(x)\}}_{:= V_{a_1}(x)}, \tag{7.14}$$

is satisfied because the edges $(a_2, b_2, 2)$ and $(b_2, c_2, 2) \in E_2$. This implies that $\mathcal{G}_2 \leq_{\mathcal{V}} \mathcal{G}_1$ for any template $\mathcal{V}$ closed under pointwise minimum. $\triangle$

The rest of this section is organized as follows: we first define both the min and the max lifts and formally demonstrate their validity with respect to the templates closed under pointwise minimum and maximum respectively. Then we prove the duality between the min and max lifts, and show how it can be used to help demonstrate some results. In Section 7.4.3, we discuss the comparison of any path-complete graph with the common Lyapunov function graph $\mathcal{G}_0$ for the two class of templates of interest. Finally, we provide a sufficient condition for the template-dependent ordering of graphs using the simulation relation and the lifts. Moreover, all the results in this section are illustrated with several examples.

7.4.1 Definition and properties

Let us now define the *min* and the *max lifts*, which respectively exploit the pointwise minimum and maximum closure properties.

Definition 7.38 (Max and min lifts)**.** Consider a graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$.

(a) The *max lift*, denoted by $\mathcal{G}_{\max} = (S_{\max}, E_{\max})$, is defined as follows:

(1) The set of nodes $S_{\max}$ is defined by

$$S_{\max} := \{S' \subseteq S \mid S' \neq \emptyset\}.$$

(2) An edge $(A, B, i) \in E_{max}$ with $A, B \in S_{\max}$ and $i \in \langle M \rangle$ if and only if for all $b \in B$, there exists at least one $a \in A$ such that $(a, b, i) \in E$.

(b) The *min lift*, denoted by $\mathcal{G}_{\min} = (S_{\min}, E_{\min})$, is defined as follows:

(1) The set of nodes $S_{\min}$ is defined by

$$S_{\min} := \{S' \subseteq S \mid S' \neq \emptyset\}.$$

(2) An edge $(A, B, i) \in E_{min}$ with $A, B \in S_{\min}$ and $i \in \langle M \rangle$ if and only if for all $a \in A$, there exists at least one $b \in B$ such that $(a, b, i) \in E$.

It is worth noting that, contrary to the addition, both the minimum and maximum operations are idempotent (i.e. for any $a \in \mathbb{R}, \min\{a, a\} = a$). This explains why we restrict the nodes to the (non empty) subsets and we do not consider multi-sets here. Therefore, both lifted graphs admit a finite number of nodes and a finite number of edges.

We need to prove that these lifts are well-defined. As a first step, let us show that both operations preserve the path-completeness.

Proposition 7.39. *The min and max lifts preserve the path-completeness of graphs.*

Proof. Given a path-complete graph $\mathcal{G} = (S, E)$, let us prove that $\mathcal{G}_{\min}$ admits a component isomorphic to the initial graph $\mathcal{G}$, which directly proves the path-completeness of the min lift $\mathcal{G}_{\min}$.

By Definition 7.38, the min lift $\mathcal{G}_{\min}$ admits a node for every singleton $\{s\}$ where $s \in S$. Moreover, for every edge $(s, p, i) \in E$ of the initial graph, the edge $(\{s\}, \{p\}, i) \in E_{\min}$ by definition of the min lift. Therefore, the restriction of the min lift to the singleton nodes is isomorphic to the initial graph and then path-complete.

A similar argument can be used to prove the path-completeness of the max lift. □

Let us now discuss the validity of those lifts.

Proposition 7.40. *Consider the family of binary operations $\{\wedge_n\}_{n\in\mathbb{N}}$ such that $\forall n \in \mathbb{N}$, $\wedge_n$ corresponds to the pointwise maximum. The max lift is valid in the sense of Definition 7.3 with respect to any template closed under $\{\wedge_n\}_{n\in\mathbb{N}}$.*

Proof. Consider a template $\mathcal{V}$ closed under pointwise maximum and any family of vector fields $\{f_i : i \in \langle M \rangle\}$. Suppose that there exists a PCLF for the initial graph $\mathcal{G}$ of the form $\{V_s : s \in S\} \subset \mathcal{V}$. Given any $A \in S_{\max}$ the corresponding Lyapunov function $W_A \in \mathcal{V}$ is defined by

$$\forall x \in \mathbb{R}^n : \; W_A(x) \; := \; \max_{a \in A} \{V_a(x)\}. \tag{7.15}$$

Given $(A, B, i) \in E_{\max}$, we have

$$W_B(f_i(x)) \; = \; \max_{b \in B} \{V_b(f_i(x))\} \; \leq \; \max_{a \in A} \{V_a(x)\} \; = \; W_A(x),$$

for any $x \in \mathbb{R}^n$, since, by Definition 7.38, for all $b \in B$ there exists at least a $a \in A$ such that $V_b(f_i(x)) \leq V_a(x)$, concluding the proof. □

Proposition 7.41. *Respectively, consider the family of binary operations* $\{\vee_n\}_{n \in \mathbb{N}}$ *such that* $\forall n \in \mathbb{N}$, $\vee_n$ *corresponds to the pointwise minimum. The min lift is valid in the sense of Definition 7.3 with respect to any template closed under* $\{\vee_n\}_{n \in \mathbb{N}}$.

Proof. The proof is the same as the proof for max, but given any $A \in S_{\min}$ the corresponding Lyapunov function $W_A \in \mathcal{V}$ is defined by $W_A(x) := \min_{a \in A} \{V_a(x)\}$, $\forall x \in \mathbb{R}^n$. □

Example 7.42. Consider the graph $\mathcal{G}_1 = (S_1, E_1)$ in Figure 7.14a, path-complete on $\langle M \rangle := \{1, 2\}$. If we apply the min-lift procedure as introduced in Definition 7.38, we obtain $(\mathcal{G}_1)_{min}$ represented in Figure 7.15. We see that $(\mathcal{G}_1)_{min}$ has one strongly-connected and path-complete component isomorphic to $\mathcal{G}_1$ itself (the subgraph induced by $\{a\}$ and $\{b\}$) as demonstrated in the proof of Proposition 7.39.

Moreover, we can see that the graph $\mathcal{G}_2$ in Figure 7.14b is a sub-component of $(\mathcal{G}_1)_{\min}$ (in bold). Since $\mathcal{G}_1 \leq_{\mathcal{V}} (\mathcal{G}_1)_{\min}$ for any template closed under pointwise minimum by Proposition 7.40 and by Proposition 6.23, we can derive the same results as in Example 7.37, that is

$$\mathcal{G}_1 \; \leq_{\mathcal{V}} \; \mathcal{G}_2$$

for any template $\mathcal{V}$ closed under pointwise minimum.

Finally, there is another sub-component induced by $\{a, b\}$ (with dashed

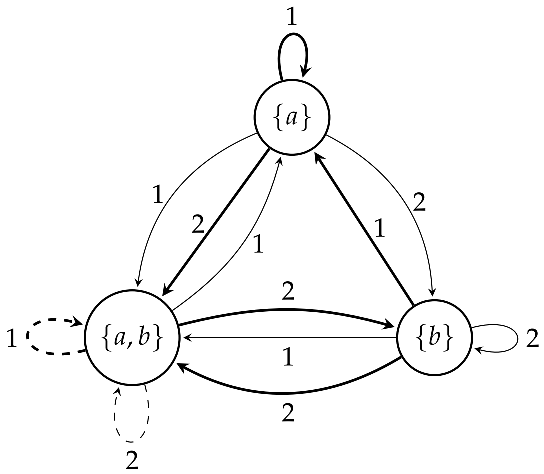


Fig. 7.15 $(\mathcal{G}_1)_{\min}$, the min lift of the path-complete graph $\mathcal{G}_1$ in Figure 7.14a. The subgraph with bold edges is the graph $\mathcal{G}_2$ in Figure 7.14b, which proves that $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_2$ for any template closed under pointwise minimum. The subgraph with dashed edges is the common Lyapunov function graph $\mathcal{G}_0$, which proves that $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_0$ for this kind of templates.

edges) which is isomorphic to $\mathcal{G}_0$. Therefore, we can derive that

$$\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_0,$$

for any template $\mathcal{V}$ closed by minimum, or, in other words, the graph $\mathcal{G}_1$ is as conservative as $\mathcal{G}_0$ (the "worst" graph) for this kind of templates. △

We can now study for which templates defined in Section 2.2.3 the min and the max lift are valid. Using Corollary 2.36, we can prove that the min lift is valid with respect to the template of primal copositive norms provided that they are not properly closed under pointwise minimum. By duality, we can derive the validity of the max lift with respect to the dual copositive norms.

Theorem 7.43. *Consider $\mathcal{G} = (S, E)$ a path-complete graph on $\langle M \rangle$, any $n \in \mathbb{N}$ and any $T \in \mathbb{N}$. Denote by $\mathcal{P}$ and $\mathcal{D}$ the template of primal and dual norms on $\mathbb{R}^n_{\geq 0}$ respectively, and by $\mathcal{L}$ the set of all the positive matrices in $\mathbb{R}^{n \times n}_{\geq 0}$. We have*

(a) $\mathcal{G} \leq_{\mathcal{P},\mathcal{L}} \mathcal{G}_{min}$,

(b) $\mathcal{G} \leq_{\mathcal{D},\mathcal{L}} \mathcal{G}_{max}$.

Proof. The first relation follows from Proposition 7.40 and Corollary 2.36. Note that the template of primal copositive norms is not closed under minimum, but the construction proposed in Proposition 7.40 is still possible by Corollary 2.36, which requires the linearity of the subvector fields. We can thus conclude $\mathcal{G} \leq_{\mathcal{P},\mathcal{L}} \mathcal{G}_{\min}$, but not, in general, $\mathcal{G} \leq_{\mathcal{P}} \mathcal{G}_{\min}$.

By duality, it is possible to develop similar arguments for the template of dual copositive norms $\mathcal{D}$. □

By abuse of language, we will say that the template of primal copostive norms is closed under pointwise minimum to refer to the inequality (a) in Theorem 7.43.

Example 7.44. Consider the path-complete graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ in Figures 7.14a and 7.14b respectively. We have already demonstrated in Example 7.37 that $\mathcal{G}_1 \not\leq \mathcal{G}_2$ and $\mathcal{G}_2 \not\leq \mathcal{G}_1$ but $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_2$ and $\mathcal{G}_2 \leq_{\mathcal{V}} \mathcal{G}_1$ for any template $\mathcal{V}$ closed under pointwise minimum.

Consider the positive linear switched system $\mathcal{A} := \{A_1, A_2\}$ with

$$A_1 := \begin{bmatrix} 0.9 & 0.3 \\ 0.9 & 0.7 \end{bmatrix}, \; A_2 := \begin{bmatrix} 0.6 & 0.9 \\ 0.6 & 0.3 \end{bmatrix}, \tag{7.16}$$

and assume that we use the template of primal copositive norms $\mathcal{P}$. Since this template is closed under pointwise minimum, we expect that the approximations of the joint spectral radius provided by $\mathcal{G}_1$ and $\mathcal{G}_2$, denoted by $\rho_{\mathcal{G}_1,\mathcal{P}}(\mathcal{A})$ and $\rho_{\mathcal{G}_2,\mathcal{P}}(\mathcal{A})$ respectively, are the same. Indeed, using the YALMIP toolbox [Löf04], we get

$$\rho_{\mathcal{G}_1,\mathcal{P}}(\mathcal{A}) = \rho_{\mathcal{G}_2,\mathcal{P}}(\mathcal{A}) = 1.549.$$

If instead we consider the template of quadratic functions $\mathcal{Q}$, which is not closed under pointwise minimum, the results provided by the graphs may be different. Indeed, for the system (7.16), we obtain

$$\rho_{\mathcal{G}_1,\mathcal{Q}}(\mathcal{A}) = 1.356 < 1.364 = \rho_{\mathcal{G}_2,\mathcal{Q}}(\mathcal{A}). \tag{7.17}$$

This inequality proves that graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are not equivalent with respect to the family of quadratic Lyapunov functions. △

7.4.2 Duality

Recalling Corollary 2.36 and the duality theory in Section 1.2, we expect the min and max lifts to be each other's dual, as stated in the following lemma.

Lemma 7.45. *The min and max lifts are dual, i.e. for any path-complete graph $\mathcal{G} = (S, E)$ on $\langle M \rangle$, it holds that*

$$\left[(\mathcal{G}^\top)_{\max}\right]^\top = \mathcal{G}_{\min}, \tag{7.18a}$$

$$\left[(\mathcal{G}^\top)_{\min}\right]^\top = \mathcal{G}_{\max}. \tag{7.18b}$$

Proof. We use the notation $\mathcal{G}^\top = (S, E^\top)$, $\mathcal{G}_{\min} = (S_{\min}, E_{\min})$, $(\mathcal{G}^\top)_{\max} = (S^\top_{\max}, E^\top_{\max})$ and $[(\mathcal{G}^\top)_{\max}]^\top = ((S^\top_{\max})^\top, (E^\top_{\max})^\top)$. First of all, by Definition 6.3 and 7.38 we have $S_{\min} = (S^\top_{\max})^\top (= \mathcal{P}(S) \setminus \{\emptyset\})$. Then, considering $S_1, S_2 \in S_{\min}$ and $i \in \langle M \rangle$, again by Definition 6.3 and 7.38 we have

$$\begin{aligned}
(S_1, S_2, i) \in E_{\min} \quad &\Leftrightarrow \quad \forall\, a \in S_1,\, \exists\, b \in S_2 \;:\; (a, b, i) \in E, \\
&\Leftrightarrow \quad \forall\, a \in S_1,\, \exists\, b \in S_2 \;:\; (b, a, i) \in E^\top, \\
&\Leftrightarrow \quad (S_2, S_1, i) \in E^\top_{\max}, \\
&\Leftrightarrow \quad (S_1, S_2, i) \in (E^\top_{\max})^\top,
\end{aligned}$$

concluding the proof of (7.18a). Equation (7.18b) trivially follows with similar arguments. □

Using the duality in Lemma 7.45, it is easier to deduce the previous result. In particular, the path-completeness of the max lift can be directly obtained.

Alternative proof of Proposition 7.39 for the max lift using duality. We have already proven that if a graph $\mathcal{G}$ is path-complete, then the min lift $\mathcal{G}_{\min}$ is path-complete. We can prove using duality that the max lift preserves the path-completeness.

Consider a path-complete graph $\mathcal{G}$. By Proposition 6.6, $\mathcal{G}^\top$ is path-complete. Then, by Proposition 7.39, $(\mathcal{G}^\top)_{\min}$ is path-complete. By combining Proposition 6.6 and Lemma 7.45, we end the proof. □

7.4.3 Comparison with the common Lyapunov function graph

The construction of the min lift reminds us of the notion of *co-observer graph* [CL10, Section 2.3.4] which is in fact a sub-component of the min-lift. This construction has already been used in [Phi17, Definition 5.29] and is repeated below.

Definition 7.46 (Co-observer graph)**.** Given a graph $G = (S, E)$ with M labels, the *co-observer graph* denoted by $cO(G) := (S_C, E_C)$ is a graph where each state correponds to a subset of S, and is constructed as follows:

(1) Set $S_C := \{S\}$ and $E_C := \emptyset$;

(2) Set $X = \emptyset$. Then, for each pair $(Q, \sigma) \in S_C \times \langle M \rangle$:

  a. Compute $P = \underset{q \in Q}{\cup} \{p \mid (p, q, \sigma) \in E\}$;

  b. If $P \neq \emptyset$, set $E_C := E_C \cup \{(P, Q, \sigma)\}$ then $X := X \cup \{P\}$.

(3) If $X \subseteq S_C$, then the co-observer graph is given by $cO(G) = (S_C, E_C)$. Else, set $S_C := S_C \cup X$ and go to step (2).

One can show that the co-observer graph is strongly connected and co-complete (therefore path-complete), see [Phi17] for a more detailed analysis. Let us take an example, and compare this construction with the min lift.

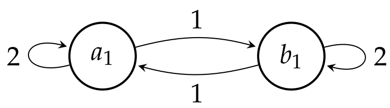
Example 7.47*.* Take the initial graph $\mathcal{G}_1 = (S_1, E_1)$ in Figure 7.16a . We compute its co-observer graph $cO(\mathcal{G}_1)$ illustrated in Figure 7.16b. We observe that $cO(\mathcal{G}_1)$ corresponds to the common Lyapunov function graph $\mathcal{G}_0$. The min lift $\mathcal{G}_{1\min}$ of the initial graph is illustrated in Figure 7.16c. We can notice that the co-observer graph is a subcomponent of the min lift. By Propositions 6.24 and 7.40, we have that

$$\mathcal{G}_0 \leq \mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_0,$$

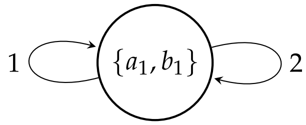
where $\mathcal{V}$ denotes any template closed under pointwise mininimum. The graph $\mathcal{G}_1$ and the common Lyapunov function graph $\mathcal{G}_0$ are then equivalent from a conservatism-based point of view for this kind of templates. $\triangle$

By Definition 7.38 and 7.46, it is clear that the co-observer graph of any path-complete graph is a subset of the min lift.

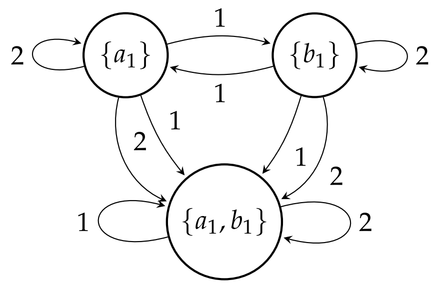
Lemma 7.48. *Given a path-complete graph $\mathcal{G}$, the co-observer graph $cO(\mathcal{G})$ is a subcomponent of the min lift $\mathcal{G}_{\min}$.*



(a) $\mathcal{G}_1 = (S_1, E_1)$, path-complete graph on $\langle 2 \rangle$.



(b) $cO(\mathcal{G}_1) = (S_{1C}, E_{1C})$, the co-observer graph of $\mathcal{G}_1$.



(c) $\mathcal{G}_{1\min} = (S_{1\min}, E_{1\min})$, the min lift of $\mathcal{G}_1$. The co-observer graph $cO(\mathcal{G}_1)$ is one of the subcomponents of $\mathcal{G}_{1\min}$.

Fig. 7.16 Comparison between the co-observer graph and the min lift in Example 7.47.

Proof. Consider an edge $(P, Q, \sigma) \in E_C$. By construction, $p \in P$ if and only if there exists $q \in Q$ such that $(p, q, \sigma) \in E$, which is exactly the condition imposed in Definition 7.38 to add an edge to the min lift. $\square$

Similarly, the definition of the max-lift reminds us of the notion of *observer graph* [CL10, Section 2.3.4] used in [PAAJ19, Definition III.5]. The definition is repeated below.

Definition 7.49 (Observer graph)**.** Given a graph $G = (S, E)$ with M labels, the *observer graph* $O(G) = (S_O, E_O)$ is a graph where each state corresponds to a subset of S, and is constructed as follows:

(1) Set $S_O := \{S\}$ and $E_O := \emptyset$;

(2) Set $X = \emptyset$. For each pair $(P, \sigma) \in S_O \times \langle M \rangle$:

  a. Compute $Q = \bigcup_{p \in P} \{q \mid (p, q, \sigma) \in E\}$;

  b. If $Q \neq \emptyset$, set $E_O := E_O \cup \{(P, Q, \sigma)\}$ then $X := X \cup \{Q\}$.

(3) If $X \subseteq S_O$, then the observer graph is given by $O(G) = (S_O, E_O)$. Else, set $S_O := S_O \cup X$ and go to step 2.

It is well known that the observer graph is strongly connected and complete by construction, see [Phi17]. Moreover, one can easily observe a duality between the observer graph and the co-observer graph (see [Phi17, Lemma 5.32], i.e.

$$O(\mathcal{G}) = \left(cO(\mathcal{G}^\top)\right)^\top.$$

Lemma 7.50. *Given a path-complete graph $\mathcal{G}$, the observer graph $O(\mathcal{G})$ is a subcomponent of the max lift $\mathcal{G}_{\max}$.*

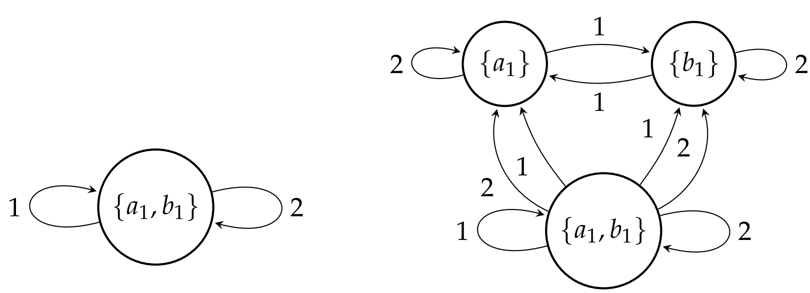
Proof. Each edge (P, Q, σ) of the observer graph satisfies that $q \in Q$ if and only if there exists $p \in P$ such that $(p, q, \sigma) \in E$, which is exactly the condition imposed in Definition 7.38 to add a new edge in the max lift. □

Example 7.51. Let us consider the same graph as for the min lift, i.e. the graph $\mathcal{G}_1 = (S_1, E_1)$ in Figure 7.16a. Once again, if we compute the observer graph $O(\mathcal{G}_1) = (S_{1O}, E_{1O})$, we find the common Lyapunov function graph as illustrated in Figure 7.17a. If we compute the max-lift $\mathcal{G}_{1\max} = (S_{1\max}, E_{1\max})$ of the graph $\mathcal{G}_1$, we find the graph in Figure 7.17b. We can observe that the observer graph is one of the subcomponents of the max lift. This means that, in this case, we have

$$\mathcal{G}_0 \;\leq\; \mathcal{G}_1 \;\leq_{\mathcal{V}}\; \mathcal{G}_0,$$

where $\mathcal{V}$ is a template closed under the maximum. The graph $\mathcal{G}_1$ and its observer graph $O(\mathcal{G}_1)$ are then equivalent for this sort of templates. △

Proposition 6.24 states that the common Lyapunov function graph $\mathcal{G}_0$ is worse than any path-complete graph in the sense of Definition 6.21. Nevertheless one can easily imagine that for some graphs and under some assumptions, the reverse inequality holds. For example, in both Examples 7.47 and 7.51, we have shown that the graph $\mathcal{G}_1$ in Figure 7.16a is as



(a) $O(\mathcal{G}_1) = (S_{1O}, E_{1O})$, the observer graph of $\mathcal{G}_1$.

(b) $\mathcal{G}_{1\max} = (S_{1\max}, E_{1\max})$, the max lift of $\mathcal{G}_1$. The observer graph $O(\mathcal{G}_1)$ is one of the subcomponents of $\mathcal{G}_{1\max}$.

Fig. 7.17 Comparison between the observer graph and the max lift in Example 7.51.

conservative as $\mathcal{G}_0$ for any template closed under pointwise minimum or maximum. This result actually derives from a graph property of $\mathcal{G}_1$.

Proposition 7.52. *Consider a path-complete graph $\mathcal{G}$ on the alphabet $\langle M \rangle$ and the common Lyapunov function graph $\mathcal{G}_0$.*

(a) If $\mathcal{G}$ is complete*, $\mathcal{G}_0$ is a path-complete component of $\mathcal{G}_{\min}$ and thus*

$$\mathcal{G} \;\leq_{\mathcal{V}}\; \mathcal{G}_0$$

for any template $\mathcal{V}$ closed under pointwise minimum.

(b) If $\mathcal{G}$ is co-complete*, $\mathcal{G}_0$ is a path-complete component of $\mathcal{G}_{\max}$ and thus*

$$\mathcal{G} \;\leq_{\mathcal{V}}\; \mathcal{G}_0$$

for any template $\mathcal{V}$ closed under pointwise maximum.

(c) The following inequalities

$$\mathcal{G} \;\leq_{\mathcal{V}}\; (\mathcal{G}_{\max})_{\min} \;\leq_{\mathcal{V}}\; \mathcal{G}_0 \tag{7.19}$$

and

$$\mathcal{G} \;\leq_{\mathcal{V}}\; (\mathcal{G}_{\min})_{\max} \;\leq_{\mathcal{V}}\; \mathcal{G}_0 \tag{7.20}$$

hold for any template $\mathcal{V}$ closed under pointwise minimum and maximum.

Proof. (a) Consider a complete graph $\mathcal{G}$ on $\langle M \rangle$. By assumption, for all $a \in S$ and all $i \in \langle M \rangle$, there exists $b \in S$ such that $(a, b, i) \in E$. By Definition of the min lift, this means that the edge $(S, S, i) \in E_{\min}$ for every label $i \in \langle M \rangle$. Therefore, the common Lyapunov function graph $\mathcal{G}_0$ is a subcomponent of $\mathcal{G}_{\min}$. Propositions 6.23 and 7.40 end the proof.

(b) Similarly, we can prove that the edge $(S, S, i) \in E_{\max}$ for every label $i \in \langle M \rangle$ when the graph $\mathcal{G}$ is co-complete.

(c) We prove the ordering in (7.19). The ordering in (7.20) can be proved using similar arguments. Consider a path-complete graph $\mathcal{G}$, and let us build its max lift $\mathcal{G}_{\max}$. By Lemma 7.50, the observer graph $O(\mathcal{G})$ is a subcomponent of $\mathcal{G}_{\max}$. However $O(\mathcal{G})$ is complete by construction. Using Proposition 7.52(a) ends the proof. □

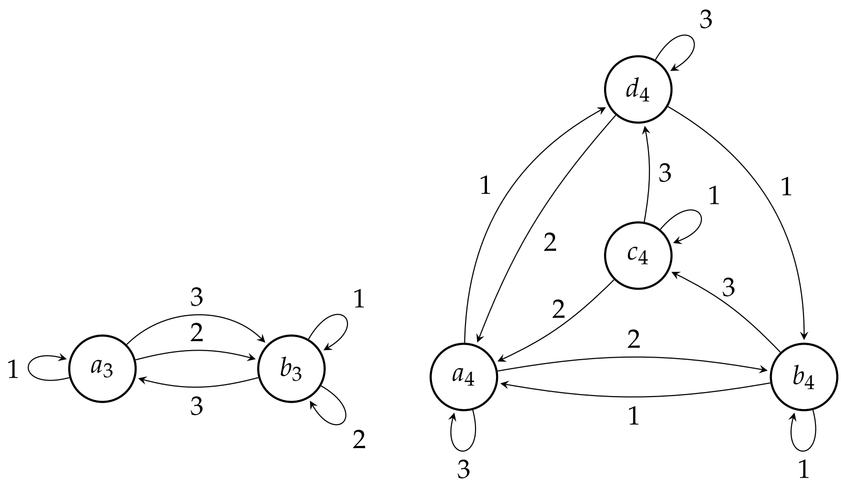
This result is similar to [PAAJ19, Theorem III.8.], which states that it is always possible to derive a common Lyapunov function from a multiple Lyapunov function defined as the composition of the minimum and the maximum. However, Proposition 7.52 transcribes this result in the template-dependent formalism and only involves graph properties and graph orderings. In particular, Proposition 7.52 states that using a template closed under pointwise minimum (resp. maximum) with a complete (resp. co-complete) graph is not useful, since this PCLF certificate is as conservative as the common Lyapunov function graph.

