A complete characterization of the ordering of Path-Complete methods*

Matthew Philippe
Université catholique de Louvain
matthew.philippe@uclouvain.be

Raphaël M. Jungers
Université catholique de Louvain
raphael.jungers@uclouvain.be

ABSTRACT

We study criteria allowing to compare the conservativeness of stability certificates for switching systems. The stability certificates under consideration are Path-Complete Lyapunov functions (PCLFs), which are multiple Lyapunov functions with an underlying combinatorial structure.

Several criteria exist to compare these certificates and we focus here on the recently introduced concept of *simulation*. This criterion, which is borrowed from automata theory, relies only on the combinatorial structures that underlie the PCLFs. We show that the concept of simulation provides a complete characterization of the ordering relation between PCLFs, independently of the algebraic properties of the dynamical system studied and the template of Lyapunov functions used. We summarize the state of the art on the comparison of PCLFs and raise future research directions.

CCS CONCEPTS

• **Mathematics of computing**; • **Theory of computation** → **Mathematical optimization**; • **Computer systems organization** → **Real-time system specification**;

KEYWORDS

Path-Complete Methods, Lyapunov stability theory, Conservativeness, Automata, Switching Systems.

ACM Reference Format:
Matthew Philippe and Raphaël M. Jungers. 2019. A complete characterization of the ordering of Path-Complete methods. In *22nd ACM International Conference on Hybrid Systems: Computation and Control (HSCC '19), April 16–18, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3302504.3311803

1 INTRODUCTION

Switching systems [20, 24, 26, 37] are adequate for modeling many processes [12, 16, 17, 27, 36], and may act as abstractions for more complex hybrid dynamical systems [15]. Their study presents major theoretical challenges [8], and among these, stability analysis is central and has been the focus of many works (see e.g. [20, 25, 26, 33, 37] and references therein).

A *discrete-time switching system* is a dynamical system of the form

$$x_{t+1} = f_{\sigma(t)}(x_t), \tag{1}$$

where at time $t \in \mathbb{N}$, the *state* x_t lies in $\mathbb{R}^n$, the *mode* of the system $\sigma(t)$ lies in a set $\langle M \rangle = \{1, \ldots, M\}$ for an integer M, and the maps $f_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ for all $\sigma \in \langle M \rangle$ are called the *dynamics* of the system. We call the functions $t \to \sigma(t)$ that select a mode for each time the *switching sequences*.

Given their range of application, switching systems have attracted tremendous attention from the research community. It is now a known fact that these systems are inherently hard to control and analyze. In particular, negative results have been proven regarding their *stability analysis*.

At lot of attention has been devoted to the stability analysis of switching systems *under arbitrary switching*, i.e. where *switching sequences can be arbitrary*. Already there, the problem of deciding whether or not a system is stable is difficult, and in general undecidable (see e.g. [8, 20]).

More generally, one can consider systems where the switching sequences (seen as *infinite words* over the *alphabet* $\langle M \rangle$), are restricted to belong to a regular language $\mathscr{L}$. We call these *constrained switching systems*; see e.g. [7, 11, 13, 21, 23, 30, 32–34] and references therein. Let us emphasize two applications in Hybrid Systems theory where a constrained switching system model can be used to encapsulate the complexity of the system:

- For a *general Hybrid automaton*, one can make abstraction of the transition conditions (the guards), and use the constrained switching systems framework as a coarse model of the hybrid automaton;
- In *abstraction and formal methods*, the state-space and the control inputs are discretized, so that formally the closed-loop system behaves as a constrained switching system, where the different dynamics correspond, e.g. to the different values of the discretized control input.

Given the hardness of assessing the stability of a switching system, several strategies and conservative conditions have been proposed. Among these, it is common to rely on *Lyapunov methods*. For instance, multiple quadratic Lyapunov functions [9, 18, 37] have attracted a lot of attention for the study of *linear switching systems*. Such tools are particularly appealing because their computation involves only convex optimization, such as in [1, 11, 13, 23, 33].

Motivated by the popularity of these techniques based on convex optimization, Ahmadi et al. introduced the concept of *Path-Complete Lyapunov functions [1]* to form a unified framework for their study and allow to compare them on common grounds. Using this analytical framework, the authors provided new insights and converse

*R.M.J. is a FNRS Research Associate. He is supported by the Walloon Region and the Innoviris Foundation. M.P. is supported by the Walloon Region.

HSCC '19, April 16–18, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6282-5/19/04…$15.00
https://doi.org/10.1145/3302504.3311803

theorems for e.g. the parameter-dependant Lyapunov functions of [11] and the path-dependent Lyapunov functions of [23].

Since their introduction for the stability analysis of arbitrary switching systems, Path-Complete Lyapunov functions have been generalized and the framework expanded [30]. They can be used to analyze the constrained switching systems mentionned above [33], and study notions of invariance and irreducibility [34, 35], performance analysis [32], and control [14]. Recent related research includes the concept of path-dominance [6], analysis of non-linear switching systems [29], invariance safety and reachability analysis [4, 5] and switching stabilization [19].

The Path-Complete framework provides guidelines for constructing stability analysis algorithms. However, it remains unclear when one of such algorithm provides less conservative stability certificates and another. As the framework grows more popular within the Hybrid Systems community, there is a need for a better understanding of its performances. As a result, severals tools have been designed to compare Path-Complete Lyapunov functions [2, 3, 30], that provide sufficient conditions for when one certificate ourperforms another. In this paper, we focus on the criterion known as *simulation*. To our knowledge, this criterion, inspired from the concept of simulation in Automata theory [10], is the first systematic tool for comparing Path-Complete Lyapunov functions [3]. Our goal is to provide a full understanding of the scope of the criterion. This goal somehow differs from that of the previously mentioned works, where the aim was rather propose more and more powerful sufficient conditions for comparing Path-Complete Lyapunov functions.

Outline. In Section 2, we introduce the main elements of the Path-Complete framework, and formalize the precise question to be tackled here, i.e., *when can one say that one Path-Complete method always produces less conservative certificates than another ?* Our technical contributions are presented in Section 3. More precisely, Subsection 3.1 introduces our main result, Theorem 3.5, that states that in the absence of any specifications on the dynamics or templates of Lyapunov functions involved, the *simulation* criterion provides a full characterization of the ordering of Path-Complete methods. Subsection 3.2 follows with the introduction of a technical result of independant interest (Theorem 3.6) playing a key role for the proof of Theorem 3.5, and subsequently presents the proof of the main result itself. Ending the section and opening further discussions, Subsection 3.3 presents simple examples showing that the comparison problem becomes much richer when considering specific families of systems or template of Lyapunov function. Finally, Section 4 concludes our work.
Examples are used along the paper to illustrate and clarify key aspects of the discussion.

2 PROBLEM SETTING

We study a family of techniques for stability analysis known as *Path-Complete methods.* Section 2.1 first introduces the notations and definitions related to Path-Complete methods. Section 2.2 then presents a numerical experiment highlighting the issues tackled herein: the ordering of Path-Complete methods with respect to their conservativeness. Finally Section 2.3 formalizes the problem of the ordering of Path-Complete methods.

2.1 Preliminaries

We borrow concepts from theoretical computer science for describing switching sequences. For an integer M, the *alphabet* $\langle M\rangle = \{1, \dots, M\}$ describes the set of *modes* of a system. A *word* on this alphabet is an infinite sequence $w = \sigma_0\sigma_1 \dots$ with $\sigma_i \in \langle M\rangle$, and a *language* $\mathscr{L}$ is a set of words.

Path-complete methods [30] (in the context of stability analysis) provide tools to decide *stability over a language:*

Definition 2.1 (Stability over a language). *Consider a system on M modes $F = \{f_\sigma, \sigma \in \langle M\rangle\}$ and a language $\mathscr{L}$ of words on $\langle M\rangle$. We say that F if stable over $\mathscr{L}$ if there exists a $\mathcal{K}_\infty$ function[1] $k : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that*

$$\forall x \in \mathbb{R}^n, \forall \sigma_0\sigma_1, \dots \in \mathscr{L}, \forall t \in \mathbb{N} : \|f_{\sigma_{t-1}} \circ \dots \circ f_{\sigma_0}(x)\| \leq k(\|x\|).$$

In the definition above, $||\cdot||$ represents an arbitrary norm in $\mathbb{R}^n$.
When clear from the context, we use the term *system* to refer both to a switching system (1) and its dynamics $F = \{f_\sigma, \sigma \in \langle M\rangle\}$, as in "the system F is stable over the language $\mathscr{L}$".

Remark 2.2. *While we focus here on stability criteria, our analysis extends directly to the study of other properties such as stabilizability or performance (see e.g. [30, Chapter 5], [22]).*

Path-Complete methods involve both algebraic (Lyapunov functions) and combinatorial (graphs) components.
The combinatorial component of a Path-Complete method is a graph $\mathbf{G}$ such as presented in e.g. Figures 1a and 1b. Its role is to encode a language $\mathscr{L}$ over which we aim to decide the stability of a system. A graph on M modes is a pair $\mathbf{G} = (S, E)$, where S is a set of states and $E \subseteq S \times S \times \langle M\rangle$ is a set of directed and labeled edges between these states. For any sequence $\sigma_t, t \in \mathbb{N}$ with $\sigma_t \in \langle M\rangle$, we say that the sequence is *accepted* by the graph if we can match it to a sequence of edges $(s_t, s_{t+1}, \sigma_t), t \in \mathbb{N}$. We can then define the *language* of a graph $\mathbf{G}$, written $\mathscr{L}(G)$, as the set of all sequences $\sigma_t, t \in \mathbb{N}$ that are accepted by $\mathbf{G}$.
The algebraic component of a Path-Complete method is a set of *candidate Lyapunov functions.*

Definition 2.3. *A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a* candidate Lyapunov function *(CLF) if there are $\mathcal{K}_\infty$ functions α and β satisfying*