Corollary 7.53. *Given $M \in \mathbb{N}$, all the complete (resp. co-complete) graphs on the alphabet $\langle M \rangle$ are as conservative as the common Lyapunov function graph $\mathcal{G}_0$ for any template $\mathcal{V}$ closed under pointwise minimum (resp. maximum), i.e.*

$$\mathcal{G}_0 \leq \mathcal{G} \quad \textit{and} \quad \mathcal{G} \leq_{\mathcal{V}} \mathcal{G}_0.$$

Let us compare complete graphs for different templates with different closure properties.

Example 7.54. Let us consider the path-complete graphs $\mathcal{G}_{db}^2$, $\mathcal{G}_3$ and $\mathcal{G}_4$ on the alphabet $\langle 3 \rangle$, illustrated in Figures 6.4a and 7.18a. One can easily observe that the three graphs are complete. By Proposition 7.52, we know that



(a) $\mathcal{G}_3 = (S_3, E_3)$, a path-complete graph with 2 nodes and 6 edges.

(b) $\mathcal{G}_4 = (S_4, E_4)$, a path-complete graph with 4 nodes and 12 edges.

Fig. 7.18 Example of two complete graphs $\mathcal{G}_3$ and $\mathcal{G}_4$ on $\langle 3 \rangle$ in Example 7.54. By Proposition 7.52, $\rho_{\mathcal{G}_3,\mathcal{P}}(\mathcal{A}) = \rho_{\mathcal{G}_4,\mathcal{P}}(\mathcal{A})$ for any finite set of 3 nonnegative matrices $\mathcal{A} := \{A_1, A_2, A_3\} \subset \mathbb{R}^n_{\geq 0}$ of any dimension $n \in \mathbb{N}$, while their approximations differ with the quadratic template.

all these graphs are equivalent with respect to their conservatism for any template closed under pointwise minimum and any family of nonnegative linear switched systems. By Theorem 7.43, we know that the template of primal copositive norms is closed under pointwise minimum. Using Proposition 6.22 and Proposition 7.52, all the graphs should provide the same JSR approximation for any finite set of nonnegative matrices.

In order to verify this result, we simulate 500 positive linear switched systems with 3 modes of dimension 3. For each of them, we compute the JSR approximation provided by each graph. As expected, the three graphs provide exactly the same JSR approximations for all the systems when we look for primal copositive norms. For comparison, we also compute the quadratic JSR approximation provided by each graph. Since this template is not closed under pointwise minimum, we expect the graphs to provide different JSR approximations. The results are illustrated in the Venn dia-

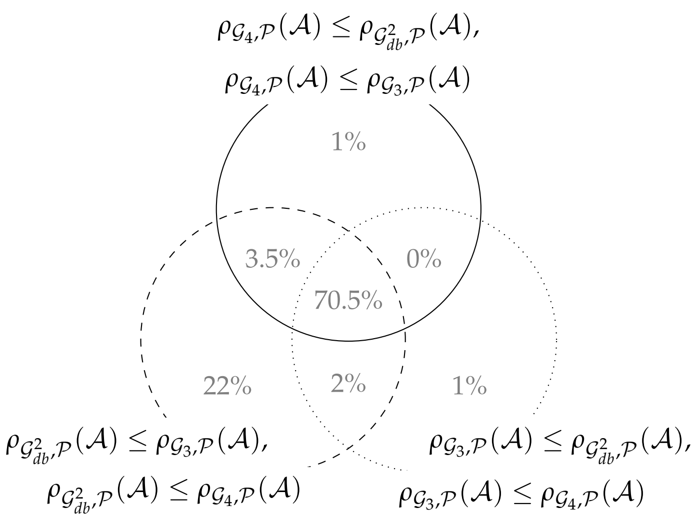


Fig. 7.19 Visualisation of the outcome of the numerical experiment in Example 7.54. As expected from the theory, the three graphs $\mathcal{G}^2_{db}$, $\mathcal{G}_3$ and $\mathcal{G}_4$ always provide the same JSR approximation when using the template of primal copositive norms. In constrat, the graphs sometimes provide different JSR approximations with the quadratic template. In most cases (70.5%), the three graphs provide the same result. However, in almost all other cases (22%), $\mathcal{G}^2_{db}$ is strictly better than $\mathcal{G}_3$ and $\mathcal{G}_4$. It appears uncommon that $\mathcal{G}_3$ or $\mathcal{G}_4$ admits a solution, and not $\mathcal{G}^2_{db}$.

gram in Figure 7.19. We observe that, for most systems (70.5%), $\mathcal{G}^2_{db}$, $\mathcal{G}_3$ and $\mathcal{G}_4$ provide the same JSR approximation. If not, the best JSR approximation is nearly always provided by the De Bruijn graph $\mathcal{G}^2_{db}$. This means that there is only a small proportion of systems (2%) for which $\mathcal{G}^2_{db}$ does not provide the best approximation, which makes it the best graph in terms of conservatism even if it does not satisfy the ordering relations in Definition 6.21.

Note that in Figure 7.19, we assume that two graphs provide the same JSR approximation if the absolute value of their difference is smaller or equal to 10^{-6}; we attribute these small differences to numerical errors. According to our numerical experiments, it seems that the few percentages of systems for which $\mathcal{G}_4$ or $\mathcal{G}_3$ provides the best JSR approximation still remain, even though we increase the bisection precision. △

7.4.4 Simulation-based sufficient condition for the template-dependent ordering of graphs

As we did for the sum lift and the family of template closed under addition, this section aims to provide a combinatorial sufficient condition for the ordering of graphs in Definition 6.21 with respect to any template closed under pointwise minimum and maximum.

Theorem 7.55. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet. If $\mathcal{G}_{\min}$ simulates $\widetilde{\mathcal{G}}$, then*

$$\mathcal{G} \ \leq_{\mathcal{V}} \ \widetilde{\mathcal{G}}$$

for any template $\mathcal{V}$ closed under pointwise minimum.

Proof. Consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ on the same alphabet. By Proposition 7.40, the inequality

$$\mathcal{G} \ \leq_{\mathcal{V}} \ \mathcal{G}_{\min}$$

is satisfied for any template $\mathcal{V}$ closed under pointwise minimum. By assumption and recalling the simulation-based characterisation Theorem 6.30 in [PJ19, Theorem 3.5],

$$\mathcal{G}_{\min} \ \leq \ \widetilde{\mathcal{G}}.$$

Then, by transitivity of the ordering (see Equation (6.17) in page 131 for more details) , we have

$$\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$$

for this class of templates. □

Using Proposition 7.39, the characterization of the general ordering in Theorem 6.30 and the transitivity, we manage to prove the same result for the max lift.

Theorem 7.56. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet. If $\mathcal{G}_{\max}$ simulates $\widetilde{\mathcal{G}}$, then*

$$\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$$

for any template $\mathcal{V}$ closed under pointwise maximum.

Note that for the linear case, using Corollary 2.36 and the duality for lift in Proposition 7.5, we obtain the same result.

Note that the necessary conditions for both Theorems 7.55 and 7.56 will be proved in Chapter 8.

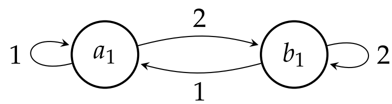
7.5 Composition lifts

In this section and in order to study the order relation (6.9) in Definition 6.21, we introduce another lift whose validity depends on both the template and the dynamics properties. This lift sheds new light, and provides a generalization of previous results in the literature, such as [AJPR14, Proposition 4.2], [PAAJ19, Example IV.11] and [PJ19, Example 3.9]. For instance, the *composition lift* introduced in Definition 7.58 below was implicitly used in the particular case of quadratic Lyapunov functions and linear switched systems, but it was not clear how it could be used in a general setting. Proposition 7.60 will answer the question.

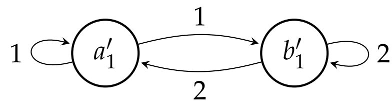
As an incentive for the composition lifts , we recall and transcribe [PJ19, Example 3.9] in the template-dependent ordering framework.

Example 7.57. Consider the path-complete graph $\mathcal{G}_1 = (S_1, E_1)$ and its dual graph $\mathcal{G}_1^\top = (S_1, E_1^\top)$ in Figure 7.20.

One can first verify that there is no simulation relation between $\mathcal{G}_1$ and $\mathcal{G}_1^\top$. Let us show for instance that $\mathcal{G}_1$ does not simulate $\mathcal{G}_1^\top$. Because of



(a) $\mathcal{G}_1 = (S_1, E_1)$, a path-complete graph with 2 nodes and 4 edges.



(b) $\mathcal{G}_1^\top = (S_1, E_1^\top)$, the dual graph of $\mathcal{G}_1$.

Fig. 7.20 Two path-complete graphs, namely $\mathcal{G}_1$ and its dual $\mathcal{G}_1$ over the alphabet $\langle 2 \rangle$ in Example 7.57 such that $\mathcal{G}_1 \not\leq \mathcal{G}_2$ and $\mathcal{G}_2 \not\leq \mathcal{G}_1$ but $\mathcal{G}_1 \leq_{\mathcal{V}} \mathcal{G}_2$ and $\mathcal{G}_2 \leq_{\mathcal{V}} \mathcal{G}_1$ for any template $\mathcal{V}$ closed under composition with invertible matrices.

the loops $(a_1', a_1', 1)$ and $(b_1', b_1', 2) \in E_1^\top$, a simulation relation $R : S_1 \to S_1$ could only be defined as $R(a_1') = a_1$ and $R(b_1') = b1$. However, $(a_1', b_1', 1) \in E_1^\top$ but $(R(a_1'), R(b_1'), 1) := (a_1, b_1, 1) \notin E_1$. This implies that $\mathcal{G}_1$ does not simulate $\mathcal{G}_1^\top$. A similar argument can be used to prove that $\mathcal{G}_1^\top$ does not simulate $\mathcal{G}_1$ either.

However it is still possible to compare both graphs using template-dependent ordering of graphs. In particular, we will show that $\mathcal{G}_1 \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}_2$ for any template $\mathcal{V}$ closed under composition with dynamics in $\mathcal{F}$, and $\mathcal{G}_1^\top \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}_1$ for any template $\mathcal{V}$ closed under composition with the inverse dynamics of $\mathcal{F}$ (provided that the dynamics are invertible). Given a switched system $F := \{f_1, f_2\}$ with 2 modes of any dimension $n \in \mathbb{N}$, let us assume that there exists an admissible solution $\{V_{a_1}, V_{b_1}\}$ for $\mathcal{G}_1$ and F. We define $\{W_{a_1'}, W_{b_1'}\}$ by

$$\begin{cases} W_{a_1'} &:= & V_{a_1} \circ f_1, \\ W_{b_1'} &:= & V_{b_1} \circ f_2. \end{cases} \tag{7.21}$$

We can prove that $\{W_{a_1'}, W_{b_1'}\}$ is admissible for $\mathcal{G}_1^\top$ and F. Consider for instance the edge $(a_1', b_1', 1) \in E_1^\top$ that encodes the following Lyapunov inequality:

$$\forall x \in \mathbb{R}^n, \ \underbrace{V_{b_1} \circ f_2\,(f_1(x))}_{:= W_{b_1'}(f_1(x))} \ \leq \ \underbrace{V_{a_1}\,(f_1(x))}_{:= W_{a_1'}(x)}.$$

This inequality is satisfied because $\{V_{a_1}, V_{b_1}\} \in PCLF(\mathcal{G}_1, F)$ and $(a_1, b_1, 2) \in E_1$. Then, the inequality

$$V_{b_1}\,(f_2(x)) \ \leq \ V_{a_1}(x)$$

is satisfied for all $x \in \mathbb{R}^n$, and in particular for the state of the form $f_1(x)$ with $x \in \mathbb{R}^n$. The three other edges of $\mathcal{G}_1^\top$ can be treated similarly.

Reversely, consider an *invertible* switched system $F := \{f_1, f_2\}$ and $\{W_{a_1'}, W_{b_1'}\} \in PCLF(\mathcal{G}_1^\top, F)$. We can show that $\{V_{a_1}, V_{b_1}\}$ defined as

$$\left\{ \begin{array}{lcl} V_{a_1} & := & W_{a_1'} \circ f_1^{-1} \\ V_{b_1} & := & W_{b_1'} \circ f_2^{-1} \end{array} \right. \tag{7.22}$$

is admissible for $\mathcal{G}_1$ and F. Let us take the edge $(b_1, a_1, 1) \in E_1$ which encodes the following Lyapunov inequality:

$$\forall x \in \mathbb{R}^n,\ \underbrace{W_{a_1'} \circ f^{-1}\left(f_1(x)\right)}_{:=\ V_{a_1}(f_1(x))} \leq \underbrace{W_{b_1'}\left(f_2^{-1}(x)\right)}_{:= V_{b_1}(x)}.$$

Since $(b_1', a_1', 2) \in E_1^\top$, this inequality is automatically satisfied.

In conclusion, both $\mathcal{G}_1 \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}_1^\top$ and $\mathcal{G}_1^\top \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}_1$ for any template $\mathcal{V}$ closed under the composition with the dynamics in $\mathcal{F}$ and their inverse. In particular, these graphs are equivalent with respect to the conservatism if we consider the usual template $\mathcal{Q}$ of quadratics and the linear switched systems with invertible matrices. Therefore, for any such system $\mathcal{A}$, $\rho_{\mathcal{G}_1,\mathcal{Q}}(\mathcal{A}) = \rho_{\mathcal{G}_1^\top,\mathcal{Q}}(\mathcal{A})$. △

This section is structured as follows: we first define both the forward and backward composition lifts that respectively exploit the composition operation with the dynamics or their inverse. We discuss their validity and duality, and we finally provide simulation-based condition to guarantee the ordering of graphs for the relevant classes of templates and families of switched systems. Using these theorems, we show how these new concepts bring new perspectives and generalize results previously obtained in the literature.

7.5.1 Definition and properties

Let us first define the *forward composition lift* which can be seen as a *future-based* multiple Lyapunov function.

Definition 7.58 (Forward composition lifts)**.** Consider a graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$.

(a) The *T-forward composition lift*, denoted by $\mathcal{G}^{\circ T} = (S^{\circ T}, E^{\circ T})$ is defined as follows:

(1) The set of nodes $S^{\circ T}$ is defined by

$$S^{\circ T} := \{(s, i_1, \ldots, i_T) \mid s \in S, i_k \in \langle M \rangle, k = 1, \ldots, T\}$$

(2) For every edge $(a, b, i) \in E$ and each T-tuple of label $(j_1, \ldots, j_T) \in \langle M \rangle^T$, the edge $((a, j_1, \ldots, j_T), (b, i, j_1, \ldots, j_{T-1}), j_T) \in E^{\circ T}$.

(b) The *forward composition lift*, denoted by $\mathcal{G}^{\circ} = (S^{\circ}, E^{\circ})$, is defined as the infinite disjoint union of the T-forward composition lift, i.e.

$$\mathcal{G}^{\circ} := \bigcup_{T \in \mathbb{N}} \mathcal{G}^{\circ T}.$$

Let us first show that, for any integer value $T \in \mathbb{N}$, the T-forward composition lift of any path-complete graph remains path-complete.

Proposition 7.59. *For every $T \in \mathbb{N}$, the T-forward composition lift preserves the path-completeness of graphs.*

Proof. Consider a path-complete graph $\mathcal{G} = (S, E)$ on $\langle M \rangle$, its T-forward composition lift $\mathcal{G}^{\circ T} = (S^{\circ T}, E^{\circ T})$ for any $T \in \mathbb{N}$ and a word $\sigma := \sigma(1) \ldots \sigma(k)$ of length $k \in \mathbb{N}$ on the alphabet $\langle M \rangle$. By path-completeness of $\mathcal{G}$, there exist k edges

$$e_i := (a_i, a_{i+1}, \sigma(i)) \in E$$

for $i = 1, \ldots, k$. We have to build a path in $E^{\circ T}$ whose word is σ.

Let us first observe that if a path in $\mathcal{G}^{\circ T}$ starts in a node $(a, i_1, \ldots, i_T) \in S^{\circ T}$, the label of the first edge is i_T by Definition 7.58. Then, the label of the second edge is i_{T-1}, and so forth.

Consider any $j_1, \ldots, j_T \in \langle M \rangle$. By Definition of the forward composition lift, the edge $((a_1, j_1, \ldots, j_T), (a_2, \sigma(1), j_1, \ldots, j_{T-1}), j_T) \in E^{\circ T}$ since $(a_1, a_2, \sigma(1)) \in E$. Similarly, the edge

$$((a_2, \sigma(1), j_1, \ldots, j_{T-1}), (a_3, \sigma(2), \sigma(1), j_1, \ldots, j_{T-2}), j_{T-1}) \in E^{\circ T}$$

since $(a_2, a_3, \sigma(2)) \in E$. By following this procedure, we can end up with two different scenarios. If $k \leq T$, we can build a path in $\mathcal{G}^{\circ T}$ such that we reach the node

$$((a_k, \sigma(k), \sigma(k-1), \ldots, \sigma(1), j_1, \ldots, j_{T-k}), (a_{k+1}, \sigma(k+1), \ldots, \sigma(1), j_1, \ldots, j_{T-k-1}), j_{T-k}) \in E^{\circ T}.$$

By Assumption 7.1, it is possible to recursively extend the path in $\mathcal{G}$ (whatever the label and the nodes) and therefore extend the path in $\mathcal{G}^{\circ T}$ such that we progressively reach the edges of label $\sigma(i)$, which proves the path-completeness of $\mathcal{G}^{\circ T}$.

Otherwise, if $k > T$, we can reach the node

$$((a_T, \sigma(T), \sigma(T-1), \ldots, \sigma(1)), (a_{T+1}, \sigma(T+1), \ldots, \sigma(2)), \sigma(1)) \in E^{\circ T},$$

which will be the starting node in $\mathcal{G}^{\circ T}$ of the path whose word is σ. By Definition 7.58, we can progressively extend the path in $\mathcal{G}^{\circ T}$ from the path in $\mathcal{G}$ and then reach the node

$$((a_k, \sigma(k), \sigma(k-1), \ldots, \sigma(k-T)), (a_{k+1}, h, \sigma(k), \ldots, \sigma(k-T+1)), \sigma(k-T)) \in E^{\circ T},$$

where $h \in \langle M \rangle$. Then, by Assumption 7.1 and following the procedure described in the first scenario ($k \leq T$), we can progressively extend the path in $\mathcal{G}^{\circ T}$ such that we reach the edges of label $\sigma(i)$ for $i = k-T+1, \ldots, k$. $\square$

We now discuss the validity of the forward composition lift. By extension of Definition 1.33, we say that a template $\mathcal{V} = \cup_{n \in \mathbb{N}} \mathcal{V}_n$ is *closed under composition with the dynamics of* $\mathcal{F}$ if for any $n \in \mathbb{N}$, for all $V \in \mathcal{V}_n$ and $f \in \mathcal{F} \cap \mathcal{C}^0(\mathbb{R}^n, \mathbb{R}^n)$, the composition $V \circ f \in \mathcal{V}_n$.

Proposition 7.60. *The (T-)forward composition lift is valid in the sense of Definition 7.3 with respect to any family $\mathcal{F}$ of systems and any template $\mathcal{V}$ closed under composition with the dynamics of $\mathcal{F}$.*

Proof. Consider a family of systems $\mathcal{F}$, a system $F = \{f_i : i \in \langle M \rangle\} \in \mathcal{F}^{\langle M \rangle}$ and a template $\mathcal{V}$ closed under composition with any dynamics in $\mathcal{F}$. Suppose that there exists a PCLF for an initial path-complete graph $\mathcal{G} = (S, E)$ of the form $\{V_s : s \in S\} \in \mathcal{V}^S$. Given $s \in S$ and $i_1, \ldots, i_T \in \langle M \rangle$, the corresponding Lyapunov function $W_{(s,i_1,\ldots,i_T)}$ is defined by

$$\forall x \in \mathbb{R}^n : \; W_{(s,i_1,\ldots,i_T)}(x) \; := \; \left(V_s \circ f_{i_1} \circ \ldots \circ f_{i_T}\right)(x). \tag{7.23}$$

Given $((a, i_1, \ldots, i_T), (b, j, i_1, \ldots, i_{T-1}), i_T) \in E^{\circ T}$, we have

$$\begin{aligned} W_{(b,j,i_1,\ldots,i_{T-1})}(f_{i_T}(x)) \;&:=\; V_b\left(f_j \circ f_{i_1} \circ \cdots \circ f_{i_{T-1}}\left(f_{i_T}(x)\right)\right) \\ &\leq\; V_a\left(f_{i_1} \circ \cdots \circ f_{i_T}(x)\right) \\ &:=\; W_{(a,i_1,\ldots,i_T)}(x), \end{aligned}$$

for any $x \in \mathbb{R}^n$, since, by Definition 7.61, $(a, b, j) \in E$ and the Lyapunov inequality encoded by this edge is satisfied by the Lyapunov functions $\{V_s : s \in S\}$, especially when they are evaluated at points of the form $y = f_{i_1} \circ \cdots \circ f_{i_T}(x)$. □

If the dynamics are invertible, a similar lift to the forward composition lift in Definition 7.58, referred as the *backward composition lift*, can be defined. In this case, the Lyapunov functions associated to the nodes of the lifted graph are defined as the composition with the inverse dynamics. Similarly to the forward case, the backward composition lift can be seen as a *memory-based* multiple Lyapunov function.

Definition 7.61 (Backward composition lifts)**.** Consider a graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$.

(a) The *T-backward composition lift*, denoted by $\mathcal{G}^{-\circ T} = (S^{-\circ T}, E^{-\circ T})$ is defined as follows:

   (1) The set of nodes $S^{-\circ T}$ is defined by

$$S^{-\circ T} := \{(s, i_1, \ldots, i_T) \mid s \in S, i_k \in \langle M \rangle, k = 1, \ldots, T\}$$

   (2) For every edge $(a, b, i) \in E$ and each T-tuple of label $(j_1, \ldots, j_T) \in \langle M \rangle^T$, the edge $((a, i, j_1, \ldots, j_{T-1}), (b, j_1, \ldots, j_T), j_T) \in E^{-\circ T}$.

(b) The *backward composition lift*, denoted by $\mathcal{G}^{-\circ} = (S^{-\circ}, E^{-\circ})$, is defined as the infinite disjoint union of the T-backward composition lift, i.e.

$$\mathcal{G}^{-\circ} := \bigcup_{T \in \mathbb{N}} \mathcal{G}^{-\circ T}.$$

As we did for all the previous lifts, let us first show that the lifted graph remains path-complete.

Proposition 7.62. *For every $T \in \mathbb{N}$, the T-backward composition lift preserves the path-completeness.*

Proof. The proof follows the same ideas as the proof of Proposition 7.62 but the path in $\mathcal{G}^{-\circ T}$ must be built from the end. □

Let us finally discuss the validity of the backward composition lift.

Proposition 7.63. *The (T-)backward composition lift is valid in the sense of Definition 7.3 with respect to any family $\mathcal{F}$ of invertible systems and any template $\mathcal{V}$ closed under composition with the inverse dynamics of $\mathcal{F}$.*

Proof. Consider a family of systems $\mathcal{F}$, an invertible system $F \in \mathcal{F}^{\langle M \rangle}$ and a template $\mathcal{V}$ closed under composition with any inverse dynamics of $\mathcal{F}$. Suppose that there exists a PCLF for an initial path-complete graph $\mathcal{G} = (S, E)$ of the form $\{V_s : s \in S\} \in \mathcal{V}^S$. Given $s \in S$ and $i_1, \dots, i_T \in \langle M \rangle$, the corresponding Lyapunov function $W_{(s,i_1,\dots,i_T)}$ is defined by

$$\forall x \in \mathbb{R}^n : \; W_{(s,i_1,\dots,i_T)}(x) \; := \; \left(V_s \circ f_{i_1}^{-1} \circ \dots \circ f_{i_T}^{-1} \right)(x). \tag{7.24}$$

Given $((a, j, i_1, \dots, i_{T-1}), (b, i_1, \dots, i_T), i_T) \in E^{-\circ T}$, we have

$$\begin{aligned} W_{(b,i_1,\dots,i_T)}(f_{i_T}(x)) \quad &:= \quad V_b \left(f_{i_1}^{-1} \circ \dots \circ f_{i_T}^{-1} \left(f_{i_T}(x) \right) \right) \\ &\leq \quad V_a \left(f_j^{-1} \circ f_{i_1}^{-1} \circ \dots \circ f_{i_{T-1}}^{-1}(x) \right) \\ &:= \quad W_{(a,j,i_1,\dots,i_{T-1})}(x), \end{aligned}$$

for any $x \in \mathbb{R}^n$, since, by Definition 7.61, $(a, b, j) \in E$ and the Lyapunov inequality encoded by this edge is satisfied by the Lyapunov functions $\{V_s : s \in S\}$, especially when they are evaluated at points of the form $y = f_j^{-1} \circ \dots \circ f_{i_{T-1}}^{-1}(x)$. $\square$

Unlike the template-dependent lifts introduced in Sections 7.3 and 7.4, the validity of the compositions lifts defined in Definition 7.58 and 7.61 depend on the properties of both the systems and the templates. In particular, it requires a strong interplay between template and dynamics since the closure properties depend on the dynamics.

7.5.2 Duality

In the context of linear switched systems and a template of norms, Lemma 1.27 states that the dual operation of the composition with a matrix remains the composition but with the inverse and transposed matrix. According to Proposition 7.5, this is a good indicator that the forward and backward composition lifts could be dual.

Proposition 7.64. *The forward and backward composition lifts are dual, i.e. for any $T \in \mathbb{N}$ and any path-complete graph $\mathcal{G}$, it holds that*

$$\mathcal{G}^{-\circ T} \; = \; \left((\mathcal{G}^\top)^{\circ T} \right)^\top .$$

Proof. Consider $M, T \in \mathbb{N}$ and a path-complete graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$. Given an edge $(s, d, i) \in E$ and $j_1, \dots, j_T \in \langle M \rangle$, the edge

$$((d, j_1, \dots, j_T), (s, i, j_1, \dots, j_{T-1}), j_T) \in (E^\top)^{\circ T}$$

by Definition 7.58. Then, the dual edge belongs to $E^{-\circ T}$ by Definition 7.61, which concludes the proof. □

The duality in Proposition 7.64 makes it easier to prove certain results. The demonstration of the path-completeness of the T-backward composition lift is a tangible example.

Proof of Proposition 7.62. Consider a path-complete graph $\mathcal{G}$, and its T-backward composition lift $\mathcal{G}^{-\circ T}$. By Proposition 7.64, $\mathcal{G}^{-\circ T}$ can be defined as the dual of the T-forward composition lift of the dual graph $\mathcal{G}$. Since the transposition and the forward lift both preserve the path-completeness (see Propositions 6.6 and 7.59), the lifted graph $\mathcal{G}^{-\circ T}$ is path-complete. □

Similarly, the validity of the backward composition lift for the linear switched systems can be directly derived using duality.

Alternative proof for Proposition 7.63. First, let us notice that Lemma 1.27 states that the dual operation of the composition is the composition with the inverse and transpose dynamics. Then, using Propositions 7.5 and 7.60, we directly derive the validity of the backward composition lift. □

7.5.3 Simulation-based sufficient condition for the template-dependent ordering of graphs

Let us now derive combinatorial sufficient conditions for the ordering of graphs (6.9) in Definition 6.21.

Theorem 7.65. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet. If $\mathcal{G}^\circ$ simulates $\widetilde{\mathcal{G}}$, then*

$$\mathcal{G} \ \leq_{\mathcal{V},\mathcal{F}} \ \widetilde{\mathcal{G}}$$

for any template $\mathcal{V}$ closed under composition with the dynamics in $\mathcal{F}$.