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

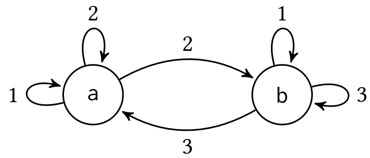
Together, graphs and candidate Lyapunov functions lead to the concept of *Path-Complete Lyapunov function.*

Definition 2.4 (Path-Complete Lyapunov function). *Given a switching system $F = \{f_\sigma, \sigma \in \langle M\rangle\}$ on $\mathbb{R}^n$, a* Path-Complete lyapunov function *(PCLF) for F is a pair $(\mathbf{G} = (S, E), \{V_s, s \in S\})$ where $\{V_s, s \in S\}$ is a set of CLFs such that the following Lyapunov inequalities are satisfied*

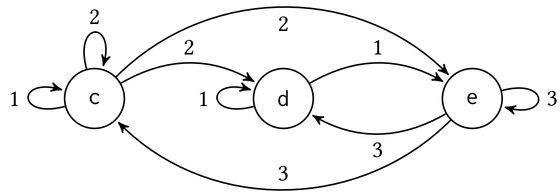
$$\forall (s, d, \sigma) \in E, \forall x \in \mathbb{R}^n : V_d(f_\sigma(x)) \leq V_s(x). \tag{2}$$

The set of CLFs $\{V_s, s \in S\}$ is said to be admissible *for $\mathbf{G}$ and F if it satisfies* (2).

[1] A scalar function is of class $\mathcal{K}_\infty$ if it is continuous, positive definitive, strictly increasing, and unbounded.



(a) Graph G_1 for the experiment of Section 2.2



(b) Graph G_2 for the experiment of Section 2.2

Figure 1: Graphs for the experiment of Section 2.2

The following theorem summarizes the use of the Path-Complete framework for stability analysis, see e.g. [30, Theorem 5.12].

THEOREM 2.5. *Consider a switching system with dynamics $F = \{f_\sigma, \sigma \in \langle M \rangle\}$ acting on $\mathbb{R}^n$. If there exists a* Path-Complete Lyapunov function ($\mathbf{G} = (S, E), \{V_s, s \in S\}$) *for F, then the switching system is stable over the language $\mathscr{L}(\mathbf{G})$.*

We stress that in general the existence of a *Path-Complete Lyapunov function* is only a sufficient condition for the stability of a system. In practice, the search for a Path-Complete Lyapunov function for a system F proceeds in three steps:

- First, chose a parameterization for the CLFs, which we later call *template* (see Section 2.3).
- Second, chose a graph $\mathbf{G} = (S, E)$ with the appropriate language.
- Third, search for a set of CLFs $\{V_s, s \in S\}$ that would satisfy the inequalities (2).

In the next section, we explain how different choices of $\mathbf{G}$ can affect the ability to find a PCLF for a given system. In other words, choosing different graphs can affect the conservativeness of the approach.

2.2 A numerical experiment.

In this experiment, we are comparing two Path-Complete methods for the stability analysis of systems on $M = 3$ modes. The first uses the graph $\mathbf{G}_1 = (S_1, E_1)$ depicted in Figure 1a, and the second uses the graph $\mathbf{G}_2 = (S_2, E_2)$ depicted in Figure 1b. Both graphs have the same language. We ask the following question: *"knowing that the methods built on each graph are only sufficient conditions for stability, is one more likely to produce a certificate than another ?"*

The experiment that we report here proceeds by sampling many random systems and evaluating the conservativeness of both methods on each[2]. For the sake of computations, we consider *linear switching systems* and will seek to compute CLFs in the set $\mathcal{V}$ of *quadratic positive definite functions*. We generate a set $\mathcal{F}$ of 10000 *linear switching systems*, i.e. where each $F = \{f_1, f_2, f_3\} \in \mathcal{F}$ has the form

$$\forall \sigma \in \langle 3 \rangle, \forall x \in \mathbb{R}^n : f_\sigma(x) = A_\sigma x.$$

For each system, we generate three matrices $A_\sigma \in \mathbb{R}^{3\times 3}$ for which each entry is a be random gaussian variable with mean 0 and variance 1.

For each system $F \in \mathcal{F}$ and for each graph $\mathbf{G}_i = (S_i, E_i), i \in \{1, 2\}$, we compute the following quantity[3]:

$$\rho_{\mathcal{V}}(F; \mathbf{G}_i) = \begin{cases} \inf\limits_{V_s, s \in S_i; \rho} \rho, \\ \forall (s, d, \sigma) \in E_i, \forall x \in X : V_d\left(\dfrac{f_\sigma}{\rho}(x)\right) \leq V_s(x), \\ V_s \in \mathcal{V}. \end{cases} \quad (3)$$

The quantity $\rho_{\mathcal{V}}(F, \mathbf{G}_i)$ is the least upper bound on the decay rate of the system F that can be obtained using a Path-Complete Lyapunov function with quadratic pieces and with the graph $\mathbf{G}_i$, see e.g. [33]. The smaller it is, the less conservative is the associated method with respect to the task of assessing stability for the given system. We then let $\tilde{\rho} = \min(\rho_{\mathcal{V}}(F; \mathbf{G}_1), \rho_{\mathcal{V}}(F; \mathbf{G}_2))$ be the best decay rate approximation obtained for F so far.
Finally, we say that $\mathbf{G}_i$ succeeded at computing[4] $\tilde{\rho}$ if $\rho_{\mathcal{V}}(F; \mathbf{G}_i) = \tilde{\rho}$. Otherwise it fails.

Running the experiment over the whole family of systems $\mathcal{F}$, we observe that:

- for roughly 2/3 (i.e., 6656 systems out of the 10000) of the systems it was the case that
$$\rho_{\mathcal{V}}(F; \mathbf{G}_1) = \rho_{\mathcal{V}}(F; \mathbf{G}_2),$$
meaning that both methods appear to be equivalent in performances there;
- for the rest (the last 1/3, i.e. 3344 systems out of the 10000), it is the case that
$$\rho_{\mathcal{V}}(F; \mathbf{G}_1) > \rho_{\mathcal{V}}(F; \mathbf{G}_2),$$
hence the method using $\mathbf{G}_2$ appears better suited for these systems.

We stress that there are no systems in $\mathcal{F}$ for which

$$\rho_{\mathcal{V}}(F; \mathbf{G}_1) < \rho_{\mathcal{V}}(F; \mathbf{G}_2).$$

This experiment seems to indicate that using $\mathbf{G}_2$ systematically leads to less conservative stability certificates compared to using $\mathbf{G}_1$. Hence, there seems to be an intuitive *ordering* between $\mathbf{G}_1$ and $\mathbf{G}_2$ (the first providing more conservative certificates over systems $\mathcal{F}$ than the second). Of course, at this point, it is unsure whether or not this would remain true if one picks

- a different family of system $\mathcal{F}$ and/or
- a different choice for the set of CLFs $\mathcal{V}$.

[2] The matlab code for reproducing the experiment is available at http://sites.uclouvain.be/scsse/philippe-jungers_hscc19_ACompleteChar.zip, and makes use of the CSSystem Matlab toolbox https://www.mathworks.com/matlabcentral/fileexchange/52723-the-cssystem-toolbox

[3] For linear systems and quadratic CLFs, this can be done using convex optimization and a bisection procedure, see e.g. [33].

[4] In practice, the equality $\rho_{\mathcal{V}}(F; \mathbf{G}_i) = \tilde{\rho}$ is considered satisfied if it is satisfied up to the bisection precision.

2.3 Deciding ordering

Theorem 2.5 offers a family of algorithms for stability analysis of a system F over a given language $\mathscr{L}$. As illustrated in section 2.2, these algorithms are parameterized mainly by two mathematical objects:

- **A template for candidate Lyapunov functions (CLFs)**, which is a *set* of CLFs in which we search for admissible Lyapunov function candidates. In what follows, we use $\mathcal{V}$, to denote such as set. For example, we may set $\mathcal{V}$ to be the set of quadratic positive definite functions.
- **A graph** $\mathbf{G} = (S, E)$ with $\mathscr{L}(\mathbf{G}) \supseteq \mathscr{L}$.

Once these objects have been selected, one can assess the stability of a system F by computing the quantity $\rho_{\mathcal{V}}(F; \mathbf{G})$ as defined in (3). Let us expand on the signification and role of this quantity.

In the particular context where we take F to be a linear system on $\mathbb{R}^n$ and a template $\mathcal{V}$ of homogeneous functions (with the same degree of homogeneity), $\rho_{\mathcal{V}}(F; \mathbf{G})$ is an upper bound on the exponential decay rate of the system. As shown in [33], if we take $\mathcal{V}$ to be *the set of norms in* $\mathbb{R}^n$, then $\rho_{\mathcal{V}}(F; \mathbf{G})$ is known as a *constrained-joint spectral radius* (i.e., it is the least upper bound on the exponential decay rate of trajectories with switching sequences in $\mathscr{L}(\mathbf{G})$). In the same setting, if $\mathbf{G}$ accepts any arbitrary sequence of symbols in $\langle M \rangle$, $\rho_{\mathcal{V}}(F; \mathbf{G})$ is the *joint-spectral radius* [20] of the set of matrices defining the dynamics of F. Hence, in general, the quantity $\rho_{\mathcal{V}}(F; \mathbf{G})$ can be understood intuitively as a measure of the speed of convergence of the trajectories of a system towards the origin, for switching sequences in $\mathscr{L}(\mathbf{G})$.