Proof. Consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ on the same alphabet. By Proposition 7.60, the inequality

$$\mathcal{G} \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}^{\circ}$$

is satisfied for any template $\mathcal{V}$ closed under compositions with the dynamics in $\mathcal{F}$. By assumption and recalling the simulation-based characterisation Theorem 6.30 in [PJ19, Theorem 3.5],

$$\mathcal{G}^{\circ} \leq \widetilde{\mathcal{G}}.$$

Then, by transitivity of the ordering (see Equation (6.17) in page 131 for more details) , we have

$$\mathcal{G} \leq_{\mathcal{V},\mathcal{F}} \widetilde{\mathcal{G}}$$

for this class of templates and systems. □

We can finally derive the same simulation-based result for the backward composition lift using the same arguments.

Theorem 7.66. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet. If $\mathcal{G}^{-\circ}$ simulates $\widetilde{\mathcal{G}}$, then*

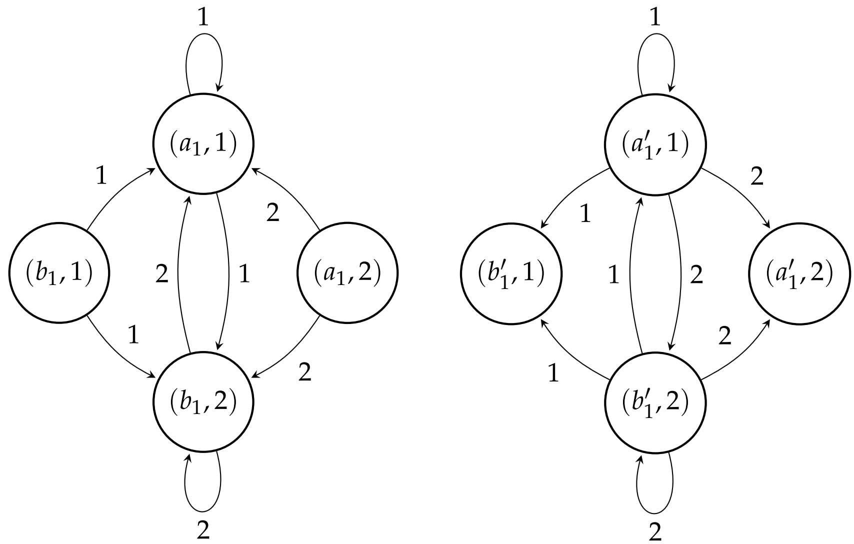
$$\mathcal{G} \leq_{\mathcal{V},\mathcal{F}} \widetilde{\mathcal{G}}$$

for any template $\mathcal{V}$ closed under composition with the inverse dynamics in $\mathcal{F}$, provided that they are invertible.

Example 7.67. Consider the path-complete graph $\mathcal{G}_1$ and its dual $\mathcal{G}_1^{\top}$ in Figure 7.20. We have already shown in Figure 7.20 that $\mathcal{G}_1 \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}_1^{\top}$ for any template $\mathcal{V}$ closed under the composition with the dynamics $\mathcal{F}$. It turns out that the 1-forward composition lift of $\mathcal{G}_1$, illustrated in Figure 7.21a, simulates $\mathcal{G}_1^{\top}$.

Similarly, the comparison $\mathcal{G}_1^{\top} \leq \mathcal{G}_1$ for any template $\mathcal{V}$ closed under composition with the inverse dynamics of $\mathcal{F}$ can be derived from Theorem 7.66. Indeed, there exists a simulation relation between $\left(\mathcal{G}_1^{\top}\right)^{-\circ 1}$, illustrated in Figure 7.21b and $\mathcal{G}_1$. △

In the rest of this section, we will show how these simulation-based Theorems 7.65 and 7.66 can be used to prove already established results but in a combinatorial way and in a more general setting. As first example, we can demonstrate, using our formalism, [Jun24, Corollary 12], which states



(a) $(\mathcal{G}_1)^{\circ 1}$, the 1-forward composition lift of $\mathcal{G}_1$.

(b) $\left(\mathcal{G}_1^{\top}\right)^{-\circ 1}$, the 1-backward composition lift of $\mathcal{G}_1^{\top}$.

Fig. 7.21 Example of the 1-forward and 1-backward composition lift of $\mathcal{G}_1$ in Figure 7.20 and its dual $\mathcal{G}_1^{\top}$. We can prove that $(\mathcal{G}_1)^{\circ 1}$ simulates $\mathcal{G}_1^{\top}$ and $\left(\mathcal{G}_1^{\top}\right)^{-\circ 1}$ simulates $\mathcal{G}_1$. Then, $\mathcal{G}_1$ and $\mathcal{G}_1^{\top}$ are equivalent for any template closed under composition with the inverse of a family of dynamics.

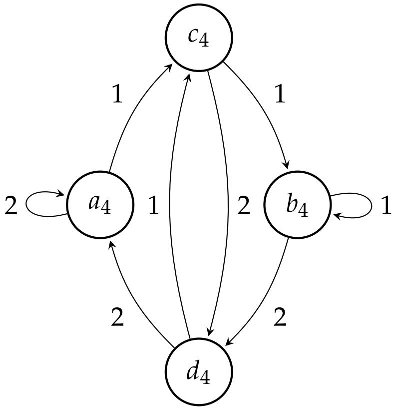
that all the generalized De Bruijn graphs of fixed order provide the same JSR approximation if the template is closed under composition with the dynamics. This result aims to explain the observations that were made in [DJ23, Section 5.2], where the authors use the template of *diagonal* quadratic functions which is not closed under composition.

Example 7.68. We consider the three generalized De Bruijn graphs of order 2 on the alphabet $\langle 2 \rangle$ denoted by $\mathcal{G}_{db}^{3,0}$, $\mathcal{G}_{db}^{3,1}$ and $\mathcal{G}_{db}^{3,2}$ of memory 0, 1 and 2 respectively and recalled in Figure 7.22. We can prove that

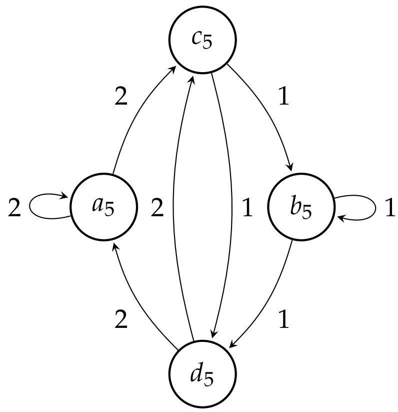
$$\mathcal{G}_{db}^{3,2} \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}_{db}^{3,1} \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}_{db}^{3,0}$$

for any template $\mathcal{V}$ closed under composition with the dynamics $\mathcal{F}$. Moreover, if the dynamics in $\mathcal{F}$ are invertible, we can prove that

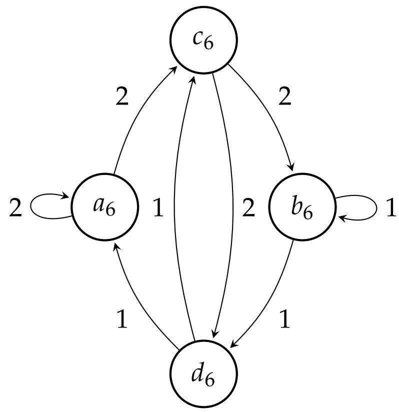
$$\mathcal{G}_{db}^{3,0} \leq_{\mathcal{V},\mathcal{F}} \mathcal{G}_{db}^{3,2}.$$



(a) The generalized De Bruijn graph $\mathcal{G}_{db}^{3,2}$ of order 2 and memory 2.



(b) $\mathcal{G}_{db}^{3,1}$, the generalized De Bruijn graph of order 2 and memory 1.



(c) $\mathcal{G}_{db}^{3,0}$, the generalized De Bruijn graph of order 2 and memory 0.

Fig. 7.22 Examples of *generalized* De Bruijn graphs of order 2 on the alphabet $\langle 2 \rangle$ in Example 7.68.

First, let us show that the 1-forward composition lift of $\mathcal{G}_{db}^{3,2}$ simulates $\mathcal{G}_{db}^{3,1}$. We can prove that the relation $R : S_{db}^{3,1} \to \left(S_{db}^{3,2}\right)^{\circ 1}$ defined by $R(a_5) := (a_4, 2)$, $R(b_5) := (b_4, 1)$, $R(c_5) := (c_4, 1)$ and $R(d_5) := (d_4, 2)$ is a simulation relation. Let us take for instance the edge $(c_5, d_5, 1) \in E_{db}^{3,1}$. The corresponding edge $((c_4, 1), (d_4, 2), 1) \in \left(E_{db}^{3,2}\right)^{\circ 1}$ because the edge $(c_4, d_4, 2) \in E_{db}^{3,2}$, recalling Definition 7.58. Using Theorem 7.65 we conclude.

Similarly, we can prove that the 1-forward composition lift of $\mathcal{G}_{db}^{3,1}$ simulates $\mathcal{G}_{db}^{3,0}$ using the relation $R : S_{db}^{3,0} \to \left(S_{db}^{3,1}\right)^{\circ 1}$ defined by $R(a_6) := (a_5, 2)$, $R(b_6) := (b_5, 1)$, $R(c_6) := (c_5, 1)$ and $R(d_6) = (d_5, 1)$. Consider for instance the edge $(b_6, d_6, 1) \in E_{db}^{3,0}$. Using R, the corresponding edge is $((b_5, 1), (d_5, 1), 1)$ and belongs to $\left(E_{db}^{3,1}\right)^{\circ 1}$ because the edge $(b_5, d_5, 1) \in E_{db}^{3,1}$. Theorem 7.65 implies the desired template and family-dependent ordering between the generalized De Bruijn graphs.

Finally, let us prove that the 2-backward lift of $\mathcal{G}_{db}^{3,0}$ simulates $\mathcal{G}_{db}^{3,2}$. To this aim, we define the relation $R : S_{db}^{3,2} \to \left(S_{db}^{3,0}\right)^{-\circ 2}$ such that $R(a_4) := (a_6, 2, 2)$, $R(b_4) := (b_6, 1, 1)$, $R(c_4) := (c_6, 2, 1)$ and $R(d_4) := (d_6, 1, 2)$. This relation can be proved to be a simulation relation, which ends the argumentation by Theorem 7.66. Let us take for instance the edge $(d_4, c_4, 1) \in E_{db}^{3,2}$ and consider the corresponding edge $((d_6, 1, 2), (c_6, 2, 1), 1)$. $(d_6, c_6, 1) \in E_{db}^{3,0}$. This edge belongs to the 2-backward composition lift of $\mathcal{G}_{db}^{3,2}$ by Definition 7.61, using $a := c_6$, $b := c_6$ and $j_1 j_2 := 21$. △

Using similar simulation-based arguments, we can prove that all the generalized De Bruijn graphs of fixed order are equivalent from a conservatism point of view if we consider a family of invertible switched systems and a template closed under composition with the dynamics and their inverse.

Along the same lines, we can support the De Bruijn hierarchy introduced in [AJPR14, Theorem 6.1] for the quadratic template and further investigated for the template of copositive norms in Section 7.7, using Theorem 7.65.

Proposition 7.69. *For any integer $K \in \mathbb{N}$,*

$$\mathcal{G}_{db}^{K} \ \leq_{\mathcal{V},\mathcal{F}} \ \mathcal{G}_{db}^{K+1} \tag{7.25}$$

for any family $\mathcal{F}$ of invertible switched systems and any template $\mathcal{V}$ closed under the composition with the inverse dynamics in $\mathcal{F}$.

Proof. Using Theorem 7.66, let us prove that the 1-backward composition lift of $\mathcal{G}_{db}^{K}$ simulates $\mathcal{G}_{db}^{K+1}$ for any $K \in \mathbb{N}$. Given $s := (i_1, \dots, i_K) \in S_{db}^{K+1}$, we define

$$R(s) \ := \ ((i_1, \dots, i_{K-1}), i_K) \in \left(S_{db}^{K}\right)^{-\circ 1}.$$

Take any edge $(a, b, j) \in E_{db}^{K+1}$ where the nodes $a := (a_1, \dots, a_K)$ and $b := (a_2, \dots, a_K, j) \in S_{db}^{K+1}$. We have to prove that the edge $(R(a), R(b), j) :=$

$(((a_1, \ldots, a_{K-1}), a_K), ((a_2, \ldots, a_K), j), j) \in \left(E^K_{db}\right)^{-\circ 1}$. By Definition 7.61, this holds if $((a_1, \ldots, a_{K-1}), (a_2, \ldots, a_K), a_K) \in E^K_{db}$ which is satisfied by Definition 6.10. Therefore, R is a simulation relation, which ends the proof. □

In particular, the ordering in (7.25) is satisfied for the quadratic template and the family of invertible linear switched systems.

Let us finally discuss the conservatism of Assumption 6.8, where we assume that we restrict our analysis to path-complete graphs with labels of length 1. We make this assumption because the expanded graph introduced in Definition 6.7 allows to derive a path-complete graph with labels of length 1 from any path-complete graph with labels of multiple lengths. In this following proposition, we understand how conservative this assumption was.

Proposition 7.70. *Consider a path-complete graph $\mathcal{G} = (S, E)$ and its expanded form $\mathcal{G}^e = (S^e, E^e)$. Then,*

$$\mathcal{G}^e \ \leq \ \mathcal{G} \ \text{and} \ \mathcal{G} \ \leq_{\mathcal{V},\mathcal{F}} \ \mathcal{G}^e,$$

for any template $\mathcal{V}$ closed under composition with the dynamics $\mathcal{F}$.

Proof. Consider a path-complete graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$ with at least one label of length strictly larger than 1.

First, let us prove that $\mathcal{G}^e \leq \mathcal{G}$. Consider a switched system F and assume that there exists $V \in PCLF(\mathcal{G}^e, F)$. We can prove that the restriction to the initial nodes $W := V_{|S} = \{V_s : s \in S \cap S^e\}$ is admissible for $\mathcal{G}$ and F. Let us consider an edge $(s, d, w) \in E$ where $w := i_1 \ldots i_k$. By Definition 6.7, we denote by $s_1, \ldots, s_k$ the intermediate nodes added in the expanded graph for this edge. The Lyapunov inequality associated to this edge is satisfied because for all $x \in \mathbb{R}^n$:

$$\begin{aligned}
V_d\left(f_{i_k} \circ \cdots \circ f_{i_1}(x)\right) \ &\leq \ V_{s_k}\left(f_{i_{k-1}} \circ \cdots \circ f_{i_1}(x)\right), \\
&\leq \ V_{s_{k-1}}\left(f_{i_{k-2}} \circ \cdots \circ f_{i_1}(x)\right), \\
&\ \vdots \\
&\leq \ V_{s_1} \circ f_{i_1}(x), \\
&\leq \ V_s(x),
\end{aligned}$$

since $V \in PCLF(\mathcal{G}^e, F)$.

For the second inequality, assume that there exists $V \in PCLF(\mathcal{G}, F)$. We need to prove that we can derive $W := \{W_s : s \in S^e\}$ admissible for $\mathcal{G}^e$ and F. To this aim, we need to define

$$V_{s_q} := V_d \circ f_{i_k} \circ \cdots \circ f_{i_{q+1}}$$

for each node $s_q \in S^e \setminus S$ added from the expansion of the edge (s, d, w) with a label $w := i_1 i_2 \ldots i_k$ of the initial graph. For the common nodes in $S \cap S^e$, we keep the same functions. Then, given an edge $(s_q, s_{q+1}, i_{q+1}) \in E^e \setminus E$, the corresponding Lyapunov inequality is trivially satisfied, i.e. we have that $\forall x \in \mathbb{R}^n$:

$$\begin{aligned} V_{s_{q+1}}\left(f_{i_{q+1}}(x)\right) &:= V_d \circ f_{i_k} \circ \cdots \circ f_{i_{q+2}} \circ f_{i_{q+1}}(x), \\ &= V_{s_q}(x), \end{aligned}$$

which ends the proof. □

7.6 Comparison of template-dependent lifts

In the previous sections, we have introduced different lifts that leverage different operations. For each of them, we have formally demonstrated that the lifted graph is better than the initial graph, in the sense of Definition 6.21, while sometimes needing to restrict their validity to specific classes of templates and switched systems. In this section, we compare the lifts with each other and we ask the following question: is one construction more powerful than another? We answer this question by characterizing the edges of the sum, min and max lifts in terms of relations between subsets.

Let us formally define a relation over two sets.

Definition 7.71 (Binary relation over two sets)**.** Given two sets X and Y, a *relation over X and Y* is a subset of $X \times Y$, i.e.

$$R \subseteq \{(x, y) \mid x \in X, y \in Y\}.$$

If $(x, y) \in R$, we say that *x is in relation with y*, which is also denoted by xRy.

We define some important properties of binary relations.

Definition 7.72 (Properties of relations)**.** Consider a relation $R \subseteq X \times Y$ over the sets X and Y. We say that:

(a) the relation R is *total* if for all $x \in X$, there exists (at least one) $y \in Y$ such that xRy. In other words, the domain of R is X;

(b) the relation R is an *application* if for all $x \in X$, there exists one and only one $y \in Y$ such that xRy;

(c) the relation R is a *function* if for all $x \in X$, there exists at most one $y \in Y$ such that xRy;

(d) the relation R is *surjective* if $\forall y \in Y$, there exists (at least one) $x \in X$ such that xRy. In other words, the codomain of R is Y;

(e) the relation R is *injective* if $\forall x, y \in X$ and all $z \in Y$, if xRz and yRz then $x = y$;

(f) the relation R is *bijective* if it is both surjective and injective;

(g) a relation $S \subseteq X \times Y$ is *contained* in R, denoted by $S \subseteq R$ if S is a subset of R, i.e. $\forall x \in X, y \in Y$, if xSy then xRy;

(h) a relation R is *finite* if there exists a finite number of pairs (x, y) in R. The relation is *infinite* if there exist infinitely many pairs (x, y) in R.

Remark 7.73. Note that a bijective relation is necessarily a function. Moreover, if R is a function over two finite sets of equal cardinality, the surjectivity implies the injectivity while there is equivalence if R is an application (i.e. a total function). △

Given a path-complete graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, the conditions required to add an edge in the min and max lifts in Definition 7.38 can be expressed using binary relations.

Definition 7.74 (Relation induced by an edge)**.** Consider a graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, two non-empty subsets $A, B \subseteq S$ and a label $i \in \langle M \rangle$. We define the *relation* R_e over A and B *induced by the edge* $e = (A, B, i)$ as follows:

$$\forall a \in A,\ \forall b \in B :\ aR_e b \ \Leftrightarrow\ (a, b, i) \in E.$$

The following proposition characterizes the edges of the min and max lift by adding some properties on the induced relation R_e.

Proposition 7.75. *Consider a path-complete graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$. Given two non-empty subsets $A, B \subseteq S$ and $i \in \langle M \rangle$, the following statements hold.*

(a) The edges of the min lift satisfy that

$$e := (A, B, i) \in E_{\min} \Leftrightarrow \exists R \subseteq R_e \text{ s.t. } R \text{ is an application.}$$

(b) The edges of the max lift satisfy that

$$e := (A, B, i) \in E_{\max} \Leftrightarrow R_e \text{ is surjective.}$$

Proof. Consider two non-empty subsets $A, B \subseteq S$ and $i \in \langle M \rangle$. We prove the two statements independently.

(a) Using Definition 7.38 and 7.74,

$$\begin{aligned} e := (A, B, i) \in E_{\min} &\Leftrightarrow \forall a \in A, \exists b \in B : aR_e b, \\ &\Leftrightarrow R_e \text{ is total,} \\ &\Leftrightarrow \exists R \subseteq R_e \text{ s.t. } R \text{ is an application,} \end{aligned}$$

since it is always possible to remove some pairs of a total relation such that the corresponding contained relation is an application (i.e. one just keeps one pair for each element in the domain).

(b) Similarly, we can characterize the definition of an edge of the max lift using the relation in Definition 7.74, i.e.

$$\begin{aligned} e := (A, B, i) \in E_{\max} &\Leftrightarrow \forall b \in B, \exists a \in A : aR_e b, \\ &\Leftrightarrow R_e \text{ is surjective.} \end{aligned}$$

This ends the proof. □

In contrast to the min and max lifts, the nodes of the sum lift are associated to multi-sets rather than sets. In order to characterize the edges of the sum lift in terms of relation, we first need to define a binary relation over multi-sets.

Definition 7.76 (Binary relation over mutli-sets)**.** Given two multi-sets X and Y (possibly of different cardinality) of a universe S_1 and S_2 respectively, a *multi-set relation mR over X and Y* is defined by

$$R \subseteq \{((x, m_X(x)), (y, m_Y(y))) \mid x \in S_1, \, y \in S_2\}.$$

If $((x, m_X(x)), (y, m_Y(y))) \in mR$, we say that $(x, m_X(x))$ *is in multi-set relation with* $(y, m_Y(y))$, which is also denoted by $(x, m_X(x))mR(y, m_Y(y))$.

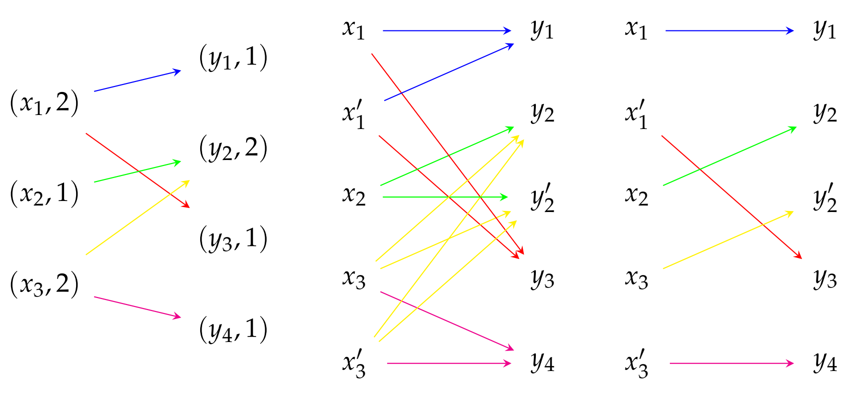
Remark 7.77. Given a multi-set relation in Definition 7.76, it is possible to induce a relation over sets as defined in Definition 7.71.

To do so, one can build a set from a multi-set by distinguishing the recurrent elements of the multi-set, as illustrated in Figures 7.23a and 7.23b. As a consequence, all the properties in Definition 7.72 can be defined for a multi-set relation by requiring the same property on a contained relation of the corresponding relation over sets. For instance, a multi-set relation mR over the multi-sets X and Y of the universe S_1 and S_2 respectively is *surjective* if and only if $\forall y \in S_1, \exists x_1, \ldots, x_l \in S_2$:

$$\sum_{i=1}^{l} m_X(x_i) \geq m_Y(y) \text{ and } (x_i, m_X(x_i))mR(y, m_Y(y)).$$

Figure 7.23 provides an example of a bijective multi-set relation and the corresponding bijective relation over sets. △

In particular, following Remark 7.77, we can extend the Definition 7.74 to an edge with multi-sets as nodes. We are now able to characterize an edge of the sum lift.



(a) A bijective multi-set relation mR over $\{x_1, x_1, x_2, x_3, x_3\}$ and $\{y_1, y_2, y_2, y_3, y_4\}$.

(b) The corresponding relation R over the sets $\{x_1, x_1', x_2, x_3, x_3'\}$ and $\{y_1, y_2, y_2', y_3, y_4\}$.

(c) The bijective relation contained in R over $\{x_1, x_1', x_2, x_3, x_3'\}$ and $\{y_1, y_2, y_2', y_3, y_4\}$.

Fig. 7.23 Illustration of the construction described in Remark 7.77 to derive a relation R from a multi-set relation mR. In this example, the relation is bijective.

Proposition 7.78. *Consider a graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$. Given $A, B \in \mathrm{Multi}^T(S)$ and $i \in \langle M \rangle$,*

$$e := (A, B, i) \in E^{\oplus T} \Leftrightarrow \exists R \subseteq R_e \text{ s.t. } R \text{ is a surjective application.}$$

Proof. By Definition 7.9 of an edge of the T-sum lift and by the characterization in Proposition 7.16,

$$\begin{aligned} e := (A, B, i) \in E^{\oplus T} \quad &\Leftrightarrow \quad \exists R \subseteq R_e \text{ s.t. } R \text{ is bijective,} \\ &\Leftrightarrow \quad \exists R \subseteq R_e \text{ s.t. } R \text{ is a surjective application,} \end{aligned}$$

where we have used Remark 7.73 to end the proof. □

Propositions 7.75 and 7.78 suggest that an edge (A, B, i) intuitively needs to satisfy both constraints of the min and max lifts to be added in the T-sum lift. The sum lift seems therefore *stronger* than the min and max lifts. The following proposition formalizes this idea.

Proposition 7.79. *Consider a graph $\mathcal{G} = (S, E)$. For any $T \in \mathbb{N}$, $\mathcal{G}_{\min}$ and $\mathcal{G}_{\max}$ simulate $\mathcal{G}^{\oplus T}$.*

Proof. Let us prove that $\mathcal{G}_{\min} = (S_{\min}, E_{\min})$ simulates $\mathcal{G}^{\oplus T} = (S^{\oplus T}, E^{\oplus T})$, and define a function $R : S^{\oplus T} \to S_{\min}$ such that any multi-set P of cardinality T in $S^{\oplus T}$ is mapped by R to the corresponding set by removing the repetitions (e.g., $R(\{a, a, b, e, e, e\}) = \{a, b, e\}$). Let us now prove that the function R is a simulation relation. Consider an edge $(P, Q, i) \in E^{\oplus T}$. Then, for any $p \in P$, there exists $q \in Q$ such that $(p, q, i) \in E$. This implies that for any $p \in R(P), \exists q \in R(Q) : (p, q, i) \in E$, i.e. $(R(P), R(Q), i) \in E_{\min}$.

The proof that $\mathcal{G}_{\max}$ simulates $\mathcal{G}^{\oplus T}$ follows a similar argument. □

Note that we could have proved Proposition 7.79 using Propositions 7.75 and 7.78. Thanks to this result, and the sufficient Theorems 7.55 and 7.56 for the min and the max lifts, we can derive the following result.

Corollary 7.80. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$. If there exists $T \in \mathbb{N}$ such that $\mathcal{G}^{\oplus T}$ simulates $\widetilde{\mathcal{G}}$, then $\mathcal{G}_{\min}$ simulates $\widetilde{\mathcal{G}}$. The same result holds for the max lift.*

Proof. Consider $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ for which there exists $T \in \mathbb{N}$ such that $\mathcal{G}^{\oplus T}$ simulates $\widetilde{\mathcal{G}}$. By Proposition 7.79, $\mathcal{G}_{\min}$ simulates $\mathcal{G}^{\oplus T}$. Then, by transitivity, $\mathcal{G}_{\min}$ simulates $\widetilde{\mathcal{G}}$ and Theorem 7.55 ends the proof.

The proof for the pointwise maximum follows a similar argument. □

Propositions 7.75 and 7.78 suggest that the reverse implication does not hold. Indeed, the following example provides a counterexample.