We use the quantity $\rho_{\mathcal{V}}(F; \mathbf{G})$ as a mean to compare Path-Complete methods. For a given *set of systems* $\mathcal{F} = \{F_1, F_2, \ldots\}$, given two graphs $\mathbf{G}_1, \mathbf{G}_2$ on the same language and two templates $\mathcal{V}_1, \mathcal{V}_2$, one can interpret the relation

$$\forall F \in \mathcal{F} : \rho_{\mathcal{V}_1}(F; \mathbf{G}_1) \leq \rho_{\mathcal{V}_2}(F; \mathbf{G}_2)$$

as the fact that, for all systems in $\mathcal{F}$, Path-Complete methods using $\mathcal{V}_1$ and $\mathbf{G}_1$ produce less conservative stability certificates that those using $\mathcal{V}_2$ and $\mathbf{G}_2$. The ability to exhibit such a relation is useful in practice for selecting which algorithm to use for a given task.

Our research, as in [1, 23, 33], focuses on the case where $\mathcal{V}_1$ and $\mathcal{V}_2$ are the same. This differs from e.g. [28], where the focus is rather on providing hierarchies of templates $\mathcal{V}_1, \mathcal{V}_2, \ldots$, (i.e., sum-of-square functions of increasing degree) leading to less and less conservative certificates. We seek to study the properties of *graphs* that lead to one method being more conservative than another.

Definition 2.6 (Graph Ordering). *Given an alphabet* $\langle M \rangle$ *and a language* $\mathscr{L}$*, a* family of switching systems $\mathcal{F}$ *on* $\mathbb{R}^n$*, a template* $\mathcal{V}$ *of candidate Lyapunov functions on* $\mathbb{R}^n$ *and two graphs* $\mathbf{G}_1$, $\mathbf{G}_2$ *where* $\mathscr{L}(\mathbf{G}_1) = \mathscr{L}(\mathbf{G}_2) = \mathscr{L}$*, we write*

$$\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$$

if, for any switching system $F \in \mathcal{F}$*,*

$$\rho_{\mathcal{V}}(F; \mathbf{G}_2) \leq \rho_{\mathcal{V}}(F; \mathbf{G}_1). \tag{4}$$

Let us illustrate the notions above through the experiment of Section 2.2. We observed that, for a particular family $\mathcal{F}$ of 10.000 linear systems, and for the template $\mathcal{V}$ of quadratic positive definite Lyapunov functions, we had

$$\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2,$$

where $\mathbf{G}_1$ and $\mathbf{G}_2$ can be found at Figures 1a and 1b respectively. We then asked whether $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$ held in *general regardless of the choice of* $\mathcal{F}$ *and* $\mathcal{V}$. This *general ordering* is central in this paper, and we fully characterize it in terms of joint properties of $\mathbf{G}_1$ and $\mathbf{G}_2$ in our main result Theorem 3.5.

Definition 2.7 (General Graph-Ordering). *Given an alphabet* $\langle M \rangle$ *and a language* $\mathscr{L}$ *and two graphs* $\mathbf{G}_1$, $\mathbf{G}_2$ *where* $\mathscr{L}(\mathbf{G}_1) = \mathscr{L}(\mathbf{G}_2) = \mathscr{L}$*, we write*

$$\mathbf{G}_1 \leq \mathbf{G}_2$$

if it holds that $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$*, for any* family of switching systems $\mathcal{F}$ *on* $\mathbb{R}^n$ *and template* $\mathcal{V}$ *of candidate Lyapunov functions on* $\mathbb{R}^n$.

As we now show, the answer to the question above is positive.

Proposition 2.8. *Consider the graphs* $\mathbf{G}_1 = (S_1, E_1)$ *and* $\mathbf{G}_2 = (S_2, E_2)$ *in Figures 1a and 1b respectively. The graphs satisfy*

$$\mathbf{G}_1 \leq \mathbf{G}_2.$$

sketch of proof. Take any system F in $\mathbb{R}^n$ and template $\mathcal{V}$. For any $\rho > \rho_{\mathcal{V}}(F; \mathbf{G}_1)$, there exists a set of CLFs $\{V_{\mathsf{a}}, V_{\mathsf{b}}\}$, where a, b are the states of $\mathbf{G}_1$, such that

$$\forall (s, d, \sigma) \in E_1, \forall x \in \mathbb{R}^n : V_d\left(\frac{1}{\rho} f_\sigma(x)\right) \leq V_s(x). \tag{5}$$

Let us now consider the set of CLFs $\{U_{\mathsf{c}}, U_{\mathsf{d}}, U_{\mathsf{e}}\}$ for $\mathbf{G}_2$ where

$$U_{\mathsf{c}} = V_{\mathsf{a}},\; U_{\mathsf{d}} = U_{\mathsf{e}} = V_{\mathsf{b}}.$$

One can observe that for any edge $(s, d, \sigma) \in E_2$, the corresponding inequality, i.e.

$$\forall x \in \mathbb{R}^n : U_d\left(\frac{1}{\rho} f_\sigma(x)\right) \leq U_s(x),$$

is in fact equivalent to one of the inequalities in (5). For example, the inequality for the edge $(\mathsf{e}, \mathsf{d}, 3) \in E_2$ reads

$$\forall x \in \mathbb{R}^n : U_{\mathsf{d}}\left(\frac{1}{\rho} f_3(x)\right) = V_{\mathsf{b}}\left(\frac{1}{\rho} f_3(x)\right) \leq V_{\mathsf{b}}(x) = U_{\mathsf{e}}(x),$$

which corresponds to the inequality for the edge $(\mathsf{b}, \mathsf{b}, 3) \in E_1$ in (5). Whenever we can satisfy the inequalities (5) for $\mathbf{G}_1$, we can also satisfy the corresponding inequalities for $\mathbf{G}_2$ with the same system and template. Hence, we conclude that $\mathbf{G}_1 \leq \mathbf{G}_2$. □

The contribution of this paper is to provide a full characterization of the *general graph-ordering* (Definition 2.7) through the concept of *simulation*. This is the object of the next section.

3 A CHARACTERIZATION OF THE ORDERING PROPERTY VIA SIMULATION

This section is devoted to the presentation of our main result, Theorem 3.5. We begin in Subsection 3.1 by introducing and illustrating the concept of *simulation*, as well as Theorem 3.5 itself. Its proof is presented in Subsection 3.2, which is more technically oriented.

3.1 Ordering via Simulation

To our knowledge, the problem of deciding whether or not two graphs are ordered was introduced in [1, Section 4.2] together with ad-hoc techniques establishing the ordering in particular cases. Different techniques have since been introduced in [2, 3, 31], with the goal of capturing a notion of ordering for linear systems with quadratic CLFs. The main differences between our goal and those of these papers is that those papers explore sufficient conditions of ordering. In the present manuscript, we seek to go beyond that and provide necessary and sufficient conditions for ordering with respect to the *simulation* criterion.

Definition 3.1 (Simulation). *Consider two graphs* $\mathbf{G}_1 = (S_1, E_1)$ *and* $\mathbf{G}_2 = (S_2, E_2)$ *on a same language* $\mathscr{L}$. $\mathbf{G}_1$ simulates $\mathbf{G}_2$ *if there is a function*

$$R : S_2 \to S_1 : s_2 \to R(s_2)$$

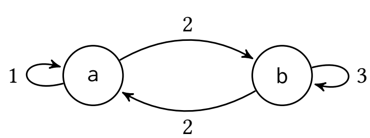
such that

$$\forall (s_2, d_2, \sigma) \in E_2 : \; (R(s_2), R(d_2), \sigma) \in E_1. \tag{6}$$

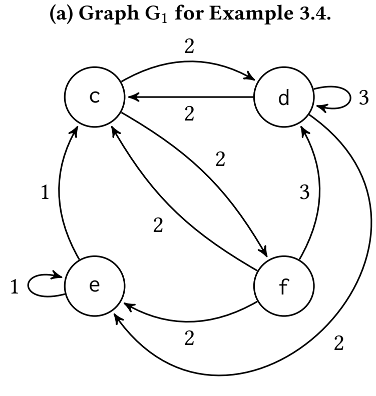
Remark 3.2. *The special case where the function R is bijective in Definition 3.1 corresponds to an isomorphism between* $\mathbf{G}_1$ *and* $\mathbf{G}_2$. *It is obvious that, in this case, for any family* $\mathcal{F}$ *and template* $\mathcal{V}$, $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$ *and* $\mathbf{G}_2 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_1$.

Example 3.3. *Consider the graphs on three modes* $\mathbf{G}_1 = (S_1, E_1)$ *and* $\mathbf{G}_2 = (S_2, E_2)$ *in Figures 1a and 1b respectively. These graphs are such that* $\mathbf{G}_1$ *simulates* $\mathbf{G}_2$. *Here, one can take a function* $R : S_2 \to S_1$ *where* $R(\mathsf{c}) = \mathsf{a}$, $R(\mathsf{d}) = R(\mathsf{e}) = \mathsf{b}$.

Example 3.4. *Consider the graphs on three modes* $\mathbf{G}_1 = (S_1, E_1)$ *and* $\mathbf{G}_2 = (S_2, E_2)$ *in Figures 2a and 2b respectively. These graphs are such that* $\mathbf{G}_1$ *simulates* $\mathbf{G}_2$. *Here, one can take a function* $R : S_2 \to S_1$ *where* $R(\mathsf{c}) = R(\mathsf{e}) = \mathsf{a}$, $R(\mathsf{d}) = R(\mathsf{f}) = \mathsf{b}$.



(a) Graph $\mathbf{G}_1$ for Example 3.4.



(b) Graph $\mathbf{G}_2$ for Example 3.4.

Figure 2: Two graphs such that $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$.

We are now in position to state our main theorem, which shows exactly to which extent the concepts of simulation (Definition 3.1) and the *general graph ordering* (Definition 2.7) are related to one another.