Example 7.81. Consider the graph $\mathcal{G}_7 = (S_7, E_7)$ in Figure 7.9. One can see that this graph is complete and co-complete since each node admits two outgoing and two incoming edges, one for each label. Therefore, by Proposition 7.52, $\mathcal{G}_7 \leq_{\mathcal{V}} \mathcal{G}_0$ for any template $\mathcal{V}$ closed under pointwise minimum or maximum. However, we have proved in Example 7.27 that there does not exist a value $T \in \mathbb{N}$ such that ${\mathcal{G}_7}^{\oplus T}$ simulates $\mathcal{G}_0$. △

7.7 De Bruijn hierarchy

In this section, we show how the templates of primal and dual copositive norms, respectively denoted by $\mathcal{P}$ and $\mathcal{D}$, can provide an estimation with arbitrary accuracy of the JSR of a set of nonnegative matrices, using a path-complete Lyapunov approach. As a reminder, $\mathcal{P}$ and $\mathcal{D}$, introduced in Section 2.2.3 and further studied in Section 7.4, are the dual templates of candidate Lyapunov functions of the form

$$g_v(x) := v^\top x,$$

and

$$g_v^\star(x) := \max_i \left\{ \frac{x_i}{v_i} \right\},$$

respectively, with $v \in \mathbb{R}^n_{>0}$.

While a similar theoretic estimation was already provided in [AJPR14, Section 6] for generic matrices and quadratic Lyapunov functions, our result will lead to a new hierarchy of *linear programs* (instead of semidefinite programs), thus drastically reducing the computation complexity. We recall that previous hierarchies of LPs (approximating the JSR of nonnegative matrices), as the one proposed in [PJB10, Corollary 3], require, in general, the computation of long products of matrices in the considered set. Our approach, instead, is not affected by this drawback. This is due to requiring a more complex structure of the candidate path-complete Lyapunov function, as we will develop in what follows. Moreover, we show how the results concerning *lifts* presented in this chapter specialize in this setting, leading to “smart” choices of graph structures for the stability analysis.

In this section, we use the notations $\rho_{\mathcal{P}}(\mathcal{A})$ and $\rho_{\mathcal{D}}(\mathcal{A})$ defined in Equations (2.12) and (2.13) which intuitively represent the best estimate of the joint spectral radius of a finite set of nonnegative matrices $\mathcal{A}$ one can obtain considering common copositive primal and dual Lyapunov norms, respectively. By Theorem 2.35, the approximation guarantees provided by $\rho_{\mathcal{P}}(\mathcal{A})$ and $\rho_{\mathcal{D}}(\mathcal{A})$ are the same. On the other hand, for *a specific set* $\mathcal{A} \subset \mathbb{R}^{n\times n}_{\geq 0}$, it is possible that one template (primal/dual copositive norms) will lead to a better estimation of the JSR. The following simple example shows how, for a particular set of matrices $\mathcal{A}$, the choice of primal or dual copositive norms is crucial in estimating the joint spectral radius, and in particular how the inequalities in Items *1.* and *2.* of Theorem 2.35 can be tight.

Example 7.82. Fix a dimension $n \in \mathbb{N}$ and consider $\mathcal{A} := \{A_1, \dots, A_n\} \subset \mathbb{R}^{n\times n}_{\geq 0}$ defined by $A_i = \mathbf{1}\,\mathbf{e}_i^\top$ for $i \in \langle n \rangle$, i.e.

$$A_1 = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & \dots & \vdots \\ 1 & \dots & 0 \end{bmatrix}, \dots, A_n = \begin{bmatrix} 0 & \dots & 1 \\ \vdots & \dots & \vdots \\ 0 & \dots & 1 \end{bmatrix},$$

where $\mathbf{1} := (1 \dots, 1)^\top$. From a straightforward computation, it holds that $\rho(\mathcal{A}) = 1$. Computing, we have, for all $x \in \mathbb{R}^n_{\geq 0}$, $A_i x = x_i \mathbf{1}$, and thus considering the norm $g^\star_{\mathbf{1}}$ (the usual infinity norm) we have that, for all $i \in \langle n \rangle$ and for any $x \in \mathbb{R}^n_{\geq 0}$,

$$g^\star_{\mathbf{1}}(A_i x) \leq g^\star_{\mathbf{1}}(x),$$

proving that $\rho_{\mathcal{D}}(\mathcal{A}) = \rho(\mathcal{A})$. In other words, dual copositive common Lyapunov norms provide an exact estimation of the JSR. It can be seen that $\rho_{\mathcal{P}}(\mathcal{A}) = n$, that is, by Theorem 2.35, the worst possible estimate. The "dual case", i.e. considering $\mathcal{A}^\top$, provides an example for which the primal norms provides an exact estimate, and the dual ones the worst possible. △

This example shows that for specific systems, it is sometimes useful to compute the approximations provided by both templates of primal and dual copositive norms. We now show that, considering *multiple* copositive primal/dual norms (or, more precisely, path-complete Lyapunov functions in these templates), we can provide an estimation of the JSR with arbitrary accuracy. Given $\mathcal{A} := \{A_1, \dots, A_M\} \subset \mathbb{R}^{n\times n}$ and a path-complete graph $\mathcal{G} = (S, E)$ the quantities $\rho_{\mathcal{G},\mathcal{P}}(\mathcal{A})$ and $\rho_{\mathcal{G},\mathcal{D}}(\mathcal{A})$ are defined as in Definition 6.17 and therefore correspond to the optimal values of the problems

$$\rho_{\mathcal{G},\mathcal{P}}(\mathcal{A}) := \inf_{v_1,\ldots v_{|S|}\in\mathbb{R}^n_{>0},\gamma\geq 0} \gamma \\ A_i^\top v_b - \gamma v_a \leq 0, \quad \forall\, e = (a,b,i) \in E, \tag{7.26}$$

and

$$\rho_{\mathcal{G},\mathcal{D}}(\mathcal{A}) := \inf_{v_1,\ldots v_{|S|}\in\mathbb{R}^n_{>0},\gamma\geq 0} \gamma \\ A_i v_b - \gamma v_a \leq 0, \quad \forall\, e = (a,b,i) \in E, \tag{7.27}$$

using Propositions 2.32 and 2.33. We can state the following "Asymptotic" converse Lyapunov theorem.

Theorem 7.83. *Let $\mathcal{A} = \{A_1, \ldots, A_M\} \subset \mathbb{R}^{n\times n}_{\geq 0}$. Given any $l \in \mathbb{N}$, considering $\mathcal{G}^l_{\mathrm{db}} = (S, E)$ the (primal) De Bruijn graph of order $l-1$ on $\langle M\rangle$, we have*

$$\frac{1}{\sqrt[l]{n}}\rho_{\mathcal{D},\mathcal{G}^l_{\mathrm{db}}}(\mathcal{A}) \leq \rho(\mathcal{A}) \leq \rho_{\mathcal{D},\mathcal{G}^l_{\mathrm{db}}}(\mathcal{A}). \tag{7.28}$$

Proof. Recalling Definition 6.10 and Proposition 2.33, $\mathcal{G}^l_{db} = (S, E)$ leads to the inequalities:

$$v_{(i_1,\ldots,i_{l-1})} > 0, \quad \forall (i_1, \ldots, i_{l-1}) \in \langle M\rangle^{l-1} \\ A_j v_{(i_1,\ldots,i_{l-1})} \leq \gamma v_{(i_2,\ldots,i_{l-1},j)}, \quad \forall (i_1, \ldots, i_{l-1}) \in \langle M\rangle^{l-1}, \forall j \in \langle M\rangle. \tag{7.29}$$

We now prove that $\rho_{\mathcal{G}^l_{db},\mathcal{P}}(\mathcal{A})$, that is the minimum γ for which Equation (7.29) is feasible, satisfies the inequalities in Equation (7.28). The inequality $\rho(\mathcal{A}) \leq \rho_{\mathcal{P},\mathcal{G}^l_{db}}(\mathcal{A})$ is straightforward. Consider now $\mathcal{A}^l$, the set of all the possible products of matrices in $\mathcal{A}$ of length l. By Theorem 2.35, we have $\frac{1}{n}\rho_{\mathcal{P}}(\mathcal{A}^l) \leq \rho(\mathcal{A}^l) = \rho(\mathcal{A})^l$ and thus

$$\frac{1}{\sqrt[l]{n}}\sqrt[l]{\rho_{\mathcal{P}}(\mathcal{A}^l)} \leq \rho(\mathcal{A}). \tag{7.30}$$

We suppose thus that $v \in \mathbb{R}^n_{>0}$ is such that $g^\star_v$ is a common copositive dual Lyapunov norm for $\mathcal{A}^l$ with decay $\gamma^l > 0$, i.e.

$$v > 0, \\ A_{i_l}\cdots A_{i_1} v - \gamma^l v \leq 0, \quad \forall (i_1, \ldots, i_l) \in \langle M\rangle^l.$$

For any $(i_1, \dots, i_{l-1}) \in \langle M \rangle^{l-1}$, defining

$$v_{(i_1,\dots,i_{l-1})} \;:=\; v + \frac{1}{\gamma} A_{l-1} v + \frac{1}{\gamma^2} A_{i_{l-1}} A_{i_{l-2}} v \;+\; \dots \;+\; \frac{1}{\gamma^{l-1}} A_{i_{l-1}} \cdots A_{i_1} v,$$

it is easy to see that inequalities in Equation (7.29) are satisfied. We have thus proved that $\rho_{\mathcal{G}^l_{db},\mathcal{D}}(\mathcal{A}) \leq \sqrt[l]{\rho_{\mathcal{D}}(\mathcal{A}^l)}$, and recalling Equation (7.30) we conclude. □

A similar proof is used in [AJPR14, Theorem 6.2] for the template of quadratic functions, obtaining an approximation guarantee for a hierarchy of semidefinite programs.

Remark 7.84. We note that, since the graph $\mathcal{G}^l_{db}$ is complete, recalling Proposition 7.52, if the inequalities in Equation (7.29) are feasible (for a certain $\gamma > 0$), we also have that the function defined by

$$V_{\mathcal{G}^l_{db}}(x) \;:=\; \min_{(i_1,\dots,i_{l-1}) \in \langle M \rangle^{l-1}} \left\{ g^\star_{v_{(i_1,\dots,i_{l-1})}}(x) \right\}$$

is a common Lyapunov function for (2.8). From Theorem 7.83 we can obtain its dual result: applying again the duality relation in Proposition 6.27 we have that, for any $l \in \mathbb{N}$,

$$\frac{1}{\sqrt[l]{n}} \rho_{(\mathcal{G}^l_{db})^\top,\mathcal{P}}(\mathcal{A}) \;\leq\; \rho(\mathcal{A}) \;\leq\; \rho_{(\mathcal{G}^l_{db})^\top,\mathcal{P}}(\mathcal{A}),$$

where $(\mathcal{G}^l_{db})^\top$ denotes the dual De Bruijn graph of order $l-1$. Moreover, the conditions encoded in $(\mathcal{G}^l_{db})^\top$ define, again by duality, a common Lyapunov function for (2.8), in the form of a *max* of primal copositive norms. We note that (convex hull of) min of dual copositive norms and max of primal copositive norms are special cases of *polyhedral functions*. In this view, Theorem 7.83 states in particular that, if the system (2.8) is asymptotically stable, then there exists a copositive polyhedral common Lyapunov function. This is consistent with the universality of polyhedral Lyapunov functions for switched systems proved in [BM99], see also [AJ19]. △

Given $\mathcal{A} = \{A_1, \dots, A_M\} \subset \mathbb{R}^{n\times n}_{\geq 0}$, Example 7.82 and Theorem 7.83 (and the subsequent Remark 7.84) suggest the following numerical scheme in order to approximate $\rho(\mathcal{A})$ with arbitrary precision, using the hierarchies of primal and dual De Bruijn graphs. This scheme is summarized in the following pseudo-algorithm.

Algorithm 7.85 (Numerical approximation of the JSR via De Bruijn Hierarchy)**.** *Given* $\mathcal{A} = \{A_1, \dots, A_M\} \subset \mathbb{R}^{n\times n}_{\geq 0}$*,*

(Init.)*: Fix a margin* $\varepsilon > 0$*, set* $l = 1$*,* $\underline{\gamma} = 0$*,* $\overline{\gamma} = +\infty$*.*

since $(\overline{\gamma} - \underline{\gamma} \geq \varepsilon)$*,*

(Step l)*: Solve the linear program in* (7.27) *for* $\mathcal{G}^l_{db}$*.*
Set $\underline{\gamma} \leftarrow \max\{\underline{\gamma},\ \frac{1}{\sqrt[l]{n}}\rho_{\mathcal{G}^l_{\mathrm{db}},\mathcal{D}}(\mathcal{A})\}$ *and*
$\overline{\gamma} \leftarrow \min\{\overline{\gamma},\ \rho_{\mathcal{G}^l_{\mathrm{db}},\mathcal{D}}(\mathcal{A})\}$*.*

(Step l^d)*: Solve the linear program in* (7.26) *for* $(\mathcal{G}^l_{db})^\top$*.*
Set $\underline{\gamma} \leftarrow \max\{\underline{\gamma},\ \frac{1}{\sqrt[l]{n}}\rho_{(\mathcal{G}^l_{\mathrm{db}})^\top,\mathcal{P}}(\mathcal{A})\}$ *and*
$\overline{\gamma} \leftarrow \min\{\overline{\gamma},\ \rho_{(\mathcal{G}^l_{\mathrm{db}})^\top,\mathcal{P}}(\mathcal{A})\}$*.*

$l \leftarrow l + 1$*.*

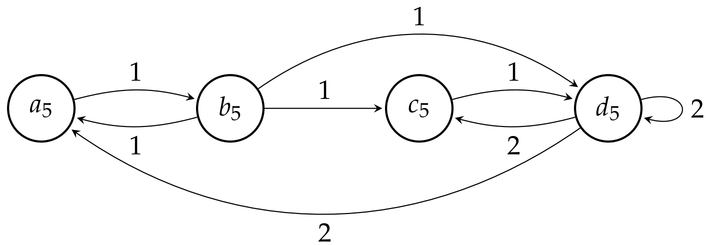
This procedure allows us, once a confidence margin $\varepsilon > 0$ is chosen, to provide tight estimations of JSR of nonnegative matrices. Other stopping criteria can be considered, as for example the condition $(\overline{\gamma} < 1)$ which ensures asymptotic stability of (2.8), or $(\underline{\gamma} > 1)$, which is an instability certificate for (2.8).

Let us consider a positive switched system as in (2.8) and let us provide the following analysis: first, given a particular path-complete graph $\mathcal{G}$, we see how, when considering dual copositive norms, the estimation of the JSR is improved considering the max lift $\mathcal{G}_{max}$, in line with Proposition 7.41. Secondly, applying the idea of Algorithm 7.85, we provide an accurate estimation of the joint spectral radius.

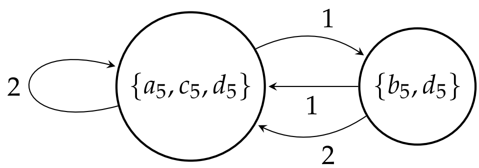
Example 7.86. We consider the positive switched system (2.8) defined by $\mathcal{A} = \{A_1, A_2\} \subset \mathbb{R}^{3\times 3}_{\geq 0}$ with

$$A_1 = \begin{bmatrix} 0.2 & 0 & 0 \\ 0.6 & 0.6 & 0.5 \\ 0.6 & 0.3 & 0.2 \end{bmatrix} \text{ and } A_2 = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.2 & 0 & 0.5 \\ 0.1 & 0.6 & 0.7 \end{bmatrix}. \tag{7.31}$$

First, in order to approximate $\rho(\mathcal{A})$, we solve the problem in Equation (7.27) for $\mathcal{G}_5$ in Figure 7.24a, obtaining $\rho_{\mathcal{G}_5,\mathcal{D}}(\mathcal{A}) = 1.3075$. We consider the max lift $(\mathcal{G}_5)_{\max}$ and in particular we select a path-complete and strongly connected component of $(\mathcal{G}_5)_{\max}$ given by $\mathcal{G}_6$ in Figure 7.24b. We know, by



(a) $\mathcal{G}_5 = (S_5, E_5)$, a path-complete graph on $\langle 2 \rangle$ with 4 nodes and 8 edges.



(b) $(\mathcal{G}_6) = (S_6, E_6)$, a strongly connected component of $(\mathcal{G}_5)_{\max}$.

Fig. 7.24 The two path-complete graphs $\mathcal{G}_5$ and $\mathcal{G}_6$ in Example 7.86.

Theorem 7.43 and Proposition 6.23 that $\mathcal{G}_5 \leq_{\mathcal{D},\mathcal{L}} \mathcal{G}_6$ and thus we expect that $\rho_{\mathcal{G}_6,\mathcal{D}}(\mathcal{A}) \leq \rho_{\mathcal{G}_5,\mathcal{D}}(\mathcal{A})$ which is confirmed, since solving (7.27), we obtain $\rho_{\mathcal{G}_6,\mathcal{D}}(\mathcal{A}) = 1.2716$. It is interesting to note how the graph $\mathcal{G}_6$, although it reduces the number of decision variables and inequalities with respect to the conditions encoded in $\mathcal{G}_5$, provides a better estimation of the JSR. Given a positive system, we know that $\mathcal{G}_6$ will provide at worst the same estimation as $\mathcal{G}_5$ and for some particular cases as (7.31), $\mathcal{G}_6$ will provide a strictly better approximation than $\mathcal{G}_5$. This highlights that, given a particular path-complete structure and a template, the lifting approach can provide a better estimation of the joint spectral radius while decreasing the number of Lyapunov inequalities and decision variables.

Concluding, we provide upper and lower bounds for $\rho(\mathcal{A})$ using the hierarchy described in Algorithm 7.85. For simplicity, we stop at the fourth iteration of the numerical scheme (and thus considering until the primal and dual De Bruijn graphs of order 3) obtaining the following results in Table 7.1. In this table, in the line denoted by $\rho_{\mathcal{G}}$ we reported the optimal values of the LPs described by (7.26), (7.27) for the corresponding (primal and dual) De Bruijn graphs. We have thus proven that $\rho(\mathcal{A}) \in [1.065, 1.070]$, having an instability certificate for the positive switched system (2.8) defined by $\mathcal{A}$. It is interesting to note how, in this particular case, the con-

Steps:	(1)	$(1)^d$	(2)	$(2)^d$	(3)	$(3)^d$	(4)	$(4)^d$
$\rho_{\mathcal{G}}(\mathcal{A})$	1.445	1.341	1.445	1.070	1.410	1.070	1.402	1.070
$\underline{\gamma}$	0.482	0.482	0.834	0.834	0.978	0.978	1.065	1.065
$\overline{\gamma}$	1.445	1.341	1.341	1.070	1.070	1.070	1.070	1.070

Table 7.1 Evolution with the iterations of the approximations of the JSR of system (7.31) using the De Bruijn hiearchy in Algorithm 7.85.

ditions arising from the primal De Bruijn graphs and the template of dual copositive norms provide better upper bounds for the JSR. △

7.8 Summary and further research directions

In this chapter, we studied the problem of establishing relations among different path-complete structures, with the goal of optimizing this structure, while at the same time controlling the computational cost. We have highlighted the strong connections between templates of candidate Lyapunov functions and the ordering relations between graph-based conditions.

Summary of Chapter 7

We propose a systematic way to compare different path-complete stability certificates, based on the notion of lifts. In order to explicitly exploit the analytical properties of the chosen template and dynamics, we introduce new abstract lifts related with a few properties, providing further insight for the comparison problem.

Section 7.1: Introduction to the lifts and their validity

We introduced new formal transformations of path-complete graphs, called lifts, that allow us to establish ordering relations between graphs. We analyzed how the effectiveness of these lifts strongly depends on the closure properties of the chosen template.

Section 7.2: Duality of lifts

Inspired by the duality introduced in Sections 1.2 and 6.3.2, we define the dual of a lift which leverages the properties of the dual template.

Sections 7.3 to 7.5: Sum, min, max and composition lifts

We introduced two classes of lifts: the template-dependent lifts (e.g. the sum lift and the min/max lifts), and the template and dynamics-dependent lifts whose validity depends on the template and dynamics properties. These lifts allowed us to generalize previous results and to provide a unifying framework, which enables for finer comparison criteria between path-complete techniques.

Section 7.6: Comparison of lifts

We have characterized the edges of the min, max and sum lifts using properties on binary relations. This allowed us to compare these lifts, and understand how they are related to each other.

Section 7.7: De Bruijn hierarchy

As particular case study, we thoroughly analyzed the template of primal and dual copositive norms, which provided a handy framework in order to provide new stability results, with applications to the stability analysis of positive switched systems. In particular, we develop a new hierarchy to approximate the JSR.

In Section 7.6, we have demonstrated that there exists a correspondence between the edges of the min, max and sum lifts and properties on the corresponding binary relation R_e in Definition 7.74. In particular, Propositions 7.75 and 7.78 use all the properties in Definition 7.72 to characterize the edges of the min, max and sum lifts. Since there does not exist any other common property for binary relations, this raises the question of the existence of another lift with nodes associates to multi-subsets.

Question 7.87. Is it possible to define another template-dependent lift for which the nodes are associated to subset or multi-subsets of the initial set of nodes ?

Note that the forward and backward composition lifts introduced in Section 7.5 do not answer Question 7.87 since their nodes depend on both the initial set of nodes and the different modes. Moreover, we did not manage to characterize their edges using the same formalism.

So far, we have introduced different lifts, each of which exploits a different closure property. However, a given template can admit several closure operations as in the case of quadratic functions for example, or polyhedral functions. This motivates to combine and apply consecutively different lifts such that the lifted graph fully harnesses the potential of the template.

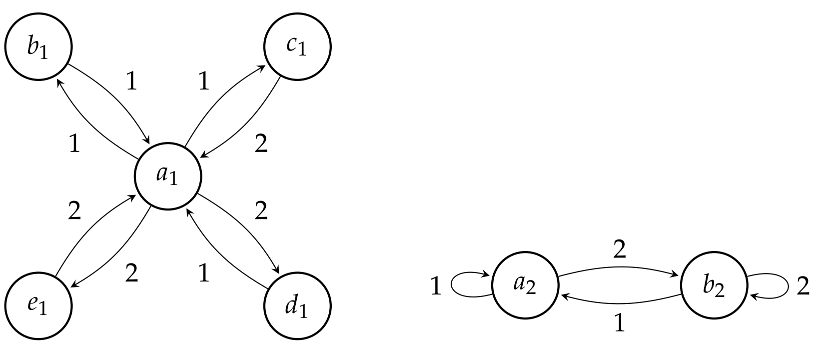
The first concrete example of the benefits of combining different lifts has already presented in Proposition 7.52. Indeed, the consecutive application of both the min and max lifts leads to the common Lyapunov function graph. This result provides two important insights. First, it is always possible to derive a common Lyapunov function from a multiple Lyapunov function. Secondly, if a template is closed under pointwise minimum and maximum, all the path-complete graphs are as conservative as the common Lyapunov function. In particular, this removes any interest in using this template in practice.

The following example, inspired by [PAAJ19, Example IV.11.], provides further motivation.

Example 7.88. Consider the path-complete graph $\mathcal{G}_1$ in Figure 7.25a and a linear switched system $\mathcal{A} := \{A_1, A_2\}$ of any dimension $n \in \mathbb{N}$. Assume that there exists $V \in PCLF(\mathcal{G}, \mathcal{A})$. Therefore, for any pair of labels $i, j \in \langle 2 \rangle$, the following inequality holds:

$$\forall x \in \mathbb{R}^n : V_{a_1}(A_i A_j x) \; \leq \; V_{a_1}(x). \tag{7.32}$$

We can prove that this property implies the existence of a PCLF for the



(a) $\mathcal{G}_1 = (S_1, E_1)$, a path-complete graph with 5 nodes and 8 edges.

(b) $\mathcal{G}_2 = (S_2, E_2)$, a path-complete graph with 2 nodes and 4 edges.

Fig. 7.25 Example of two path-complete graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ on $\langle 2 \rangle$ such that $\mathcal{G}_1 \leq_{\mathcal{V},\mathcal{L}} \mathcal{G}_2$ for any template $\mathcal{V}$ closed under addition and composition with invertible dynamics.

graph $\mathcal{G}_2$ in Figure 7.25b. Indeed, let us define $W := \{W_{a_2}, W_{b_2}\}$ by taking

$$\begin{cases} W_{a_2} & := & V_{a_1} + V_{a_1} \circ {A_1}^{-1}, \\ W_{b_2} & := & V_{a_1} + V_{a_1} \circ {A_2}^{-1}. \end{cases}$$

We can prove that $W \in PCLF(\mathcal{G}_2, \mathcal{A})$. For example, let us consider the edge $(a_2, b_2, 2) \in E_2$, which corresponds to

$$\forall x \in \mathbb{R}^n : \underbrace{V_{a_1}(A_2 x) + V_{a_1}(x)}_{:= W_{b_2}(A_2 x)} \leq \underbrace{V_{a_1}(x) + V_{a_1}({A_1}^{-1} x)}_{:= W_{a_2}(x)}.$$

This inequality is satisfied by evaluating the inequality (7.32) at the point ${A_1}^{-1}x$ for $i = 1$ and $j = 2$, which leads to $\forall x \in \mathbb{R}^n : V_{a_1}(A_2 x) \leq V_{a_1}({A_1}^{-1}x)$. A similar argument can be used for the other edges of $\mathcal{G}_2$.

This proves that $\mathcal{G}_1 \leq_{\mathcal{V},\mathcal{L}} \mathcal{G}_2$ for any template $\mathcal{V}$ closed under composition with invertible dynamics and addition. $\triangle$

To conclude this chapter, we want to point out that we have assumed from the beginning that we study the deterministic comparison of path-complete graphs, i.e. when the expression (6.10) is satisfied for any switched

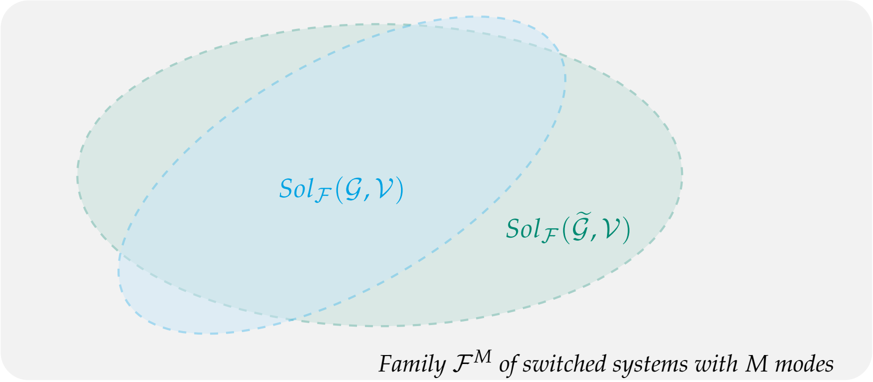


Fig. 7.26 Illustration, using the same notation as in Figure 6.7, of an example of two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ which cannot be compared using *deterministic* ordering relations in Definition 6.21. However, for *almost all the systems* for which the graph $\mathcal{G}$ admits a solution in $\mathcal{V}$, $\widetilde{\mathcal{G}}$ admits a solution in the template as well.

system (or within a specific family of switched systems), as illustrated in Figure 7.26. However in practice, this situation does not occur in most cases. The corresponding "statistical" approach would provide a new route for open research: given two path-complete graphs, we would like to provide a probabilistic conservatism-based relation between them. More precisely, given two graphs, we want to compute/approximate the probability, given a random (with respect to a certain probability distribution) switched system, that the estimation provided by the first graph is better than the one provided by the second. This approach has already been addressed in [APAJ17] for instance and is further studied in [SJ24, Jun24]. We believe that the notion of lift and the simulation-based sufficient condition for the template-dependent ordering of graphs could help compute these probabilities. Indeed we think that the template-dependent probabilistic comparison of graphs is highly related to "how far" there exists a simulation relation, i.e. how many edges must be modified to guarantee the simulation relation.

8

Characterization of the template-dependent ordering of graphs with lifts

INSPIRED by the complete characterization of the general ordering relation (6.12) in Theorem 6.30, we investigate in this section the same problem for the less restrictive ordering relation (6.11) for a family of templates that all share the same closure property. However in this setting, the characterization is more involved, as it considers a *template-dependent lift* of graphs, introduced in Chapter 7. By means of these concepts we provide the following equivalence result: a graph is more conservative than another for all the templates closed under addition (resp. min and max), if and only if the *sum lift* (resp. min and max lifts) of the first graph *simulates* the second graph. While the "if" part was already proved in Chapter 7, the "only if" is more challenging, in that it amounts to proving that whichever template is used, the algebraic closure property is completely expressed by the lift operation, which is a purely combinatorial operation on graphs.

We first provide a characterization of the template-dependent ordering of path-complete graphs for the specific classes of templates closed under pointwise minimum and maximum. In particular, given a switched system and a template, these results can help guide the search for a better stability certificate by checking the existence of a simulation relation.

We then consider another family of templates, namely the templates closed under addition. For this latter family we show that the situation is more complicated, essentially because the binary operation on functions given by the sum is not idempotent, contrary to the minimum or the maximum. Indeed, the sum lift of a given graph is a graph with infinitely many nodes. However, we manage to circumvent this difficulty, and provide a proof that one may restrict oneself to a finite truncation of this graph without loss of generality, providing a finite procedure for the decision procedure. As it turns out, this finite procedure can even be made to run in polynomial time.

We also illustrate how our work provides a general method and proof technique that can be used for more broader settings.

This work, which has been done in collaboration with Matteo Della Rossa, has been published in [DDJ22b, DDJ23].

8.1 Min/max lifts and minimum/maximum-closed templates

In this section, we deal with the closure property of pointwise minimum in Theorem 8.1 and we prove that the validity of the relation (6.11) for any template closed under pointwise minimum is captured by the min lift.

Theorem 8.1. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$. The following statements are equivalent:*

(1) $\mathcal{G}_{\min}$ simulates $\widetilde{\mathcal{G}}$.

(2) $\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ for any template $\mathcal{V}$ closed under pointwise minimum.

Our proof follows the same path of ideas as Theorem 6.30. However, some modifications are needed to manage the closure property. Therefore, we split the proof of Theorem 8.1 in two parts. We first prove a technical lemma that will be central in the main proof.

Then, we will use duality to prove a similar theorem (see Theorem 8.5) for the max lift.

8.1.1 Key lemma

We start by proving a technical result that states that for any graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, i.e. for any set of Lyapunov inequalities, it is possible to build a switched system F on M modes and a solution $V \in$

$PCLF(\mathcal{G}, F)$ for which the associated pointwise minimum Lyapunov functions satisfy the inequalities encoded by $\mathcal{G}_{\min}$ by construction, but none of the non-existing edges of $\mathcal{G}_{\min}$ hold. This construction has been implemented in MATLAB and can be found in [Deb22].

Lemma 8.2. *For any graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, there exist an integer $n \geq 1$, a system $F := \{f_i : i \in \langle M \rangle\}$ in dimension n and $|S|$ candidate Lyapunov functions V_s, $s \in S$ for which*

$$\forall (p, q, i) \in E,\ \forall x \in \mathbb{R}^n :\ V_q(f_i(x)) \leq V_p(x), \tag{8.1}$$

$$\forall (P, Q, i) \in \overline{E_{\min}},\ \exists \overline{x} \in \mathbb{R}^n :\ \min_{q \in Q} V_q(f_i(\overline{x})) > \min_{p \in P} V_p(\overline{x}), \tag{8.2}$$

where $\overline{E_{\min}} = (S_{\min} \times S_{\min} \times \langle M \rangle) \setminus E_{\min}$ refers to the set of non-existing edges *of $\mathcal{G}_{\min}$.*

Proof. Given a graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, we define a set of M block-diagonal $\{0, 1\}$-matrices $\{A_j : j \in \langle M \rangle\}$ of dimension $n = 2|\overline{E_{\min}}|$. Each block is associated to a non-existing edge $\tilde{e}$ of $\mathcal{G}_{\min}$, and is defined by

$$A_j[\tilde{e}] := \begin{cases} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} & \text{if } j = label(\tilde{e}), \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \text{otherwise,} \end{cases} \tag{8.3}$$

so that each matrix only acts on the blocks associated to the edges of the same label. We consider as template a finite set of *weighted L_1 norms*, i.e. for any $s \in S, \forall x \in \mathbb{R}^n$, $V_s(x) := {v_s}^\top |x|$, where $v_s \in \mathbb{R}^n_{>0}$ and $|x|$ denotes the componentwise absolute value of $x \in \mathbb{R}^n$. In this context, since $A_j \in \mathbb{R}^{n \times n}_{\geq 0}$ for any $j \in \langle M \rangle$, satisfying a Lyapunov inequality $(s, d, i) \in E$ amounts to satisfying a set of $|\overline{E_{\min,i}}|$ scalar inequalities where $\overline{E_{\min,i}} := \{\tilde{e} \in \overline{E_{\min}} \mid label(\tilde{e}) = i\}$ since

$$\begin{aligned} & \forall x \in \mathbb{R}^n,\ V_d(A_i x) \ \leq \ V_s(x), \\ \Leftrightarrow \quad & \qquad {A_i}^\top v_d \ \leq_c \ v_s, \\ \Leftrightarrow \quad & \forall \tilde{e} \in \overline{E_{\min,i}},\ v_d[\tilde{e}]_2 \ \leq \ v_s[\tilde{e}]_1, \end{aligned} \tag{8.4}$$

where $\leq_c$ depicts a componentwise inequality, i.e. if $a, b \in \mathbb{R}^n$, $a \leq_c b \Leftrightarrow \forall i \in \langle n \rangle, a_i \leq b_i$. Given an edge $\tilde{e} = (P, Q, i) \in \overline{E_{\min}}$, we denote by

$I(\tilde{e}) := \{p \in P \mid \forall q \in Q,\ (p,q,i) \notin E\}$. We define the blocks $v_s[\tilde{e}]$ for any $s \in S$ such that all the inequalities induced from E_i are satisfied but $\tilde{e}$ is violated. Therefore,

$$v_s[\tilde{e}]_1 := \begin{cases} 1 & \text{if } s \in I(\tilde{e}), \\ 3 & \text{otherwise.} \end{cases} \tag{8.5}$$

$$v_s[\tilde{e}]_2 := \begin{cases} 1 & \text{if } \exists l \in I(\tilde{e}) \text{ s.t. } (l,s,i) \in E, \\ 2 & \text{otherwise.} \end{cases} \tag{8.6}$$

We show now that this construction satisfies the expressions (8.1) and (8.2).

We begin with the expression (8.1) that is equivalent to (8.4) holding for any $(s,d,i) \in E$ here. Given both edges $e = (s,d,i) \in E$ and $\tilde{e} = (P,Q,i) \in \overline{E_{\min}}$, the inequality in (8.4) would be violated only if $v_s[\tilde{e}]_1 = 1$ and $v_d[\tilde{e}]_2 = 2$. However, this cannot happen since it implies that $(s,d,i) \notin E$. In all the other configurations, the inequality (8.4) is satisfied.

We now focus on (8.2). Consider $\tilde{e}_1 = (P_1, Q_1, i) \in \overline{E_{\min}}$, by the same argument as in (8.4) and Corollary 2.36, we have to prove that

$$\exists \tilde{e}_2 = (P_2, Q_2, i) \in \overline{E_{\min}} : \bigvee_{q \in Q_1} v_q[\tilde{e}_2]_2 > \bigvee_{p \in P_1} v_p[\tilde{e}_2]_1, \tag{8.7}$$

where $v_a \vee v_b$ denotes the componentwise minimum between the vectors v_a and v_b. We show that we can choose $\tilde{e}_2 = \tilde{e}_1$ to achieve this. Since $\tilde{e}_1 \in \overline{E_{\min}}$, $I(\tilde{e}_1)$ is not empty and the minimum of $v_p[\tilde{e}_1]_1$ over P_1 is 1. Moreover, for any $q \in Q_1$, $v_q[\tilde{e}_1]_2 = 2$ because for all $l \in I(\tilde{e}_1)$ and for all $q \in Q_1$, $(l,q,i) \notin E$ by definition of $I(\tilde{e}_1)$. Then, the minimum of $v_q[\tilde{e}_1]_2$ over Q_1 is 2. This concludes the proof of Lemma 8.2. □

Note that this result is stronger than Lemma 6.31 since it requires to violate all the non-edges of the min lift, some of which are the non-edges of the initial graph. Therefore, Lemma 8.2 trivially implies Lemma 6.31.

Example 8.3. Consider the graph $\mathcal{G}_1 = (S_1, E_1)$ in Figure 7.14a and its min lift $(\mathcal{G}_1)_{\min}$ in Figure 7.15.

Let us apply the construction in the proof of Lemma 8.2 to $\mathcal{G}_1$. Since there are 6 non-edges in $(\mathcal{G}_1)_{\min}$ (4 for the initial graph, and 2 for the added nodes), the matrices in (8.3) are of dimension 12, as well as the vectors $\{v_{a_1}, v_{b_1}\}$. These vectors are illustrated in Figure 8.1, where each 2-dimensional block is associated to one of the non-edges of $(\mathcal{G}_1)_{\min}$.

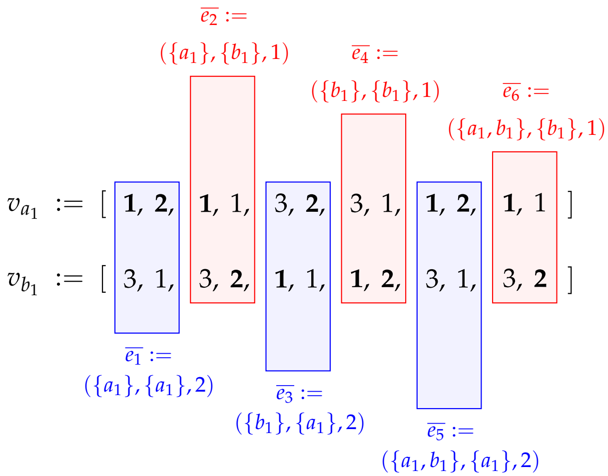


Fig. 8.1 Illustration of the procedure described in the proof of Lemma 8.2 to build the Lyaunov functions for the path-complete graph $\mathcal{G}_1 := (S_1, E_1)$ in Figure 7.14a. Each 2-dimensional block is associated to one of the non-edges of $(\mathcal{G}_1)_{\min}$ of label 1 in red and of label 2 in blue. By construction, the block associated to $e \in \overline{E_{1\min}}$ violates the Lyapunov inequality associated to e. The values that make this possible are highlighted in bold.

We can finally verify the statements (8.1) and (8.2) of Lemma 8.2. Recalling (8.4) and (8.7), we can easily check whether the vectors $\{v_{a_1}, v_{b_1}\}$ in Figure 8.1 satisfy the Lyapunov inequalities encoded by $\mathcal{G}_1$ and violate the non-edges of $(\mathcal{G}_1)_{\min}$. Table 8.1 summarizes this verification. $\triangle$

We already noted the duality relation between the max and min lifts in Lemma 7.45. This allows us to obtain a result similar to Lemma 8.2 for the max lift.

Lemma 8.4. *For any graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, there exist an integer $n \geq 1$, a system $F := \{f_i : i \in \langle M \rangle\}$ in dimension n and $|S|$ candidate Lyapunov functions U_s, $s \in S$ for which*

$$\forall (p, q, i) \in E,\ \forall x \in \mathbb{R}^n :\ U_q(f_i(x)) \leq U_p(x), \tag{8.8}$$

$$\forall (P, Q, i) \in \overline{E_{\max}},\ \exists \overline{x} \in \mathbb{R}^n :\ \max_{q \in Q} U_q(f_i(\overline{x})) > \max_{p \in P} U_p(\overline{x}), \tag{8.9}$$

where $\overline{E_{\max}} = (S_{\max} \times S_{\max} \times \langle M \rangle) \setminus E_{\max}$ refers to the set of non-existing edges *of $\mathcal{G}_{\max}$.*

Edge (P,Q,i)	$\bigvee_{q\in Q} v_q[\overline{e}]_2 \overset{?}{\leq} \bigvee_{p\in P} v_p[\overline{e}]_1$ $\overline{e_1}$	$\overline{e_2}$	$\overline{e_3}$	$\overline{e_4}$	$\overline{e_5}$	$\overline{e_6}$
$(a_1,a_1,1) \in E_1$		$1 \leq 1$		$1 \leq 3$		$1 \leq 1$
$(a_1,b_1,2) \in E_1$	$1 \leq 1$		$1 \leq 3$		$1 \leq 1$	
$(b_1,b_1,2) \in E_1$	$1 \leq 3$		$1 \leq 1$		$1 \leq 3$	
$(b_1,a_1,1) \in E_1$		$1 \leq 3$		$1 \leq 1$		$1 \leq 3$
$(\{a_1\},\{a_1\},2) \in \overline{E_{1\min}}$	**2 > 1**		$2 \leq 3$		$2 > 1$	
$(\{a_1\},\{b_1\},1) \in \overline{E_{1\min}}$		**2 > 1**		$2 \leq 3$		$2 > 1$
$(\{b_1\},\{a_1\},2) \in \overline{E_{1\min}}$	$2 \leq 3$		**2 > 1**		$2 \leq 3$	
$(\{b_1\},\{b_1\},1) \in \overline{E_{1\min}}$		$2 \leq 3$		**2 > 1**		$2 \leq 3$
$(\{a_1,b_1\},\{a_1\},2) \in \overline{E_{1\min}}$	$2 > 1$		$2 > 1$		**2 > 1**	
$(\{a_1,b_1\},\{b_1\},1) \in \overline{E_{1\min}}$		$2 > 1$		$2 > 1$		**2 > 1**

Table 8.1 Illustration of the verification of the Lyapunov inequalities using the outcome of the proof of Lemma 8.2 for the graph $\mathcal{G}_1$ in Figure 7.15. As expected, the solution in Figure 8.1 satisfies the Lyapunov inequalities encoded by the graph but violates all the non-edges of the min lift.

Proof. Our construction is derived from the one in the proof of Lemma 8.2. In particular, the (linear) system F given by matrices $\{\overline{A}_1, \dots, \overline{A}_M\}$ and the functions U_s are obtained from the ones introduced in the proof of Lemma 8.2, simply defining

$$\overline{A}_j := A_j^\top, \; j \in \langle M \rangle \text{ and } U_s(x) := \max_{i\in\langle n\rangle} \left\{ \frac{|x_i|}{v_{si}} \right\}, \; s \in S,$$

with matrices A_j defined in (8.3) and vectors v_s defined in (8.5) and (8.6). Indeed, the functions U_s are the conjugate functions of the V_s in proof of Lemma 8.2, and by convex duality theory (see Lemma 1.28) we have that, given any convex functions $g_1, g_2 \in \mathcal{C}^0(\mathbb{R}^n, \mathbb{R}^n)$ and $A \in \mathbb{R}^{n\times n}$, the following equivalence holds:

$$g_2(Ax) \leq g_1(x) \;\; \forall x \in \mathbb{R}^n \;\; \Leftrightarrow \;\; g_1^\star(A^\top x) \leq g_2^\star(x), \;\; \forall x \in \mathbb{R}^n,$$

concluding the proof. □

8.1.2 Main characterization theorem

Using Lemma 8.2, it is now possible to prove the main Theorem 8.1.

Proof of Theorem 8.1. (1) $\Rightarrow$ (2): This implication is proved by Theorem 7.55. (2) $\Rightarrow$ (1): Our proof follows the same path of ideas as Theorem 6.30. Consider two graphs $\mathcal{G}$, $\widetilde{\mathcal{G}}$ and suppose that $\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ for any template $\mathcal{V}$ closed under pointwise minimum. First, by applying Lemma 8.2 to $\mathcal{G}$, we obtain a system $F := \{f_i : i \in \langle M \rangle\}$ and a set of candidate Lyapunov functions $\{V_s : s \in S\}$ such that the Lyapunov inequalities encoded by the edges of $\mathcal{G}$ are satisfied and none of the non-existing edges of $\mathcal{G}_{\min}$ are satisfied, i.e.

$$\forall (p_1, q_1, i) \in E,\ \forall\, x \in \mathbb{R}^n :\ V_{q_1}(f_i(x)) \leq V_{p_1}(x), \tag{8.10}$$

$$\forall (P_1, Q_1, i) \notin E_{\min},\ \exists\, x \in \mathbb{R}^n :\ \min_{q \in Q_1} V_q(f_i(x)) > \min_{p \in P_1} V_p(x). \tag{8.11}$$

Let us define the family $\mathcal{F} = \{F\}$ and the template

$$\mathcal{V} \ :=\ \{W_{P_1} := \min_{p \in P_1} V_p \mid P_1 \in S_{\min}\}$$

where $S_{\min} = \mathcal{P}(S) \setminus \emptyset$. Then, $\{V_s : s \in S\} \subset \mathcal{V}$ and the template $\mathcal{V}$ is closed under minimum such that $\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$. Obviously, there exists a solution admissible for $\mathcal{G}$ and F in $\mathcal{V}$. Then, by assumption, there exists a set of Lyapunov functions in the template $\mathcal{V}$ which are admissible for F and $\widetilde{\mathcal{G}}$, i.e. there exist $\{U_{\widetilde{s}} : \widetilde{s} \in \widetilde{S}\} \subseteq \mathcal{V}$ that satisfy the Lyapunov inequalities encoded by $\widetilde{\mathcal{G}}$. Since these functions belong to the template $\mathcal{V}$, we can associate a subset of S to each node of $\widetilde{\mathcal{G}}$, i.e. we can define a function $R : \widetilde{S} \to S_{\min}$ such that $U_{\widetilde{s}} = W_{R(\widetilde{s})}$.

Finally, we just have to prove that this function R satisfies the definition of simulation, i.e.

$$\forall (p_2, q_2, i) \in \widetilde{E},\ (R(p_2), R(q_2), i) \in E_{\min}.$$

Assume by contradiction that there exists an edge $(p_2, q_2, i) \in \widetilde{E}$ such that $(R(p_2), R(q_2), i) \notin E_{\min}$. Using Lemma 8.2, this means that there exists $\overline{x} \in \mathbb{R}^n$ such that

$$U_{R(q_2)}(\overline{x}) \ :=\ \min_{q \in R(q_2)} V_q(f_\sigma(\overline{x})) \ >\ \min_{p \in R(p_2)} V_p(\overline{x}) \ :=\ U_{R(p_2)}(\overline{x})$$

i.e. the set $\{U_{\tilde{s}} : \tilde{s} \in \widetilde{S}\}$ is not admissible. However these inequalities are satisfied by construction of the functions $U_{\tilde{s}}$, here is the contradiction. □

We can now obtain a result similar to Theorem 8.1 for the max lift and the class of templates closed under pointwise maximum.

Theorem 8.5. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$. The following statements are equivalent:*

(1) $\mathcal{G}_{\max}$ simulates $\widetilde{\mathcal{G}}$.

(2) $\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ for any template $\mathcal{V}$ closed under pointwise maximum.

Proof. We only sketch the proof, which follows the one of Theorem 8.1. We develop here, avoiding some details, the main ideas.

Similarly to Theorem 8.1, the implication (1) ⇒ (2) has already been proved in Theorem 7.56.

Regarding the reverse implication, we consider two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ such that $\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ for any template $\mathcal{V}$ closed under pointwise maximum. Applying Lemma 8.4, we obtain a system F and a set of candidate Lyapunov functions $\{U_s : s \in S\}$ such that (8.8) and (8.9) are satisfied. Define the family $\mathcal{F} := \{F\}$ and the pointwise maximum closure template

$$\mathcal{V} := \{Z_{P_1} := \max_{p \in P_1} U_p \mid P_1 \in S_{\max}\}.$$

By construction, there exists a solution in $\mathcal{V}$ admissible for F and $\mathcal{G}$. Then, by assumption, we can find a solution $\{Y_{\tilde{s}} : \tilde{s} \in \widetilde{S}\}$ in $\mathcal{V}$ admissible for F and $\widetilde{\mathcal{G}}$ that implicitly defines a function $R : \widetilde{S} \to S_{\max}$ such that $Y_{\tilde{s}} := Z_{R(\tilde{s})}$. We conclude following the same reasoning by contradiction as in the proof of Theorem 8.1. □

8.2 Sum lift and addition-closed templates

In this section, we continue the analysis and provide a complete characterization of templates closed *under addition*. The main technical tools in our proofs are the notion of *sum lift* of graphs, already introduced in Chapter 7, and the concept of simulation between graphs already used, for comparison of path-complete criteria, in [PJ19]. By means of these concepts we provide the following equivalence result: a graph is more conservative than another for all the templates closed under addition, if and only if the *sum lift* of the first *simulates* the second graph.

Theorem 8.6. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$. The following statements are equivalent:*

(1) $\mathcal{G}^{\oplus}$ simulates $\widetilde{\mathcal{G}}$.

(2) $\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ for any template $\mathcal{V}$ closed under addition.

As already mentioned, the "if" part of Theorem 8.6 was already proved in Theorem 7.18, but the "only if" part is more challenging, since it amounts to prove that the combinatorial operation of the sum lift completely characterizes the addition-closed property, regardless of the template.

As a second challenge in our proofs, the sum lift of a given graph is a graph with infinitely many nodes, such that the simulation in Theorem 8.6 cannot be checked. However, using Theorem 7.35, we will prove that this simulation relation can be verified in polynomial time.

8.2.1 Key lemma

Following the path of ideas of the proof of Theorem 8.1, we need to consider the sum-closure template of a set of Lyapunov functions. However, this template will contain a countable infinite number of functions whereas both the min and max-closure templates admit a finite number of elements due to the idempotent character of these closure properties. This major difference is reflected by the fact the the sum lift is defined on an infinite number of nodes, while the min and max lifts have exactly $|\mathcal{P}(S)| - 1 = 2^{|S|} - 1$ nodes. In order to prove Theorem 8.6, we thus need an auxiliary technical result which provides, given a path-complete graph $\mathcal{G}$, a switched system and a candidate Lyapunov function which satisfies a finite number of Lyapunov inequalities (encoded by the sum lift of $\mathcal{G}$) but violates several Lyapunov inequalities as well, as stated by the following lemma.

Lemma 8.7. *For any path-complete graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, there exist an integer $n \geq 1$, a switched system $F := \{f_i : i \in \langle M \rangle\}$ on M modes in dimension n and a candidate Lyapunov function $V_S := \{V_s : s \in S\}$ for which*

$$\forall (p, q, i) \in E, \; \forall x \in \mathbb{R}^n : \; V_q(f_i(x)) \leq V_p(x), \quad (8.12)$$

$$\forall T \in \mathbb{N}, \forall (P, Q, i) \in \overline{E^{\oplus T}}, \; \exists \tilde{x} \in \mathbb{R}^n : \sum_{q \in Q} V_q(f_i(\tilde{x})) > \sum_{p \in P} V_p(\tilde{x}), \quad (8.13)$$

where $\overline{E^{\oplus T}} = (S^{\oplus T} \times S^{\oplus T} \times \langle M \rangle) \setminus E^{\oplus T}$ refers to the set of edges of the complement graph of $\mathcal{G}^{\oplus T}$.

Note that this construction has been implemented in MATLAB and is available for consultation in [Deb23].

The proof of this lemma relies in particular on the characterization of an edge in the sum lift by graph theory presented in Section 7.3.3. Moreover, Hall's Marriage Theorem recalled hereafter, provides a necessary and sufficient condition for the existence of a perfect matching.

Proposition 8.8 (Hall's Marriage Theorem [Hal35]). *Let G be a finite bipartite graph with bipartite sets X and Y. There is an X-perfect matching, if and only if for every subset W of X,*

$$|W| \leq |N_G(W)|$$

where $N_G(W)$ denotes the neighborhood of W in G, i.e., the set of all vertices in Y adjacent to some element of W.

Proof of Lemma 8.7. Given the graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$, we define a set of M block-diagonal square $\{0,1\}$-matrices $\{A_i : i \in \langle M \rangle\}$ of dimension $n := 2 \times M \times (2^{|S|} - 1)$, where $2^{|S|} - 1$ is the cardinality of $\mathcal{P}_0(S)$. Each 2×2 diagonal block is associated to a pair $(W, i) \in \mathcal{P}_0(S) \times \langle M \rangle$, and is defined by

$$A_j[W,i] := \begin{cases} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} & \text{if } j = i, \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \text{otherwise} \end{cases} \tag{8.14}$$

so that each matrix only acts on the blocks associated to a pair of the same label. We consider the template of primal copositive norms where $V_s(x) := {v_s}^\top x$ for $s \in S$ and $x \in \mathbb{R}^n_{\geq 0}$, and for which the vectors $\{v_s : s \in S\}$ are defined below. In this context, satisfying the Lyapunov inequality associated to an edge $(p, q, i) \in S \times S \times \langle M \rangle$ amounts to satisfying a set of $2^{|S|} - 1$ scalar inequalities since

$$\begin{aligned} & \forall x \in \mathbb{R}^n_{\geq 0},\ V_q(f_i(x)) \leq V_p(x), \\ \Leftrightarrow \quad & \forall x \in \mathbb{R}^n_{\geq 0},\ x^\top {A_i}^\top v_q \leq x^\top v_p \\ \Leftrightarrow \quad & {A_i}^\top v_q \leq_c v_p, \\ \Leftrightarrow \quad & \forall W \in \mathcal{P}_0(S),\ v_q[W,i]_2 \leq v_p[W,i]_1, \end{aligned} \tag{8.15}$$

where $\leq_c$ depicts the componentwise inequality. We define the blocks $v_s[W,i]$ for $s \in S$ by

$$v_s[W,i]_1 := \begin{cases} 2 & \text{if } s \in PRE(W,i), \\ 1 & \text{otherwise,} \end{cases} \tag{8.16}$$

$$\text{and } v_s[W,i]_2 := \begin{cases} 2 & \text{if } s \in W, \\ 1 & \text{otherwise,} \end{cases} \tag{8.17}$$

where $PRE(W,i) = \{s \in S \mid \exists w \in W : (s,w,i) \in E\}$. We show now that this construction satisfies the expressions (8.12) and (8.13).

Let us start with the expression (8.12). Consider an edge $(a,b,i) \in E$. The condition (8.15) holds except if there exists a pair (W,i) such that $v_b[W,i]_2 = 2$ and $v_p[W,i]_2 = 1$. This happens if and only if $b \in W$ and $a \notin PRE(W,i)$. However $(a,b,i) \in E$ by assumption, which proves the contradiction. In all the other configurations, the inequality holds.