Theorem 3.5. *Consider two graphs* $\mathbf{G}_1 = (S_1, E_1)$ *and* $\mathbf{G}_2 = (S_2, E_2)$ *on a same language. The following are equivalent:*

(1) $\mathbf{G}_1$ *simulates* $\mathbf{G}_2$.
(2) $\mathbf{G}_1 \leq \mathbf{G}_2$ *in the sense of Definition 2.7.*

Theorem 3.5 tells us that *simulation* between graphs is equivalent to a broad notion of ordering, that holds for any family of systems or choice of Lyapunov function template.
From there, we understand that if a graph $\mathbf{G}_1$ does not simulate a graph $\mathbf{G}_2$, then there must be a family of switching systems $\mathcal{F}$ and a template $\mathcal{V}$ for which we have

$$\mathbf{G}_1 \not\leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$$

In other words, $\mathbf{G}_1$ *will be better suited* than $\mathbf{G}_2$ for the analysis of some particular families of systems using some particular Lyapunov function templates.
Note that when considering particular families $\mathcal{F}$ and templates $\mathcal{V}$, we may still have ordering without simulation. This is in particular true for linear systems using quadratic Lyapunov functions, for which more advanced ordering tools have been proposed in [3].

In the next section, we provide a proof of Theorem 3.5 alongside with an important intermediate result, namely Theorem 3.6.

3.2 Proving the main result

We start with a technical result which is central in the proof of Theorem 3.5. It states that if we are given a graph $\mathbf{G} = (S, E)$, there is no way *a priori* to add a new edge in E and thereby keep an equivalent PCLF criterion. Indeed, the theorem exhibits a system F such that one criterion would be satisfied and the other would not.

Theorem 3.6. *For any graph* $\mathbf{G} = (S, E)$ *on M modes, there exist an integer $n \geq 1$, a system F in dimension n, and $|S|$ CLFs $V_s, s \in S$ for which*

$$\forall (s, d, \sigma) \in E, \forall x \in \mathbb{R}^n : V_d(f_\sigma(x)) \leq V_s(x), \tag{7}$$

$$\forall (s, d, \sigma) \in \bar{E}, \exists \bar{x} \in \mathbb{R}^n : V_d(f_\sigma(\bar{x})) > V_s(\bar{x}), \tag{8}$$

where $\bar{E} = (S \times S \times \langle M \rangle) \backslash E$.

Proof. The proof is constructive. Given a graph $\mathbf{G}$, we build a switching system F and CLFs $\{V_s, s \in S\}$ satisfying (7) and (8).

The dimension of the system is $n = 2|\bar{E}|$. For each $\sigma \in \langle M \rangle$, we consider linear dynamics $f_\sigma(x) = A_\sigma x$ for $A \in \mathbb{R}^{n \times n}$.
The matrices A_σ are defined block-wise, with $|\bar{E}|$ blocks of 2×2 entries on the diagonal, and zeros everywhere else. For each $e = (s, d, \sigma) \in \bar{E}$, we set the diagonal block for e as

$$A_\sigma[e] = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

All other blocks contain only zeros.

For the CLFs, we consider functions of the form

$$V_s(x) = x^\top \cdot diag(v_s) \cdot x,$$

where $diag(v_s) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the vector $v_s \in \mathbb{R}^n_{>0}$ on the diagonal. We define the vectors v_s block-wise, each

block $v_s[e]$ indexed by an element $e \in \bar{E}$.
Remark that for this type of CLF, for any $(s_1, d_1, \sigma_1) \in (S \times S \times \langle M \rangle)$, we have

$$\forall x \in \mathbb{R}^n : V_{d_1}(A_{\sigma_1} x) \leq V_{s_1}(x) \Leftrightarrow \tag{9}$$

$$\forall e_2 = (s_2, d_2, \sigma_2) \in \bar{E} : v_{d_1}^\top [e_2] \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \leq v_{s_1}^\top [e_2]. \tag{10}$$

with the inequality taken entrywise. By positivity of the vectors v_s, inequality (10) is equivalent to having the second entry in $v_{d_1}[e_2]$ be smaller or equal than the first in $v_{s_1}[e_2]$. There is no constraint on the second entry of $v_{s_1}^\top$ because it is positive, and thus the corresponding inequality is automatically satisfied.
Note that for $e_2 = (s_2, d_2, \sigma_2) \in \bar{E}$ such that $\sigma_2 \neq \sigma_1$, $v_{d_1}^\top[e_2] A_{\sigma_1}[e_2] = 0$. Hence, for an edge $(s, d, \sigma) \in E$ corresponding to (14), we may concentrate on blocks corresponding to edges $(s_2, d_2, \sigma) \in \bar{E}$ sharing the same mode.

The blocks of the vector v_s, $s \in S$, are defined as follows for $e_2 = (s_2, d_2, \sigma_2) \in \bar{E}$:

- if $s = s_2 = d_2$ then $v_s^\top[e_2] = (3, 4)$;
- if $s = s_2$ and $s_2 \neq d_2$ then $v_s^\top[e_2] = (2, 1)$;
- if $s = d_2$ and $s_2 \neq d_2$ then $v_s^\top[e_2] = (4, 3)$;
- otherwise $v_s^\top[e_2] = (4, 1)$.