Let us now consider the expression (8.13), $T \in \mathbb{N}$ and $(P,Q,i) \in \overline{E^{\oplus T}}$. By the Hall's Marriage Theorem in Proposition 8.8, there exists a multi-set $W \subseteq Q$ such that $|W| > |N_{P,i}(W)|$, where the multi-set $N_{P,i}(W) := \{p \in P \mid \exists s \in W : (p,s,i) \in E\}$. Therefore, if we denote $\widetilde{W}$ as the underlying set of W formed from its distinct elements, we have

$$\sum_{q\in Q} v_q[\widetilde{W},i]_2 := \underbrace{\sum_{q\in W} v_q[\widetilde{W},i]_2}_{:=2|W|} + \underbrace{\sum_{q\in Q\setminus W} v_q[\widetilde{W},i]_2}_{:=T-|W|} = T + |W|$$

since for all $q \in W$, $q \in \widetilde{W}$ and $v_q[\widetilde{W},i]_2 = 2$. Similarly,

$$\begin{aligned} \sum_{p\in P} v_p[\widetilde{W},i]_1 &:= \underbrace{\sum_{p\in N_{P,i}(W)} v_p[\widetilde{W},i]_1}_{:=2|N_{P,i}(W)|} + \underbrace{\sum_{p\in P\setminus N_{P,i}(W)} v_p[\widetilde{W},i]_1}_{:=T-|N_{P,i}(W)|} \\ &= T + |N_{P,i}(W)|. \end{aligned}$$

Since $|W| > |N_{P,i}(W)|$, the inequality encoded by (P,Q,i) is violated. □

Similarly to Lemma 8.2 for the min lift, Lemma 8.7 is a stronger version of Lemma 6.31 because the non-edges of the initial graph are non-edges of any T-sum lifted graph. Then, Lemma 8.7 automatically implies Lemma 6.31.

Let us take an example to see how this procedure works.

Example 8.9. Consider the path-complete graph $\mathcal{G}_1 = (S_1, E_1)$ in Figure 7.14a on the alphabet $\langle 2 \rangle$. We follow the procedure described in the proof of Lemma 8.7 to obtain the matrices $\{A_1, A_2\} \subseteq \{0,1\}^{12\times 12}$ and the vectors v_{a_1} and $v_{b_1} \in \{1,2\}^{12}$ illustrated in Figure 8.2.

As expected, each 2-dimensional block of the vectors and matrices is associated to a couple $(W, i) \in \mathcal{P}_0(S_1) \times \langle M \rangle$. Then we can use the characterization in (8.15) to verify that the Lyapunov inequalities encoded by $\mathcal{G}_1$ are satisfied, as illustrated in Table 8.2. Moreover, we have manually checked whether the Lyapunov inequalites encoded by non-edges of $\mathcal{G}_1{}^{\oplus 2}$ were indeed violated. The result can be seen in Table 8.2. Note that we have implemented a code in MATLAB to build the outcome and automatically check the statement of Lemma 8.7 (in particular the expression (8.13) for any given value $T \in \mathbb{N}$). Using this code, we managed to check expression (8.13) up to $T = 60$. △

We need an additional result, stated in the following Lemma 8.10. The proof is provided below and follows the notation introduced in the proof of Lemma 8.7.

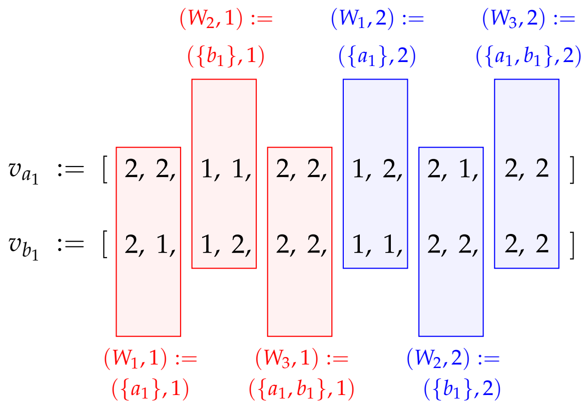


Fig. 8.2 Illustration of the procedure described in the proof of Lemma 8.7 to build the Lyapunov functions for the path-complete graph $\mathcal{G}_1 := (S_1, E_1)$ in Figure 7.14a. Each 2-dimensional block is associated to one of couples $(W, i) \in \mathcal{P}_0(S) \times \langle M \rangle$ of label 1 in red and of label 2 in blue.

Edge (P,Q,i)	$\sum_{q\in Q} v_q[W,i]_2 \overset{?}{\leq} \sum_{p\in P} v_p[W,i]_1$ $(W_1,1)$	$(W_2,1)$	$(W_3,1)$	$(W_1,2)$	$(W_2,2)$	$(W_3,2)$
$(a_1,a_1,1)\in E_1$	$2\leq 2$	$1\leq 1$	$2\leq 2$			
$(a_1,b_1,2)\in E_1$				$1\leq 1$	$2\leq 2$	$2\leq 2$
$(b_1,b_1,2)\in E_1$				$1\leq 1$	$2\leq 2$	$2\leq 2$
$(b_1,a_1,1)\in E_1$	$2\leq 2$	$1\leq 1$	$2\leq 2$			
$(\{a_1,a_1\},\{a_1,a_1\},2)$				$\mathbf{4>2}$	$2\leq 4$	$4\leq 4$
$(\{b_1,b_1\},\{a_1,a_1\},2)$				$\mathbf{4>2}$	$2\leq 4$	$4\leq 4$
$(\{a_1,a_1\},\{b_1,b_1\},1)$	$2\leq 4$	$\mathbf{4>2}$	$4\leq 4$			
$(\{b_1,b_1\},\{b_1,b_1\},1)$	$2\leq 4$	$\mathbf{4>2}$	$4\leq 4$			
$(\{a_1,a_1\},\{a_1,b_1\},1)$	$3\leq 4$	$\mathbf{3>2}$	$4\leq 4$			
$(\{a_1,a_1\},\{a_1,b_1\},2)$				$\mathbf{3>2}$	$3\leq 4$	$4\leq 4$
$(\{b_1,b_1\},\{a_1,b_1\},1)$	$3\leq 4$	$\mathbf{3>2}$	$4\leq 4$			
$(\{b_1,b_1\},\{a_1,b_1\},2)$				$\mathbf{3>2}$	$3\leq 4$	$4\leq 4$
$(\{a_1,b_1\},\{a_1,a_1\},2)$				$\mathbf{4>2}$	$2\leq 4$	$4\leq 4$
$(\{a_1,b_1\},\{b_1,b_1\},1)$	$2\leq 4$	$\mathbf{4>2}$	$4\leq 4$			
$(\{a_1,b_1\},\{a_1,b_1\},1)$	$3\leq 4$	$\mathbf{3>2}$	$4\leq 4$			
$(\{a_1,b_1\},\{a_1,b_1\},2)$				$\mathbf{3>2}$	$3\leq 4$	$4\leq 4$

Table 8.2 Illustration of the verification of the Lyapunov inequalities using the outcome of the proof of Lemma 8.7 for the graph $\mathcal{G}_1$ in Figure 7.15. As expected, the solution in Figure 8.2 satisfies the Lyapunov inequalities encoded by the graph but especially violates all the non-edges of the 2-sum lift.

Lemma 8.10. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the alphabet $\langle M\rangle$. Suppose also that the switched system F and the candidate Lyapunov function $V_S := \{V_s : s \in S\}$ are constructed in the proof of Lemma 8.7 applied to $\mathcal{G}$. The following statement holds:*

$$\forall W_{\widetilde{S}} \in \left(V_S{}^{\oplus}\right)^{\widetilde{S}} \text{ s.t. } W_{\widetilde{S}} \in PCLF(\widetilde{\mathcal{G}}, F),\ \exists T \in \mathbb{N} : W_{\widetilde{S}} \in \left(V_S{}^{\oplus T}\right)^{\widetilde{S}}, \quad (8.18)$$

where $V_S{}^{\oplus}$ and $V_S{}^{\oplus T}$ refer to the addition closure and the T-addition closure of V_S respectively.

Proof. Without loss of generality, we assume that $\widetilde{\mathcal{G}}$ is strongly connected. Otherwise, the argument is valid for each strongly connected component of the graph. More precisely, each strongly connected component $\widetilde{\mathcal{H}} \subseteq \widetilde{\mathcal{G}}$ can be associated to a value of $T(\widetilde{\mathcal{H}})$ for which condition (8.18) holds. Moreover, any integer multiple of $T(\widetilde{\mathcal{H}})$ also satisfies the statement (8.18) for $\widetilde{\mathcal{H}}$. Thus, taking the least common multiple of the $T(\widetilde{\mathcal{H}})$, for all the strongly connected components $\widetilde{\mathcal{H}}$ of $\widetilde{\mathcal{G}}$, we can conclude.

Consider a path-complete graph $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on $\langle M \rangle$ and $W_{\widetilde{S}} \in PCLF(\widetilde{\mathcal{G}}, F)$ an admissible solution in the template $\mathcal{V}$ defined as the sum-closure of $\{V_s : s \in S\}$, i.e.

$$\mathcal{V} := \{V_{i_1} + \cdots + V_{i_T} \mid (i_1, \ldots, i_T) \in S^{\oplus T}, T \in \mathbb{N}\}.$$

Each node $\tilde{s}$ of $\widetilde{S}$ is associated to a multi-set of S denoted by $R(\tilde{s})$. Let us prove that for all $\tilde{s}_1, \tilde{s}_2 \in \widetilde{S}$, $|R(\tilde{s}_1)| = |R(\tilde{s}_2)|$. Assume by contradiction that there exists $i \in \langle M \rangle$, $\tilde{s}_1$ and $\tilde{s}_2 \in \widetilde{S}$ such that $(\tilde{s}_1, \tilde{s}_2, i) \in \widetilde{E}$ and $|R(\tilde{s}_2)| \neq |R(\tilde{s}_1)|$, and $|R(\tilde{s}_2)| > |R(\tilde{s}_1)|$ without loss of generality. Since $(\tilde{s}_1, \tilde{s}_2, i) \in \widetilde{E}$, it means that

$$\forall x \in \mathbb{R}^n_{\geq 0}, \sum_{q \in R(\tilde{s}_2)} V_q(A_i x) \leq \sum_{p \in R(\tilde{s}_1)} V_p(x), \tag{8.19}$$

$$\Leftrightarrow \forall (W, j) \in \mathcal{P}_0(S) \times \langle M \rangle, \sum_{q \in R(\tilde{s}_2)} v_q[W, j]_2 \leq \sum_{p \in R(\tilde{s}_1)} v_p[W, j]_1. \tag{8.20}$$

Consider W as the underlying set of $R(\tilde{s}_2)$ and $j = i$. Then by construction, $v_q[W, i]_2 = 2$ for all $q \in R(\tilde{s}_2)$. Therefore the sum over $R(\tilde{s}_2)$ is equal to $2|R(\tilde{s}_2)|$. By expression (8.20), we have that

$$\sum_{q \in R(\tilde{s}_2)} v_q[W, i]_2 = 2|R(\tilde{s}_2)| \leq 2|R(\tilde{s}_1)|$$

since for any $p \in S_1$, $v_p[W, j]_1 \leq 2$. However, we know by assumption that $|R(\tilde{s}_2)| > |R(\tilde{s}_1)|$ which contradicts the previous expression. It means that $|R(\tilde{s}_2)| = |R(\tilde{s}_1)|$. Since the graph $\widetilde{\mathcal{G}}$ is strongly connected, the inequality holds between all the nodes. □

8.2.2 Main characterization theorem

We can now prove Theorem 8.6.

Proof. (1) $\Rightarrow$ (2) : It is already proved in Theorem 7.18 .

(2) $\Rightarrow$ (1) : Consider two graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ such that statement (2) in Theorem 8.6 is satisfied. Let us first apply Lemma 8.7 to $\mathcal{G}$. It provides a switched system $F := \{f_i : i \in \langle M \rangle\}$ and a candidate Lyapunov function $V_S := \{V_s : s \in S\}$ such that expressions (8.12) and (8.13) hold. Let us define the template $\mathcal{V}$ as the addition-closure of V_S. By construction, $V_S \subseteq \mathcal{V}$. This implies by hypothesis that there exists $W_{\widetilde{S}}$ in $\mathcal{V}^{\widetilde{S}}$ such that $W_{\widetilde{S}} \in PCLF(\widetilde{\mathcal{G}}, F)$. Since $W_{\widetilde{S}} \subseteq \mathcal{V}$, we can associate a multi-set of S to each node of $\widetilde{\mathcal{G}}$, i.e. we can define a function $R : \widetilde{S} \to S^{\oplus}$ such that

$$W_{\tilde{s}} \ := \sum_{p \in R(\tilde{s})} V_p.$$

By Lemma 8.10, there exists $T \in \mathbb{N}$ such that $R : \widetilde{S} \to S^{\oplus T}$. Suppose by contradiction that the function R is not a simulation relation. This means that there exists an edge $(p, q, i) \in \widetilde{E}$ such that $(R(p), R(q), i) \notin E^{\oplus T}$. By the expression (8.13) in Lemma 8.7, there exists $\tilde{x} \in \mathbb{R}^n$ such that

$$W_q(\tilde{x}) := \sum_{s \in R(q)} V_s(f_i(\tilde{x})) \ > \sum_{d \in R(p)} V_d(\tilde{x}) := W_p(\tilde{x}),$$

i.e. the indexed set $\{W_s : s \in \widetilde{S}\}$ is not admissible. However these inequalities are satisfied by construction of the functions $W_{\tilde{s}}$, here is the contradiction. □

Combining the results in Theorems 7.35 and 8.6, we can finally derive the following corollary that summarizes the algorithmic contribution of our work.

Corollary 8.11. *Given two graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ on the same alphabet, the following statements are equivalent:*

(1) $\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ for any template $\mathcal{V}$ closed under addition.

(2) $\mathcal{G} \leq_{\Sigma} \widetilde{\mathcal{G}}$.

Therefore, since Item (2) can be verified by solving a linear program, so is Item (1).

Example 8.12. Consider now the graph $\mathcal{G}_2 = (S_2, E_2)$ in Figure 8.3. We have already shown in Example 7.17 that 8-sum lift of $\mathcal{G}_2$ simulates $\mathcal{G}_0 = (\{s_0\}, \{(s_0, S_0, 1), (s_0, S_0, 2)\})$, the common Lyapunov function graph with

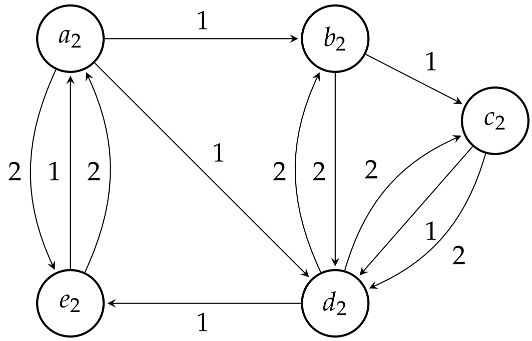


Fig. 8.3 $\mathcal{G}_2 = (S_2, E_2)$, a path-complete graph on $\langle 2 \rangle$ in Example 8.12. It has already been proved in Example 7.17 that ${\mathcal{G}_2}^{\oplus 8}$ simulates $\mathcal{G}_0$. Moreover, we show in Example 8.12 that $\mathcal{G}_2 \leq_\Sigma \mathcal{G}_0$.

2 modes. Indeed, given a switched system F, if the indexed set $V_{S_2} := \{V_s : s \in S_2\}$ is admissible for $\mathcal{G}_2$ and F, the function

$$W(\cdot) \ := \ 2V_{a_2}(\cdot) + V_{b_2}(\cdot) + V_{c_2}(\cdot) + 2V_{d_2}(\cdot) + 2V_{e_2}(\cdot)$$

is a common Lyapunov function for F. By Lemma 7.33 and using the same construction as the previous example, one can derive the following matrix

$$C^2_{sim} \ := \ \begin{bmatrix} 2 & 1 & 1 & 2 & 2 \end{bmatrix},$$

through which the graphs $\mathcal{G}_2$ and $\mathcal{G}_0$ satisfy Definition 7.30. Moreover, if we use the LP criterion provided in [PAAJ19], we find (after rounding) the matrix $C^2_{LP} = \frac{1}{4} \times C^2_{sim}$, which provides the same simulation relation. $\triangle$

The sum lift exploits the idea that the solution to a graph $\widetilde{\mathcal{G}}$ can be expressed as a linear combination with integer coefficients of the solution of a graph $\mathcal{G}$. So far, in all the examples that we have provided, this linear combination only involves coefficients equal to 1. However, we have deliberately chosen to define the T-sum lift with multi-sets, meaning that we allow linear combination with coefficients strictly greater than 1. Therefore, the following question arises: can we restrict the definition of the T-sum lift to linear combinations with coefficients equal to 1, i.e. can we restrict $S^\oplus$ to the power set? Otherwise, do we have examples of comparison of graphs with addition-closed templates where we need a linear combination with integer coefficients strictly greater than 1?

Question 8.13. Are there two graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ path-complete on the same alphabet $\langle M \rangle$ such that

$$\mathcal{G} \ \leq_{\mathcal{V}} \ \widetilde{\mathcal{G}}$$

for any template $\mathcal{V}$ closed under addition, but

$$\mathcal{G}^{\oplus}|_{S^{\oplus}:=P_0(S)} \ \not\leq \ \widetilde{\mathcal{G}}$$

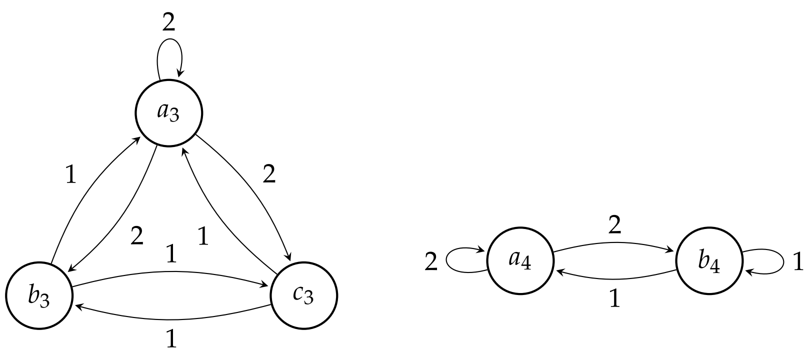
where $\mathcal{G}^{\oplus}|_{S^{\oplus}:=P_0(S)}$ denotes the restriction of $\mathcal{G}^{\oplus}$ to the nodes associated to non-empty subsets of S.

The following example answers in the affirmative.

Example 8.14. Consider the graph $\mathcal{G}_3 = (S_3, E_3)$ and its 2-sum lift in Figures 8.4a and 8.5 respectively. As expected, the lifted graph involves 6 nodes, one for each multi-set of S_3 of cardinality 2. We can easily observe that the 2-sum lift of $\mathcal{G}_3$ comprises two strongly connected and path-complete components; first, a duplicate of the initial graph, that is the component associated to the 2-multiples of the nodes of $\mathcal{G}_3$, i.e. $\{a_1, a_1\}$, $\{b_1, b_1\}$ and $\{c_1, c_1\}$. Secondly, it includes a strongly connected and path-complete component (in bold) isomorphic to the graph $\mathcal{G}_4 = (S_4, E_4)$ in Figure 8.4b. This implies that

$$\mathcal{G}_3 \ \leq_{\mathcal{V}} \ \mathcal{G}_4,$$

for any template $\mathcal{V}$ closed under addition, since $\mathcal{G}_3 \leq_{\mathcal{V}} {\mathcal{G}_3}^{\oplus 2}$, and the rela-



(a) $\mathcal{G}_3 = (S_3, E_3)$, a path-complete graph over the alphabet $\langle 2 \rangle$.

(b) $\mathcal{G}_4 = (S_4, E_4)$, a strongly connected and path-complete component of ${\mathcal{G}_3}^{\oplus 2}$.

Fig. 8.4 Counter-example to Question 8.13. As discussed in Example 8.14, $\mathcal{G}_4$ is a component of $\mathcal{G}_3^{\oplus 2}$ but not of its restriction to the subsets of the nodes of $\mathcal{G}_3$.

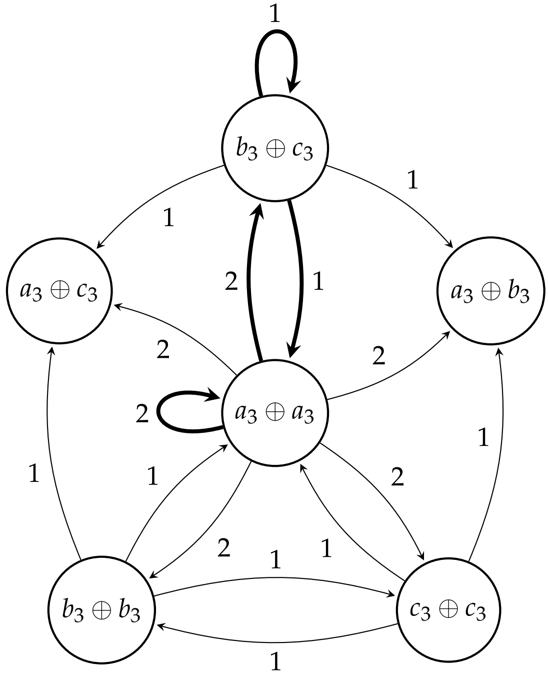


Fig. 8.5 $\mathcal{G}_3{}^{\oplus 2} = (S_3{}^{\oplus 2}, E_3{}^{\oplus 2})$, the 2-sum lift of $\mathcal{G}_3$. We can observe that $\mathcal{G}_4$ in Figure 8.4b is a component (in bold) of $\mathcal{G}_3{}^{\oplus 2}$ which proves that $\mathcal{G}_2 \leq_{\mathcal{V}} \mathcal{G}4$ for any template $\mathcal{V}$ closed under addition.

tion holds for any component of the 2-sum lift. This example demonstrates that the restriction of the nodes of the sum lift to the non-empty subsets of S_3 (rather than multisets) cannot explain the ordering relation between the graphs $\mathcal{G}_3$ and $\mathcal{G}_4$. △

8.2.3 Numerical experiment

In this section, we consider the path-complete graph $\mathcal{G}_5 = (S_5, E_5)$ already studied in Example 7.27, recalled in Figure 8.6, and the common Lyapunov function graph $\mathcal{G}_0 = (\{s_0\}, \{(s_0, s_0, 1), (s_0, s_0, 2)\})$ on two modes. The complete code for this section can be found in [Deb23].

First, let us recall that by Proposition 7.52 and Theorems 8.1 and 8.5, one can show that the graphs $\mathcal{G}_{5\min}$ and $\mathcal{G}_{5\max}$ both simulate $\mathcal{G}_0$. In this sense, $\mathcal{G}_5$ seems to be a very inefficient graph. Indeed, despite the fact that it defines a multiple Lyapunov function criterion with five node-functions,

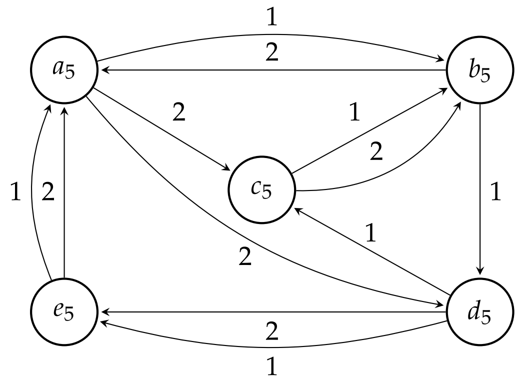


Fig. 8.6 $\mathcal{G}_5 = (S_5, E_5)$, a path-complete graph on $\langle 2 \rangle$ used for the numerical example in Section 8.2.3. It can be shown that $\mathcal{G}_{5\min} \leq \mathcal{G}_0$ and $\mathcal{G}_{5\max} \leq \mathcal{G}_0$. However, there does not exist any value $T \in \mathbb{N}$ such that $\mathcal{G}_5{}^{\oplus}$ simulates $\mathcal{G}_0$.

it is as conservative as $\mathcal{G}_0$ for the class of templates closed under minimum or maximum. Thus, one might wonder whether it is more efficient than $\mathcal{G}_0$ with a template such as the quadratic Lyapunov functions, which is not closed under minimum nor maximum while being closed under addition.

We have already shown in Example 7.27 that there does not exist any value of $T \in \mathbb{N}$ such that $\mathcal{G}_5{}^{\oplus T}$ simulates $\mathcal{G}_0$. The LP characterization in Theorem 7.35 confirms this result. By Theorem 8.6 and by Definition 6.21, this means that there exists at least a template $\mathcal{V}$ closed under addition for which

$$\begin{aligned} \exists n \in \mathbb{N}, \exists F := \{f_1, f_2\} \subset \mathcal{C}^0(\mathbb{R}^n, \mathbb{R}^n) : \Big[\exists V_{S_5} \in \mathcal{V}^{|S_5|} \text{ s.t.} \\ V_{S_5} \in PCLF(\mathcal{G}_5, F) \Big] \wedge \Big[\forall W_{S_0} \in \mathcal{V}^{|S_0|}, W_{S_0} \notin PCLF(\mathcal{G}_0, F) \Big]. \end{aligned} \tag{8.21}$$

In order to numerically verify this statement, we consider the template of quadratic functions $\mathcal{Q}$ and we sample randomly 10 000 2×2 linear switched systems with 2 modes of the form

$$x(k+1) \; = \; A_{\sigma(k)} x(k),$$

where $x(k) \in \mathbb{R}^2$ for any $k \in \mathbb{N}$, and $\mathcal{A} = \{A_1, A_2\} \subset \mathbb{R}^{2\times 2}_{\geq 0}$. For each system $\mathcal{A}$ and for any path-complete graph $\mathcal{G} = (S, E)$, we can compute the JSR approximation provided by $\mathcal{G}$ and the template $\mathcal{V}$ in Definition 6.17, denoted by $\rho_{\mathcal{G},\mathcal{V}}(\mathcal{A})$. Therefore, if a switched system $\mathcal{A}$ satisfies statement (8.21) for the quadratic template, $\mathcal{G}_5$ will provide a strictly smaller upper bound than $\mathcal{G}_0$, i.e. $\rho_{\mathcal{G}_5,\mathcal{Q}}(\mathcal{A}) < \rho_{\mathcal{G}_0,\mathcal{Q}}(\mathcal{A})$.

For each system $\mathcal{A}_\ell$, we compute $\rho_{\mathcal{G}_0,\mathcal{Q}}(\mathcal{A}_\ell)$ and $\rho_{\mathcal{G}_5,\mathcal{Q}}(\mathcal{A}_\ell)$ respectively for $\ell = 1, \dots, 10\,000$, and we compare them by defining the following index:

$$I(\ell) := \log\left(\frac{\rho_{\mathcal{G}_0,\mathcal{Q}}(\mathcal{A}_\ell)}{\rho_{\mathcal{G}_5,\mathcal{Q}}(\mathcal{A}_\ell)}\right). \tag{8.22}$$

The distribution of this index is illustrated in Figure 8.7. As expected, the index I is non-negative for all the switched systems since it can be proven that $\mathcal{G}_0 \leq_{\mathcal{Q}} \mathcal{G}_5$, see Proposition 6.24. On the other hand, the sampled systems also provide instances for proving the non-relation $\mathcal{G}_5 \not\leq_{\mathcal{Q}} \mathcal{G}_0$, i.e. satisfying (8.21). More specifically, the difference between the output of $\mathcal{G}_5$ and $\mathcal{G}_0$ is significant (meaning that $I(l) \geq 10^{-6}$) for 1668 systems out of 10 000, and thus these systems satisfy (8.21).

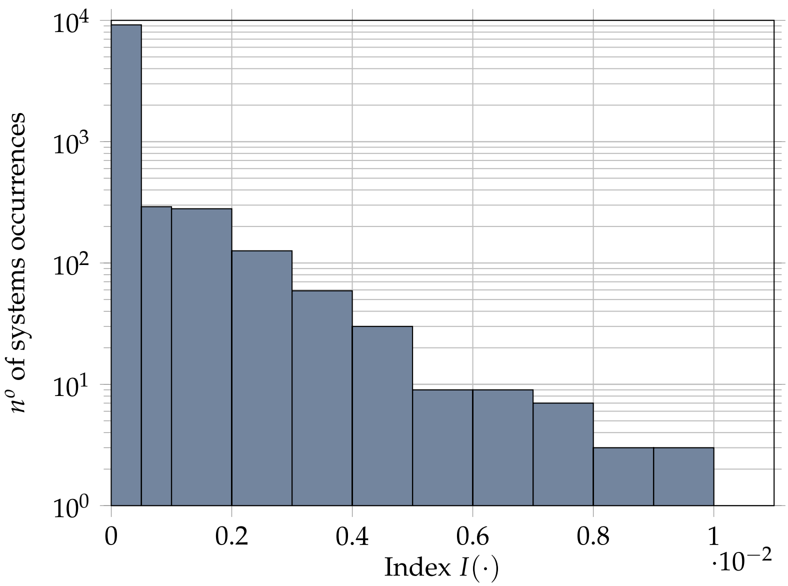


Fig. 8.7 Histogram of the index $I(\cdot)$ in Equation (8.22) for 10 000 linear switched systems with 2 matrices of dimension 2. Note that in the first column of the histogram, among the 9182 systems composing it, for 8332 of them, $I(i)$ is 0: for these systems, $\mathcal{G}_0$ and $\mathcal{G}_5$ provide exactly the same estimation of the JSR.

On the other hand, for 8332 out of the 10 000 sampled systems, $\mathcal{G}_5$ and $\mathcal{G}_0$ provide exactly the same approximation[1] of the JSR, i.e. the graph $\mathcal{G}_5$ provides the same JSR estimation as the classical common quadratic approach, which can be considered as the "most conservative" one. This numerical result is consistent with the theoretical results since we know that both graphs $\mathcal{G}_5$ and $\mathcal{G}_0$ provide the same approximation of the JSR if we consider a template closed under minimum or maximum. We highlight that the goal of this example was to validate numerically the "conservatism relations" between $\mathcal{G}_0$ and $\mathcal{G}_5$ (proven in Theorem 8.6), and *not* to provide numerically appealing approximation of the JSR: in this case the hierarchy of De Bruijn path-complete graphs (described in Section 7.7) is more relevant.

To further analyse the relations between path-complete graphs, we focus on the switched system for which the index I is maximal, given by the matrices

$$A_1 = \begin{bmatrix} 1.5519 & 0.4474 \\ 7.6412 & 7.4716 \end{bmatrix} \text{ and } A_2 = \begin{bmatrix} 0.4750 & 9.1755 \\ 1.8955 & 0.1850 \end{bmatrix}. \tag{8.23}$$

The graph-based results are reflected in the approximations of the JSR of system (8.23) in Table 8.3 provided by $\mathcal{G}_0$ and $\mathcal{G}_5$ and two different templates: first the template of linear primal copositive norms $\mathcal{P}$ which is closed under minimum and addition (see Theorem 7.43 for the details) and the template of quadratic functions $\mathcal{Q}$ which is closed under addition. As expected, both $\mathcal{G}_0$ and $\mathcal{G}_5$ provide the same approximation of the JSR with the copositive norms, i.e.

$$\rho_{\mathcal{G}_0,\mathcal{L}}(\mathcal{A}) \;=\; \rho_{\mathcal{G}_5,\mathcal{L}}(\mathcal{A}) \;:=\; 9.2696,$$

while $\mathcal{G}_5$ provides a better approximation than $\mathcal{G}_0$ when we use the quadratic template, i.e.

$$\rho_{\mathcal{G}_5,\mathcal{Q}}(\mathcal{A}) \;:=\; 9.4886 \;<\; 9.5868 \;:=\; \rho_{\mathcal{G}_0,\mathcal{Q}}(\mathcal{A}).$$

Note also that for both graphs, the copositive norms provide a better approximation than the quadratic ones whereas they are easier to solve numerically.

[1]We assume that all the values of I smaller than 10^{-6} are due to numerical errors, and are then set at 0.

Template	$\rho_{\mathcal{G},\mathcal{V}}(\mathcal{A})$				
	$\mathcal{G}_0$	$\mathcal{G}_1$	$\mathcal{G}_{db}^{3,1}$	$\mathcal{G}_2$	$\mathcal{G}_5$
$\mathcal{P}$	9.2696	9.1712	8.7019	9.2696	9.2696
$\mathcal{Q}$	9.5868	9.0161	8.6881	9.5868	9.4886

Table 8.3 Graph-based approximations of the JSR of system (8.23) with $\mathcal{G}_0$, $\mathcal{G}_1$, $\mathcal{G}_{db}^{3,1}$, $\mathcal{G}_2$ and $\mathcal{G}_5$ for the templates $\mathcal{Q}$ of quadratic functions and $\mathcal{P}$, the primal copositive norms.

We have also computed the approximations provided by the graphs $\mathcal{G}_1$, $\mathcal{G}_{db}^{3,1}$ and $\mathcal{G}_3$ in Figures 7.13 and 8.3 for system (8.23). The results provided in Table 8.3 highlight the comparison relations that we have proved in previous examples: $\mathcal{G}_2$ provides a better approximation than $\mathcal{G}_1$ for both templates while $\mathcal{G}_3$ is as bad as the common Lyapunov function graph $\mathcal{G}_0$.

8.3 Preparatory work for a general necessary condition

So far, we have succeeded in proving the characterization Theorems 8.1, 8.5 and 8.6 by providing ad hoc auxiliary results, namely Lemmas 8.2, 8.4 and 8.7. In this section, we identify the common structure of these proofs and we compare it with previous characterization results. We believe that it is possible to prove the characterization by the simulation using less demanding results, and in particular by requiring a common property to the template-dependent lifts. In this section, we summarize preliminary work on this topic and we present our main conjecture.

We start by emphasizing that all the proofs of the characterization theorems follow the same structure. In particular, we compare the constructions in Lemmas 6.31, 8.2, 8.4 and 8.7 and Theorem 6.15. By definition of the path-complete formalism, any labeled and directed graph encodes Lyapunov inequalities on both a switched system and a set of candidate Lyapunov functions. Then for each graph, there exist several pairs $(F := \{f_1, \dots, f_M\}, V := \{V_s : s \in S\})$ that satisfy the corresponding graph-based Lyapunov inequalities. Among them, it is possible to draw specific pairs (V, F) which are *extremal* in the sense that they allow to derive properties on the graph, and as a consequence on all the other pairs (F, V). For each different property, we require a different *extremal constraint*. Table 8.4 summarizes this comparison.

	Characterization of graphs	Characterization of the (template-dependent) ordering			
	Theorem 6.15	Theorem 6.30	Theorem 8.1	Theorem 8.5	Theorem 8.6
Hypothesis:					
	$\mathcal{G}$ is *not* path-complete $\Leftrightarrow\ \exists w \in \langle M \rangle^k : w \notin \mathscr{P}_k(\mathcal{G})$	$\mathcal{G} \leq \widetilde{\mathcal{G}}$	$\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ s.t. $\min(\mathcal{V}) \subseteq \mathcal{V}$	$\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ s.t. $\max(\mathcal{V}) \subseteq \mathcal{V}$	$\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ s.t. $\sum(\mathcal{V}) \subseteq \mathcal{V}$
Step 1: Given any graph $\mathcal{G} = (S, E)$, we build an *extremal* couple switched system - solution (F, V) such that					
(a)	$V := \{V_s : s \in S\}$ satisfies the Lyapunov inequalities encoded by the graph $\mathcal{G} = (S, E)$ for the switched system $F := \{f_1, \dots, f_M\}$ $\Leftrightarrow\ \forall e = (s, d, i) \in E,\ \forall x \in \mathbb{R}^n : \ V_d(f_i(x)) \ \leq \ V_s(x)$				
(b)	$f_{w(1)} \circ \cdots \circ f_{w(k)}$ is unstable	The edges in $\overline{E}$ are violated	The edges in $\overline{E_{\min}}$ are violated	The edges in $\overline{E_{\max}}$ are violated	The edges in $\bigcup_{T \in \mathbb{N}} \overline{E^{\oplus T}}$ are violated
Step 2: We derive a general property on the graphs					
	$\mathcal{G}$ is not valid	$\mathcal{G}$ simulates $\widetilde{\mathcal{G}}$	$\mathcal{G}_{\min}$ simulates $\widetilde{\mathcal{G}}$	$\mathcal{G}_{\max}$ simulates $\widetilde{\mathcal{G}}$	$\mathcal{G}^{\oplus}$ simulates $\widetilde{\mathcal{G}}$

Table 8.4 Comparison of the proofs of different characterization results: the path-completeness in Theorem 6.15 and simulation-based characterization of the general ordering (6.12) in Theorem 6.30, the template-dependant ordering (6.11) for the class of templates closed under minimum, maximum and addition in Theorems 8.1, 8.5 and 8.6 respectively.

In particular, Theorem 6.15 states that the path-completeness is a necessary and sufficient condition for the corresponding Lyapunov certificate to be valid. The proof of the necessary condition required a special construction developed and fully detailed in [JAPR17]. In brief, the authors show that it is possible to derive an unstable system F and an index set of candidate Lyapunov functions $V := \{V_s : s \in S\}$ from a non-path-complete graph $\mathcal{G} = (S, E)$ such that the pair (F, V) satisfies the Lyapunov inequalities encoded by $\mathcal{G}$. In this case, this specific pair (F, V) is *extremal* with respect to the path-completeness property since it allows to characterize the validity of a certificate by the path-completeness. Similarly, Lemma 6.31 proves the existence of a pair (F, V) which is *extremal* with respect to the simulation and the general ordering in (6.12). Lemmas 8.2, 8.4 and 8.7 in turn show that there exists an *extremal* pair (F, V) for the simulation and the template-dependent ordering relation in (6.11) for the family of templates closed under minimum, maximum and addition respectively.

Although the proofs of the different characterization theorems follow the same structure, we believe that the corresponding *extremal constraint* that we required might be weakened but still strong enough to characterize the template-dependent ordering relations (6.11) for specific classes of templates. In what follows, we describe preliminary thoughts on the appropriate extremal property.

By definition of the PCLF formalism, a path-complete graph $\mathcal{G} = (S, E)$ encodes a set of Lyapunov inequalities between some Lyapunov functions $\{V_s : s \in S\}$ that we are looking for in a given template $\mathcal{V}$. If $\star$ denotes a closure property of this template, we know that the template remains stable through $\star$, i.e.

$$\forall k \in \mathbb{N},\ \forall g_1, \ldots, g_k \in \mathcal{V} :\ g_1 \star \ldots \star g_k \in \mathcal{V}.$$

We would like to derive a lifted graph of the initial graph, denoted by $L_\star(\mathcal{G})$, which fully exploits this closure property $\star$, so that we can compare them in terms of the ordering relations in Definition 6.21. In other words, we aim to build $L_\star(\mathcal{G})$ such that it exclusively encodes the Lyapunov inequalities which are satisfied for any switched system F by the composition with $\star$ of the Lyapunov functions in $V := \{V_s : s \in S\}$ provided that V is admissible for $\mathcal{G}$ and F.

We propose the following definition.

Definition 8.15 (Lift which symbolizes an operation)**.** Consider a binary operation $\star$ which preserves the Lyapunov properties and $M \in \mathbb{N}$. The lift *symbolizes the operation* $\star$, denoted by $L_\star : Graphs_M \to Graphs_M$, if for any path-complete graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$,

$$L_\star(S) := \bigcup_{T \in \mathbb{N}} Multi^T(S)$$

and an edge $(\{s_1, \dots, s_k\}, \{d_1, \dots, d_l\}, i) \in L_\star(E)$ *if and only if* for any switched system $F := \{f_1, \dots, f_M\}$ and any $V \in \mathcal{C}^0_+(\mathbb{R}^n, \mathbb{R})^S$:

$$\begin{gathered} V \in PCLF(\mathcal{G}, F) \;\Rightarrow\; \forall x \in \mathbb{R}^n : \\ \left(V_{d_1} \star \cdots \star V_{d_l}\right)(f_i(x)) \;\leq\; \left(V_{s_1} \star \cdots \star V_{s_k}\right)(x), \end{gathered} \tag{8.24}$$

where $i \in \langle M \rangle$, $k, l \in \mathbb{N}$ and $s_j, d_h \in S$ for $j = 1, \dots, k$ and $h = 1, \dots, l$.

First of all, let us show that the lift in Definition 8.15 is well-defined, meaning that it satisfies Definition 7.2 and let us discuss its validity.

Proposition 8.16. *Consider $L_\star$ which symbolizes a binary operation $\star$ which preserves the Lyapunov properties. The lift $L_\star$ is valid with respect to any template closed under $\star$.*

Proof. Consider a path-complete graph $\mathcal{G} = (S, E)$ on the alphabet $\langle M \rangle$.

Let us first demonstrate that $L_\star$ is a lift in the sense of Definition 7.2, i.e. $L_\star(\mathcal{G})$ is path-complete. Definition 8.15 implies in particular that the initial graph $\mathcal{G}$ is a subcomponent of the lifted graph $L_\star(\mathcal{G})$. Indeed, all the singletons $\{s\} \in L_\star(S)$ for every $s \in S$ and all the edges of the form $(\{s\}, \{d\}, i) \in L_\star(E)$ where $(s, d, i) \in E$. Since $\mathcal{G}$ is path-complete by assumption, so is $L_\star(\mathcal{G})$.

Let us now discuss the validity of $L_\star$. Given a template $\mathcal{V}$ closed under the operation $\star$, we have to prove that $\mathcal{G} \leq_{\mathcal{V}} L_\star(\mathcal{G})$. Assume that $V \in \mathcal{V}^S$ is admissible for $\mathcal{G}$ and a switched system $F := \{f_1, \dots, f_M\}$. We define

$$W := \{W_{(s_1, \dots, s_k)} := V_{s_1} \star \cdots \star V_{s_k} : (s_1, \dots, s_k) \in L_\star(S)\}.$$

By definition of the edges of the lift $L_\star$ in Equation (8.24), the Lyapunov inequalities for the system F encoded by the lifted edges are trivially satisfied by W, which ends the proof. $\square$

Note that we could define the lift that symbolizes an operation $\star$ within a specific family of switched systems $\mathcal{F}$, denoted by $L_{\star,\mathcal{F}}$ by restricting Definition 8.15 to $\mathcal{F}$. Then, we could prove that $L_{\star,\mathcal{F}}$ is valid with respect to any template closed under $\star$ and $\mathcal{F}$.

Definition 8.15 suggests to use the following core methodology. At first, we associate a node of the lifted graph to any multi-set of finite cardinality of the universe S, such that it is associated to the $\star$-composition of the Lyapunov functions $\{V_s : s \in S\}$. Then, we hope that the lift is expressive enough to capture all the possible inequalities which are satisfied by construction.

In fact, Definition 8.15 seems relevant to characterize the lifts since Lemmas 8.2 and 8.4 allow us to show that the min and max lifts symbolize the pointwise minimum and maximum operations.

Proposition 8.17. *The min and max lifts symbolize the pointwise minimum and maximum operation respectively.*

Proof. Consider a path-complete graph $\mathcal{G}$ on the alphabet $\langle M \rangle$, with $M \in \mathbb{N}$. We have to prove that the following equivalence holds for any subsets $A, B \subseteq S$ and any label $i \in \langle M \rangle$:

$$(A, B, i) \in E_{\min} \Leftrightarrow \forall F \subset \mathcal{C}^0(\mathcal{R}^n, \mathcal{R}^n),$$
$$\left[V \in PCLF(\mathcal{G}, F) \Rightarrow \forall x \in \mathbb{R}^n : \min_{b \in B}\{V_b(f_i(x))\} \leq \min_{a \in A}\{V_a(x)\} \right]. \tag{8.25}$$

The necessary condition has already been shown in the proof of Proposition 7.40. The sufficient condition is a direct consequence of Lemma 8.2.

By contraposition, we have to prove that if an edge $e := (A, B, i) \notin E_{\min}$, then there exists a system F and an admissible solution $V \in PCLF(\mathcal{G}, F)$ which violates the Lyapunov inequality encoded by e. Lemma 8.2 guarantees the existence of a common system and a common solution which violates all the non-edges of the min lift, which ends the proof.

Similarly for the max lift, the necessary condition has already been shown in the proof of Proposition 7.41 while the sufficient condition is directly derived from Lemma 8.4 using the same argument. □

Similarly, Lemma 8.7 allows us to show that the sum lift symbolizes the addition in the sense of Definition 8.15.

Proposition 8.18. *The sum lift symbolizes the addition.*

Proof. Consider a path-complete graph $\mathcal{G}$ on the alphabet $\langle M \rangle$, with $M \in \mathbb{N}$. We want to prove that the sum lift satisfies Definition 8.15, i.e. for all multi-set $A, B \subseteq S$ of any cardinality $T \in \mathbb{N}$ and any label $i \in \langle M \rangle$:

$$\begin{gathered}(A, B, i) \in E^{\oplus T} \Leftrightarrow \forall F \subset \mathcal{C}^0(\mathcal{R}^n, \mathcal{R}^n), \\ \left[V \in PCLF(\mathcal{G}, F) \Rightarrow \forall x \in \mathbb{R}^n : \sum_{b \in B} V_b(f_i(x)) \leq \sum_{a \in A} V_a(x) \right]. \end{gathered} \tag{8.26}$$

The necessary condition has already been demonstrated in the proof of Proposition 7.11. The sufficient condition is a direct consequence of Lemma 8.7 and follows the same ideas as Lemma 8.2. □

We have shown that the min, max and sum lifts symbolize the corresponding binary operation of pointwise minimum, maximum and addition respectively. For each operation, the demonstration mainly relies on the auxiliary Lemmas 8.2, 8.4 and 8.7. These lemmas and Definition 8.15 differ as follows: Lemmas 8.2, 8.4 and 8.7 require the existence of a common switched system and a common admissible solution which violates all the non-existing edges of the min lift, while Definition 8.15 allows the system and solution to be different for each edge. Then, the notion in Definition 8.15 is weaker than the conditions required in Lemmas 8.2, 8.4 and 8.7. We think that these lemmas are actually too strong, and it is sufficient to require that the lift satisfies Definition 8.15 to prove the simulation-based characterisation. To go one step further, we conjecture that Definition 8.15 is a necessary and sufficient condition to derive the simulation-based characterization for any binary operation.

Conjecture 8.19. *Consider a binary operation $\star$ which preserves the Lyapunov properties and two path-complete graphs $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ on the alphabet $\langle M \rangle$. The following statements are equivalent:*

(a) $\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$ *for any template $\mathcal{V}$ closed under $\star$,*

(b) $L_\star(\mathcal{G})$ *simulates* $\widetilde{\mathcal{G}}$.

So far, the existence of a switched system and a set of candidate Lyapunov functions which are both common to all the edges is crucial for the construction in the proof of Theorems 8.1, 8.5 and 8.6. Consequently, the main challenge in proving this Conjecture 8.19 is to succeed in building the simulation recursively on the edges, using similar arguments. In return, it would be easier (since less demanding) to prove that a lift satisfies Definition 8.15 than to prove the current corresponding key lemma.

8.4 Summary and further research directions

The path-complete Lyapunov framework generates a wide range of stability criteria for discrete-time switched systems by leveraging two degrees of freedom: the choice of the path-complete graph, and the template. In this section, we have provided a characterization of the template-dependent ordering of path-complete graphs for the specific classes of templates closed under pointwise minimum and maximum and addition by means of the combinatorial tool of simulation. In particular, given a switched system and a template, these results can help guiding the search of a better stability certificate by checking the existence of a simulation relation.

Summary of Chapter 8

In this chapter, we use combinatorial tools to provide a complete characterization of the conservatism-degree of stability conditions arising from graph-based structures. This characterization, already tackled in the past for the general case, is here proven for all the sets of candidate Lyapunov functions closed under minimum, maximum and addition.

Section 8.1: Min/max lifts and minimum/maximum-closed templates
We show in particular that we can leverage the duality between the min and max lifts to derive the characterization results from one lift to the other.

Section 8.2: Sum lift and addition-closed templates
The characterization results in this section provide a step forward for the analysis and taxonomy of multiple quadratic or SOS Lyapunov criteria, since both sets are closed under addition. Moreover, we provide a numerical appealing method (in the form of a linear program) to compare different path-complete criteria when it comes to templates closed under addition.

Section 8.3: Preparatory work for a general necessary condition
In this section, we provide an overview and compare the different proofs to identify their similarities and differences. In addition, we summarize preliminary research to provide a general proof of the characterization of template-dependent graph ordering.

The proof technique developed in this chapter seems to be generalizable (with appropriate minor modifications) and relevant for other settings. Given a closure operation on candidate Lyapunov functions, all the complexity lies in the proof of the corresponding key lemma. In particular, we expect this methodology to be suitable for the class of templates closed under composition introduced in Section 7.5. Moreover, the result provided in Theorem 7.65 suggests the following conjecture.

Conjecture 8.20. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet. The following statements are equivalent:*

(1) $\mathcal{G}^\circ$ simulates $\widetilde{\mathcal{G}}$.

(2) $\mathcal{G} \leq_{\mathcal{V},\mathcal{F}} \widetilde{\mathcal{G}}$ for any template $\mathcal{V}$ closed under composition with the dynamics in $\mathcal{F}$.

The same conjecture holds for the backward composition lift and the family of templates closed under composition with the inverse dynamics in $\mathcal{F}$.

Finally, the characterization results presented in this chapter provide a crucial step in the problem of classifying stability criteria based on multiple Lyapunov functions with respect to a family of templates which share a common closure property. However the complete characterization of the ordering for *a specific template* is still an open question. We believe that such a characterization should involve all the closure properties of this specific template.

Conjecture 8.21. *Consider two path-complete graphs $\mathcal{G} = (S, E)$ and $\widetilde{\mathcal{G}} = (\widetilde{S}, \widetilde{E})$ on the same alphabet, and a template $\mathcal{V}$ of candidate Lyapunov functions. The following statements are equivalent:*

(1) $\mathcal{G} \leq_{\mathcal{V}} \widetilde{\mathcal{G}}$.

(2) $\displaystyle\bigcup_{\star \in clos(\mathcal{V})} L_\star(\mathcal{G})$ simulates $\widetilde{\mathcal{G}}$,

where $clos(\mathcal{V})$ refers to the set of the closure properties of the template $\mathcal{V}$.

Note that Conjecture 8.21 differs from Conjecture 8.19 in that it characterizes the template-dependent ordering of graphs for a *specific* template rather than for a *family of templates* sharing a common closure property.

For the future, it could be interesting to proceed further in this analysis, with particular attention to the template of quadratic functions, which is probably the most common (for both theoretical and numerical reasons) candidate Lyapunov functions template in control theory.

Conclusions and perspectives

THROUGHOUT this thesis, we have studied the stability of switched systems under arbitrary switching. These are a family of popular models for complex Cyber-Physical systems, that is algorithmically challenging, but rather well understood from a theoretical standpoint. They form an entry point to more complex systems, such as hybrid systems, and we believe that the methodology that we have applied can be generalized to these systems.

In particular, we have developed different tools to provide a better understanding of and improve the conservatism of different stability certificates for switched systems by merging Lyapunov and graph theories. Moreover, we have theoretically leveraged the representation power of neural networks to approximate the joint spectral radius by learning polytopic Lyapunov functions, and we have provided an empirical study of their efficiency.

⋆ Brief summary of the contributions

In what follows, we briefly summarize our contributions throughout this thesis.

Part II: Neural Lyapunov functions

Motivated by recent developments in neural Lyapunov techniques, we have introduced for it a benchmark application and a theoretical framework, for the particular case of switched systems.

We have shown that one can determine theoretical bounds for the accuracy of neural Lyapunov functions, as a function of the parameters of the network. These guarantees are competitive with classical SDP-based Lyapunov approaches in terms of number of variables. From the empirical point of view, we have shown that in practice as well, the approach is competitive while our neural networks were trained on simple personal computers, which leaves an important room for improvement. We have emphasized the problem of overfitting, and proposed several avenues for mitigating it.

We have also introduced an automatic and sound algorithm to study the stability of switched systems by approximating the joint spectral radius of the corresponding set of matrices. Our architecture relies on two elements: a ReLU neural network, and an SMT solver. We therefore benefit from the advantages of these two components: notably the flexibility of neural networks and the soundness of SMT solvers. We also suffer from their disadvantages, like the poor scalability of SMT solvers. However, we introduce post processed norms to address this problem, and ensure a valid approximation of the JSR. Our algorithm has shown promising results on several examples, nearly always beating the usual quadratic approximation, but further comparison is required with more advanced methods, and in higher dimensions.

Part III: Template-dependent comparison of Path-Complete Lyapunov functions

The path-complete approach is an appealing tool for stability analysis of switching systems because it provides a way of building ad-hoc, non-standard, Lyapunov stability criteria while alleviating the combinatorial explosion of classical optimization techniques. In this thesis, we studied the problem of establishing relations among different path-complete structures, with the goal of optimizing this structure, while controlling the computational cost at the same time. We have demonstrated the strong connections between templates of candidate Lyapunov functions and the ordering relations between graph-based conditions.

We provided new results concerning the comparison problem for path-complete Lyapunov conditions. We introduced new formal transformations of path-complete graphs, called lifts, which allow us to establish ordering relations between graphs. We analyzed how the effectiveness of these lifts strongly depends on the closure properties of the chosen template. This allowed us to generalize previous results and to provide a unifying framework which enables for finer comparison criteria between path-complete techniques.

As particular case study, we thoroughly analyzed positive systems and copositive functions, for which we showed that our techniques did outperform the state of the art. The results have been validated with several examples. This work has opened the way for a larger research avenue, such as the generalization of this approach to a wider class of templates/systems (notably the quadratic functions cases).

Moreover, we have provided a complete characterization of the degree of conservatism of stability conditions arising from graph-based structures. While, in the past, this characterization had been tackled for the general case, in this work, we have proven a similar characterization for all the sets of candidate Lyapunov functions closed under a closure property. These results marked a significant step in the comparison problem stability criteria based on multiple quadratic Lyapunov functions, even if the complete characterization of the ordering for this specific template remains an open question.

⋆ Perspectives

This work has opened several paths for future research. At the end of each chapter, we have already sketched several perspectives for future work. In addition, we present hereunder a few general ideas for future research. At first, we list some research areas in which the path-complete Lyapunov could be used to generalize current results to switched systems. The second research direction aims to merge Parts II and III of this thesis in order to generate path-complete neural Lyapunov functions. Finally, we propose to exploit the similarities between template-dependent lifts and the abstraction-based techniques to help smartly refine the abstraction.

Future fields of application: over the past few years, the path-complete Lyapunov formalism initially introduced for the stability of discrete-time switched systems [AJPR14] has been generalized to various fields. Examples include a partial extension in the continuous-time setting [DPA22] and a general framework for the stability analysis of constrained switched systems [PEDJ15, Phi17]. Moreover, the path-complete Lyapunov methods have allowed to extend some concepts to switched systems, such as a generalization of the *p-dominance* for switched systems [BFJ18] and the notion of *path-complete barrier functions* [AJZA].

We believe that the path-complete Lyapunov approach could be further used to study switched systems. In particular, we think that the notion of recurrent sets [SBM22] could be extended to switched systems through the path-complete formalism, as well as the concept of dissipativity [BP07, WZG13]. In addition, the path-complete Lyapunov formalism could also be adapted to deal with different sorts of systems and problems: *state-dependent* switched systems, where each node of the graph would be associated to one region of the state-space partition for instance. The path-complete Lyapunov formalism has already been partially extended to notion of *almost sure stability* in [DJ22a], while the *stabilization problem* of finding a specific switching sequence which stabilizes the system could also be tackled through the prism of the path-complete Lyapunov methods. On the other hand, the stabilization problem in the presence of input variables has been already partially tackled in [DAJJ24, DAG24].

Path-complete neural Lyapunov functions: in this thesis, we have developed in parallel two ways of adding complexity to stability certificates for switched systems, in order to reduce their conservatism: either by using the neural template, or by smartly choosing a mutliple Lyapunov structure appropriate to the template used. We are convinced that these two approaches are complementary, and could be combined to synthesize *neural path-complete Lyapunov functions*. In particular, we expect that we could leverage the information learned by the network to refine the underlying path-complete graph and iteratively produce better stability certificates. In particular, we could maybe draw inspiration from [SP24] where the authors consider neural Lyapunov functions, whose structure is iteratively refined in response to the failure to satisfy the Lyapunov inequalities.

Moreover, it is well known that neural networks are particularly good at capturing and exploiting patterns in data, and graph-structured data especially, see [ZCH$^+$20]. Possible links with this field could be considered.

Application to abstractions: nowadays, abstraction-based techniques are more and more popular [AHKV98, Tab09, RZ16, BPB19, ELJ22, CMGJ24]. These methods rely on a 3-step approach. First, a finite state abstraction of the concrete system is derived for which an abstract controller is synthesized as a second step. Then, the concretization step aims to derive a controller for the concrete system based on the abstract controller. Different abstraction relations between the concrete system and its abstraction have been introduced, each of which leads to a different concretization step. In practice, the properties of the abstraction relations directly influence (or even characterize) the controller design: the initial definition of *feedback refinement relation* [RZ16] (FRR for short) requires to use the same input, and the concrete control is defined as the abstract control quantized. Recently, new abstraction techniques have been introduced in order to build more complex concrete controllers. Moreover, recent results have shown that these new abstraction relations can be translated as a FRR relation between an augmented system and the abstract system. This methodology echoes the lift procedures that we have developed in this thesis. Indeed, we first used the simulation relation to characterize the general ordering of graphs. In practice, the simulation allows to build a solution for a graph $\widetilde{\mathcal{G}}$ from the solution for another graph $\mathcal{G}$, by keeping the same functions. Then, we have characterized the template-dependent ordering of graphs for specific class of templates. In this case, the simulation relation involves a lifted graph which allows to consider more complex relations between the solutions of $\widetilde{\mathcal{G}}$ and $\mathcal{G}$. In brief, we believe that the core ideas of both methodologies are similar: the template-dependent lifts and the new abstraction relations allow to consider more complex expressions for the Lyapunov pieces and for the controller respectively, and then reduce the conservatism. Therefore, both topics could be mutually enhanced, in particular by translating into their formalism the various tools introduced respectively.

In addition, we are convinced that path-complete techniques could be used to smartly refine the abstractions, and thus improve the scalability of these methods. Indeed, both path-complete and abstraction-based methods rely on the choice and/or the construction of combinatorial tools which capture the dynamics of the underlying system. However, in both settings, we are facing similar challenges. In particular, enlarging the graph usually goes with an increase of the computation time, which prevents us from using these techniques for higher dimensional systems: in the abstraction-based techniques, the number of cells of a grid grows exponentially with the dimension, while the number of nodes and edges increases exponentially with respect to the order in the De Bruijn hierarchy. It becomes therefore necessary to introduce smart methods to improve graphs/abstractions, while taking into account the computational cost. This thesis deals specifically with this question, and proposes combinatorial operations to improve path-complete graphs. We argue that the underlying methodology of the lift approach could be generalized to abstraction-based techniques in order to iteratively locally refine a coarse abstraction.

Bibliography

[AAE+21] Alessandro Abate, Daniele Ahmed, Alec Edwards, Mirco Giacobbe, and Andrea Peruffo. FOSSIL: A software tool for the formal synthesis of Lyapunov functions and barrier certificates using neural networks. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, HSCC '21. Association for Computing Machinery (ACM), 2021.

[AB99] Martin Anthony and Peter Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[ABMM18] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *6th International Conference on Learning Representations, ICLR 2018*. OpenReview.net, 2018.

[ADJ+11] Rajeev Alur, Alessandro D'Innocenzo, Karl H. Johansson, George J. Pappas, and Gera Weiss. Compositional modeling and analysis of multi-hop control networks. *IEEE Transactions on Automatic Control*, 56(10):2345–2357, 2011.

[AGMG22] Georges Aazan, Antoine Girard, Paolo Mason, and Luca Greco. Stability of discrete-time switched linear systems with ω-regular switching sequences. In *Proceedings of the 25th International Conference on Hybrid Systems: Computation and Control*, HSCC '22. Association for Computing Machinery (ACM), 2022.

[AHKV98] Rajeev Alur, Thomas A. Henzinger, Orna Kupferman, and Moshe Y. Vardi. Alternating refinement relations. In *International Conference on Concurrency Theory*, 1998.

[AJ14] Amir Ali Ahmadi and Raphaël M. Jungers. On complexity of Lyapunov functions for switched linear systems. *IFAC Proceedings Volumes*, 47(3):5992–5997, 2014. 19th IFAC World Congress.

[AJ18a] Amir Ali Ahmadi and Raphaël M. Jungers. SOS-convex Lyapunov functions and stability of difference inclusions. *arXiv preprint arXiv:1803.02070*, 2018.

[AJ18b] Nikolaos Athanasopoulos and Raphaël M. Jungers. Combinatorial methods for invariance and safety of hybrid systems. *Automatica*, 98:130–140, 2018.

[AJ19] Nikolaos Athanasopoulos and Raphaël M. Jungers. Polyhedral path-complete Lyapunov functions. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3399–3404, 2019.

[AJPR14] Amir Ali Ahmadi, Raphaël M. Jungers, Pablo A. Parrilo, and Mardavij Roozbehani. Joint spectral radius and path-complete graph Lyapunov functions. *SIAM Journal on Control and Optimization*, 52(1):687–717, 2014.

[AJZA] Mahathi Anand, Raphaël M. Jungers, Majid Zamani, and Frank Allgöwer. Path-complete barrier functions for safety of switched linear systems. Submitted for publication to *IEEE 63th Conference on Decision and Control (CDC)*, 2024.

[APAJ17] David Angeli, Matthew Philippe, Nikolaos Athanasopoulos, and Raphaël M. Jungers. Path-complete graphs and common Lyapunov functions. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, HSCC '17, page 81–90, New York, NY, USA, 2017. Association for Computing Machinery.

[Bar13] Alexander Barvinok. Thrifty Approximations of Convex Bodies by Polytopes. *International Mathematics Research Notices*, (16):4341–4356, 2013.

[BBLJ23] Christopher Brix, Stanley Bak, Changliu Liu, and Taylor T. Johnson. The fourth international verification of neural networks competition (VNN-COMP 2023): Summary and results, 2023. arXiv:2312.16760.

[BBS08] Andrzej Białynicki-Birula and Andrzej Schinzel. Representations of multivariate polynomials by sums of univariate polynomials in linear forms. *Colloquium Mathematicae*, 112(2):201–233, 2008.

[BC08] Vincent D. Blondel and Vincent Canterini. Undecidable problems for probabilistic automata of fixed dimension. *Theory of Computing Systems*, 36:231–245, 2008.

[BFJ18] Guillaume O. Berger, Fulvio Forni, and Raphaël M. Jungers. Path-complete p-dominant switching linear systems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6446–6451, 2018.

[Bli88] Wayne D. Blizard. Multiset theory. *Notre Dame Journal of Formal Logic*, 30(1):36 – 66, 1988.

[BM99] Franco Blanchini and Stefano Miani. A new class of universal Lyapunov functions for the control of uncertain linear systems. *IEEE Transactions on Automatic Control*, 44(3):641–647, 1999.

[BM07] Franco Blanchini and Stefano Miani. *Set-Theoretic Methods in Control*. Systems & Control: Foundations & Applications. Birkhäuser, 01 2007.

[BNT05] Vincent D. Blondel, Yurii Nesterov, and Jacques Theys. On the accuracy of the ellipsoid norm approximation of the joint spectral radius. *Linear Algebra and its Applications*, 394:91–107, 2005.

[BP07] Jie Bao and Lee Peter. *Process Control: The Passive Systems Approach*. 2007.

[BPB19] Alessandro Borri, Giordano Pola, and Maria Domenica Di Benedetto. Design of symbolic controllers for networked control systems. *IEEE Transactions on Automatic Control*, 64(3):1034–1046, 2019.

[Bra98] Michael S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.

[BS23] Guillaume O. Berger and Sriram Sankaranarayanan. Counterexample-guided computation of polyhedral lyapunov functions for piecewise linear systems. *Automatica*, 155, 2023.

[BSST21] Clark Barrett, Roberto Sebastiani, Sanjit Seshia, and Cesare Tinelli. Satisfiability modulo theories. In Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability, Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, chapter 33, pages 825–885. 2021.

[BT97] Vincent D. Blondel and John N. Tsitsiklis. When is a pair of matrices mortal? *Information Processing Letters*, 63(5):283–286, 1997.

[BTSK17] Felix Berkenkamp, Matteo Turchetta, Angela P. Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees, 2017. `arXiv:1705.08551`.

[CGPS21] Yacine Chitour, Nicola Guglielmi, Vladimir Y. Protasov, and Mario Sigalotti. Switching systems with dwell time: Computing the maximal Lyapunov exponent. *Nonlinear Analysis: Hybrid Systems*, 40:101021, 2021.

[CL10] Christos G. Cassandras and Stephane Lafortune. *Introduction to Discrete Event Systems*. 2010.

[CLJ+15] Rayan Chikhi, Antoine Limasset, Shaun Jackman, Jared T. Simpson, and Paul Medvedev. On the representation of De Bruijn graphs. *Journal of Computational Biology*, 22(5):336–352, 2015.

[CLM16] Rayan Chikhi, Antoine Limasset, and Paul Medvedev. Compacting de Bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics*, 32(12):i201–i208, 2016.

[CMGJ24] Julien Calbert, Sébastien Mattenet, Antoine Girard, and Raphaël M. Jungers. Memoryless concretization relation. In *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '24, New York, NY, USA, 2024. Association for Computing Machinery (ACM).

[CRG19] Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural lyapunov control. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019.

[Cyb89] George V. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[DAG24] Matteo Della Rossa, Thiago Alves Lima, and Antoine Girard. Feedback stabilization of discrete-time switched systems under Büchi-constrained signals. *IEEE Control Systems Letters*, 8:418–423, 2024.

[Dai12] Xiongping Dai. A gel'fand-type spectral radius formula and stability of linear constrained switching systems. *Linear Algebra and its Applications*, 436(5):1099–1113, 2012.

[DAJJ24] Matteo Della Rossa, Thiago Alves Lima, Marc Jungers, and Raphaël M. Jungers. Graph-based conditions for feedback stabilization of switched and LPV systems. *Automatica*, 160:111427, 2024.

[DB01] Jamal Daafouz and Jacques Bernussou. Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems & Control Letters*, 43(5):355–359, 2001.

[DDJ21] Virginie Debauche, Matteo Della Rossa, and Raphaël M. Jungers. Template-dependent lifts for path-complete stability criteria and application to positive switching systems. *IFAC-PapersOnLine*, 54(5):151–156, 2021. 7th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2021.

[DDJ22a] Virginie Debauche, Matteo Della Rossa, and Raphaël M. Jungers. Comparison of path-complete Lyapunov functions via template-dependent lifts. *Nonlinear Analysis: Hybrid Systems*, 46, 2022.

[DDJ22b] Virginie Debauche, Matteo Della Rossa, and Raphaël M. Jungers. Necessary and sufficient conditions for template-dependent ordering of path-complete Lyapunov methods. In *Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '22. Association for Computing Machinery, 2022.

[DDJ23] Virginie Debauche, Matteo Della Rossa, and Raphaël M. Jungers. Characterization of the ordering of path-complete stability certificates with addition-closed templates. In *Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '23. Association for Computing Machinery, 2023.

[de 46] Nicolaas G. de Bruijn. A combinatorial problem. *Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam*, 49(7):758–764, 1946.

[Deb22] Virginie Debauche. Lift and simulation of path-complete graphs. https://www.codeocean.com/, 2022.

[Deb23] Virginie Debauche. Characterization of ordering of path-complete graphs for addition-closed templates. https://www.codeocean.com/, 2023.

[DEJA24] Virginie Debauche, Alec Edwards, Raphaël M. Jungers, and Alessandro Abate. Stability analysis of switched linear systems with neural Lyapunov functions. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, AAAI '24. AAAI Press, 2024.

[DGTZ21] Matteo Della Rossa, Rafal Goebel, Aneel Tanwani, and Luca Zaccarian. Piecewise structure of Lyapunov functions and densely checked decrease conditions for hybrid systems. *Mathematics of Control, Signals, and Systems*, 33:123–149, 2021.

[DHvdWH11] M. C. F. Tijs Donkers, W. P. Maurice H. Heemels, Nathan van de Wouw, and Laurentiu Hetel. Stability analysis of networked control systems using a switched linear systems approach. *IEEE Transactions on Automatic Control*, 56(9):2101–2115, 2011.

[DJ20] Virginie Debauche and Raphaël M. Jungers. On path-complete Lyapunov functions : comparison between a graph and its expansion. In *Proceedings of the 39th Benelux Meeting on Systems and Control*, Benelux Meeting, 2020.

[DJ22a] Matteo Della Rossa and Raphaël M. Jungers. Almost sure stability of stochastic switched systems: Graph lifts-based approach. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 1021–1026, 2022.

[DJ22b] Matteo Della Rossa and Raphaël M. Jungers. Memory-based Lyapunov functions and path-complete framework: Equivalence and properties. In *2022 10th International Conference on Systems and Control (ICSC)*, pages 12–17, 2022.

[DJ23] Matteo Della Rossa and Raphaël M. Jungers. Multiple Lyapunov functions and memory: A symbolic dynamics approach to systems and control. In *SIAM Journal on Control and Optimization (SICON) (to appear)*, 2023.

[DK22] Steffen Dereich and Sebastian Kassing. On minimal representations of shallow ReLU networks. *Neural Networks*, 148:121–128, 2022.

[dMB08] Leonardo de Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. volume 4963, pages 337–340, 2008.

[DPA22] Matteo Della Rossa, Mirko Pasquini, and David Angeli. Continuous-time switched systems with switching frequency constraints: Path-complete stability criteria. *Automatica*, 137:110099, 2022.

[DQGF21] Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. Safe nonlinear control using robust neural Lyapunov-barrier functions. In *Conference on Robot Learning*, 2021.

[ELJ22] Lucas N. Egidio, Thiago Alves Lima, and Raphaël M. Jungers. State-feedback abstractions for optimal control of piecewise-affine systems. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 7455–7460, 2022.

[EPA24] Alec Edwards, Andrea Peruffo, and Alessandro Abate. Fossil 2.0: Formal certificate synthesis for the verification and control of dynamical models, 2024.

[FLYL22] Milad Farsi, Yinan Li, Ye Yuan, and Jun Liu. A piecewise learning framework for control of unknown nonlinear systems with stability guarantees. In *Learning for Dynamics and Control Conference, L4DC*, volume 168 of *Proceedings of Machine Learning Research*, pages 830–843. PMLR, 2022.

[FMP21] Mahyar Fazlyab, Manfred Morari, and George J Pappas. An introduction to neural network analysis via semidefinite programming. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6341–6350. IEEE, 2021.

[FV12] Ettore Fornasini and Maria Elena Valcher. Stability and stabilizability criteria for discrete-time positive switched systems. *IEEE Transactions on Automatic Control*, 57(5):1208–1221, 2012.

[GFPC18] Henry G. R. Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J. Cree. Regularisation of neural networks by enforcing Lipschitz continuity. *Machine Learning*, 110:393 – 416, 2018.

[GHT06] Rafal Goebel, Tingshu Hu, and Andrew R. Teel. *Dual Matrix Inequalities in Stability and Performance Analysis of Linear Differential/Difference Inclusions*, pages 103–122. Birkhäuser Boston, 2006.

[GTHL06] Rafal Goebel, Andrew R. Teel, Tingshu Hu, and Zongli Lin. Conjugate convex Lyapunov functions for dual linear differential inclusions. *IEEE Transactions on Automatic Control*, 51(4):661–666, 2006.

[GWZ05] Nicola Guglielmi, Fabian Wirth, and Marino Zennaro. Complex polytope extremality results for families of matrices. *SIAM Journal on Matrix Analysis and Applications*, 27(3):721–743, 2005.

[GZ08] Nicola Guglielmi and Marino Zennaro. An algorithm for finding extremal polytope norms of matrix families. *Linear Algebra and its Applications*, 428:2265–2282, 2008.

[GZ19] Guodong Guo and Na Zhang. A survey on deep learning based face recognition. *Computer Vision and Image Understanding*, 189:102805, 2019.

[Hal35] Philip Hall. On representatives of subsets. *Journal of the London Mathematical Society*, s1-10(1):26–30, 1935.

[HBDSS21] Christoph Hertrich, Amitabh Basu, Marco Di Summa, and Martin Skutella. Towards lower bounds on the depth of ReLU neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 3336–3348. Curran Associates, Inc., 2021.

[HGB04] Yildirim Hurmuzlu, Frank Génot, and Bernard Brogliato. Modeling, stability and control of biped robots a general framework. *Automatica*, 40:1647–1664, 2004.

[Hil88] David Hilbert. Ueber die darstellung definiter formen als summen von formenquadraten. *Mathematische Annalen*, 32:342–350, 1888.

[HN04] Pavol Hell and Jaroslav Nešetřil. *Graphs and Homomorphisms*, volume 28 of *Oxford Lecture Series in Mathematics and Its Applications*. Oxford University Press, 2004.

[HN08] Pavol Hell and Jaroslav Nešetřil. Colouring, constraint satisfaction, and complexity. *Computer Science Review*, 2(3):143–163, 2008.

[Hor91] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[HT97] Gena Hahn and Claude Tardif. Graph homomorphisms: Structure and symmetry. *Graph Symmetry*, 497:107–166, 1997.

[HVCMB11] Esteban Abelardo Hernandez Vargas, Patrizio Colaneri, Richard H. Middleton, and Franco Blanchini. Discrete-time control for switched positive systems with application to mitigating viral escape. *International Journal of Robust and Nonlinear Control*, 21:1093 – 1111, 2011.

[JAPR17] Raphaël M. Jungers, Amir Ali Ahmadi, Pablo A. Parrilo, and Mardavij Roozbehani. A characterization of Lyapunov inequalities for stability of switched systems. *IEEE Transactions on Automatic Control*, 62(6):3062–3067, 2017.

[JCG14] Raphaël M. Jungers, Antonio Cicone, and Nicola Guglielmi. Lifted polytope methods for computing the joint spectral radius. *SIAM Journal on Matrix Analysis and Applications*, 35:391–410, 2014.

[JHK16] Raphaël M. Jungers, W.P.M.H. (Maurice) Heemels, and Atreyee Kundu. Observability and controllability analysis of linear systems subject to data losses. *IEEE Transactions on Automatic Control*, PP, 2016.

[Joh48] Fritz John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays, in Honor of R. Courant,*, pages 187–204. Interscience, New York, 1948.

[Jun09] Raphaël M. Jungers. *The Joint Spectral Radius: Theory and Applications*, volume 385 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2009.

[Jun24] Raphaël M. Jungers. Statistical comparison of path-complete lyapunov functions: a discrete-event systems perspective. *IFAC-PapersOnLine*, 2024. 17th IFAC Workshop on Discrete Event Systems WODES 2024.

[JZHZH13] Junfeng J. Zhang, Zhengzhi Han, Fubo Zhu, and Jun Huang. Stability and stabilization of positive switched systems with mode-dependent average dwell time. *Nonlinear Analysis: Hybrid Systems*, 9:42–55, 2013.

[KB15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[Koz90] Victor Kozyakin. Algebraic unsolvability of a problem on the absolute stability of desynchronized systems. *Automation and Remote Control*, 51:754–759, 1990.

[KT04] Christopher M. Kellett and Andrew R. Teel. Smooth Lyapunov functions and robustness of stability for difference inclusions. *Systems & Control Letters*, 52(5):395 – 405, 2004.

[LA09] Hai Lin and Panos Antsaklis. Stability and stabilizability of switched linear systems: A survey of recent results. *Automatic Control, IEEE Transactions on*, 54:308 – 322, 2009.

[Lay44] Steven R. Lay. *Convex Sets and Their Applications*. A Wiley-Interscience publication. New York:Wiley, 1944.

[LB08] Fabien Lauer and Gérard Bloch. Switched and piecewise nonlinear hybrid system identification. 2008.

[LDH20] Donghwan Lee, Geir E. Dullerud, and Jianghai Hu. Graph Lyapunov function for switching stabilization and distributed computation. *Automatica*, 116:108923, 2020.

[LH17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2017.

[LH20] Donghwan Lee and Niao He. A unified switching system perspective and convergence analysis of q-learning algorithms. In *Advances in Neural Information Processing Systems*, volume 33, pages 15556–15567. Curran Associates, Inc., 2020.

[Lib03] Daniel Liberzon. *Switching in Systems and Control*. Systems & Control: Foundations & Applications. Birkhäuser, 2003.

[LM99] Daniel Liberzon and A. Stephen Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19(5):59–70, 1999.

[Löf04] Johan Löfberg. YALMIP : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[LPW+17] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[LS17] Shiyu Liang and R. Srikant. Why deep neural networks for function approximation? In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[MB17] Anirbit Mukherjee and Amitabh Basu. Lower bounds over boolean inputs for deep neural networks with ReLU gates. *ArXiv*, abs/1711.03073, 2017.

[MKKY18] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[MOS19] ApS MOSEK. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019. URL: `http://docs.mosek.com/9.0/toolbox/index.html`.

[MS07] Oliver Mason and Robert Shorten. On linear copositive Lyapunov functions and the stability of switched positive linear systems. *IEEE Transactions on Automatic Control*, 52(7):1346–1349, 2007.

[OMK21] Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624, 2021.

[PAAJ19] Matthew Philippe, Nikolaos Athanasopoulos, David Angeli, and Raphaël M. Jungers. On path-complete Lyapunov functions: Geometry and comparison. *IEEE Transactions on Automatic Control*, 64(5):1947–1957, 2019.

[PEDJ15] Matthew Philippe, Ray Essick, Geir Dullerud, and Raphaël M. Jungers. Stability of discrete-time switching systems with constrained switching sequences. *Automatica*, 72, 2015.

[Pep19] Pierdomenico Pepe. Converse Lyapunov theorems for discrete-time switching systems with given switches digraphs. *IEEE Transactions on Automatic Control*, 64(6):2502–2508, 2019.

[Phi17] Matthew Philippe. *Path-Complete Methods and analysis of constrained switching systems*. Phd thesis, UCLouvain, 2017.

[PJ08] Pablo A. Parrilo and Ali Jadbabaie. Approximation of the joint spectral radius using sum of squares. *Linear Algebra and its Applications*, 428(10):2385–2402, 2008. Special Issue on the Joint Spectral Radius: Theory, Methods and Applications.

[PJ19] Matthew Philippe and Raphaël M. Jungers. A complete characterization of the ordering of path-complete methods. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '19, page 138–146. Association for Computing Machinery (ACM), 2019.

[PJB10] Vladimir Protasov, Raphaël M. Jungers, and Vincent D. Blondel. Joint spectral characteristics of matrices: A conic programming approach. *SIAM J. Matrix Analysis Applications*, 31:2146–2162, 2010.

[Pro94] Danil V. Prokhorov. A Lyapunov machine for stability analysis of nonlinear systems. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 2, pages 1028–1031, 1994.

[Roc70] Ralph T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[Ros92] Lionel Rosier. Homogeneous Lyapunov function for homogeneous continuous vector field. *Systems & Control Letters*, 19(6):467–473, 1992.

[RS60] Gian-Carlo Rota and W. Gilbert Strang. A note on the joint spectral radius. In *Proceedings of the Netherlands Academy*, pages 379–381, 1960.

[RZ16] Matthias Rungger and Majid Zamani. SCOTS: A tool for the synthesis of symbolic controllers. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, HSCC '16, page 99–104. Association for Computing Machinery (ACM), 2016.

[SBM22] Yue Shen, Maxim Bichuch, and Enrique Mallada. Model-free learning of regions of attraction via recurrent sets. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 4714–4719, 2022.

[Ser05] Gursel Serpen. Empirical approximation for Lyapunov functions with artificial neural nets. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 735–740, 2005.

[SJ24] Somya Singh and Raphaël M. Jungers. Using symbolic dynamics to compare path-complete Lyapunov functions. In *2024 IEEE 58th Conference on Decision and Control (CDC)*, 2024.

[SP24] Pouya Samanipour and Hasan A. Poonawala. Stability analysis and controller synthesis using single-hidden-layer ReLU neural networks. *IEEE Transactions on Automatic Control*, 69(1):202–213, 2024.

[SWM+07] Robert Shorten, Fabian Wirth, Oliver Mason, Kai Wulff, and Christopher King. Stability criteria for switched and hybrid systems. *SIAM Review*, 49(4):545–592, 2007.

[Tab09] Paulo Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. 2009.

[VHJ14] Guillaume Vankeerberghen, Julien Hendrickx, and Raphaël M. Jungers. JSR: a toolbox to compute the joint spectral radius. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*, HSCC '14, page 151–156. Association for Computing Machinery (ACM), 2014.

[VS18] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[WIZ+24] Haoze Wu, Omri Isac, Aleksandar Zeljić, Teruhiro Tagomori, Matthew Daggitt, Wen Kokke, Idan Refaeli, Guy Amir, Kyle Julian, Shahaf Bassan, Pei Huang, Ori Lahav, Min Wu, Min Zhang, Ekaterina Komendantskaya, Guy Katz, and Clark Barrett. Marabou 2.0: A versatile formal analyzer of neural networks, 2024. arXiv:2401.14461.

[WJ20] Zheming Wang and Raphaël M. Jungers. A data-driven method for computing polyhedral invariant sets of black-box switched linear systems. *IEEE Control Systems Letters*, 2020.

[WS05] Shuning Wang and Xusheng Sun. Generalization of Hinging hyperplanes. *IEEE Transactions on Information Theory*, 51:4425–4431, 2005.

[WZG13] Ligang Wu, Wei Xing Zheng, and Huijun Gao. Dissipativity-based sliding mode control of switched stochastic systems. *IEEE Transactions on Automatic Control*, 58:785–791, 2013.

[YDS+24] Lujie Yang, Hongkai Dai, Zhouxing Shi, Cho-Jui Hsieh, Russ Tedrake, and Huan Zhang. Lyapunov-stable neural control for state and output feedback: A novel formulation, 2024. arXiv:2404.07956.

[YM17] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning, 2017. arXiv:1705.10941.

[ZCH+20] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[Zha00] Peter Zhang. Neural networks for classification: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30:451 – 462, 2000.

[ZXQF23] Songyuan Zhang, Yumeng Xiu, Guannan Qu, and Chuchu Fan. Compositional neural certificates for networked dynamical systems. In *Learning for Dynamics and Control Conference, L4DC 2023*, volume 211 of *Proceedings of Machine Learning Research*, pages 272–285. PMLR, 2023.