We now show that this construction satisfies the Theorem.
We begin with (7), equivalent to (10) holding for all $(s_1, d_1, \sigma) \in E$ here. Take any $e_1 = (s_1, d_1, \sigma_1) \in E$. We wish to show that (10) holds, that is, for any $e_2 = (s_2, d_2, \sigma_2) \in \bar{E}$, the second entry of $v_{d_1}[e_2]$ is smaller or equal to the first entry of $v_{s_1}[e_2]$.
If $\sigma_1 \neq \sigma_2$, by definition $A_{\sigma_1}[e_2] = 0$, hence (10) holds. We therefore focus on the case $\sigma_1 = \sigma_2$.

- If $s_1 = d_1$ then by definition $v_{s_1} = v_{d_1}$. From the definition of our vectors v_s above, (10) holds everywhere except if the second entry of $v_s[e_2]$ is larger than its first entry, which can occur only if $s_1 = s_2 = d_2$, where $v_{s_1}^\top[e_2] = (3, 4)$. However, this case cannot arise, since it would imply $e_1 = e_2$, impossible by definition because $e_1 \in E$ and $e_2 \in \bar{E}$.
- If $s_1 \neq d_1$, there are exactly 6 scenarios to consider as depicted in Table 1. Indeed, we are to consider whether or not $s_1 = s_2$, $s_2 = d_2$, and $d_1 = d_2$, which leads to a maximum of 8 cases. We discard the case when $s_1 = s_2 = d_1 = d_2$ since we focus on $s_1 \neq d_1$, and the case $s_1 = s_2 \neq d_2 = d_1$ is not to be considered either since it implies $e_1 = e_2$. As shown in Table 1, (10) holds in all relevant cases, and we conclude that (7) holds for the construction.

We now focus on (8), which in our case is equivalent to (10) *not holding* for $(s_1, d_1, \sigma) \in \bar{E}$. Take any $e_1 = (s_1, d_1, \sigma_1) \in \bar{E}$. We wish to show that there is $e_2 = (s_2, d_2, \sigma_2) \in \bar{E}$ for which (9) does not hold. We show we can pick $e_2 = e_1$ to achieve this.

- If $s_1 \neq d_1$, we have $v_{d_1}^\top[e_1] = (4, 3)$, hence $v_{d_1}^\top[e_1] A_{\sigma_1}[e_1] = (3, 0) \not\leq (2, 1) = v_{s_1}^\top[e_1]$.
- If $s_1 = d_1$, then $v_{s_1} = (3, 4)$, and $v_{s_1}^\top[e_1] A_{\sigma_1}[e_1] = (4, 0) \not\leq (3, 4)$.

This concludes the proof of Theorem 3.6. □

	$e_2 = (s_2, d_2, \sigma) \in \bar{E}$	$v_{s_1}^\top[e_2]$	$v_{d_1}^\top[e_2]$	$v_{d_1}^\top[e_2] A_\sigma[e_2]$
1:	$s_1 = s_2 = d_2 \neq d_1$	(3, 4)	(4, 1)	(1, 0)
2:	$s_1 = s_2 \neq d_2 \neq d_1$	(2, 1)	(4, 1)	(1, 0)
3:	$s_1 \neq s_2 = d_2 = d_1$	(4, 1)	(4, 1)	(1, 0)
4:	$s_1 \neq s_2 = d_2 \neq d_1$	(4, 1)	(3, 4)	(4, 0)
5:	$s_1 \neq s_2 \neq d_2 = d_1$	(4, 1)	(4, 3)	(3, 0)
6:	$s_1 \neq s_2 \neq d_2 \neq d_1$	(4, 1)	(4, 1)	(1, 0)

Table 1: The 6 scenarios to discuss when considering edges $e_1 = (s_1, d_1, \sigma) \in E$ where $s_1 \neq d_1$.

We are now in position to prove our main result, Theorem 3.5.

Proof of Theorem 3.5. **1.** $\Rightarrow$ **2.**: The direct implication has been proven in [3, Proposition 4.7]. We present here a proof in our slightly more general setting for the sake of completeness. We are given two graphs $\mathbf{G}_1 = (S_1, E_1)$, $\mathbf{G}_2 = (S_2, E_2)$ such that $\mathbf{G}_1$ simulates $\mathbf{G}_2$ through a function $R : S_2 \to S_1$.
Consider an arbitrary family of switching systems $\mathcal{F}$ with a state space $\mathbb{R}^n$, a system $F \in \mathcal{F}$ and a template of CLFs $\mathcal{V}$ as described in Section 2. From Definition 2.6 we have to show that $\rho_{\mathcal{V}}(F; \mathbf{G}_2) \leq \rho_{\mathcal{V}}(F; \mathbf{G}_1)$, for $\rho_{\mathcal{V}}(F; \mathbf{G}_i)$ as defined in (10).
To do so, take any $\rho > 0$, and assume that there is a set of CLFs $\{V_{s_1}, s_1 \in S_1\}$ such that

$$\forall (s_1, d_1, \sigma) \in E_1, \forall x \in \mathbb{R}^n : V_{d_1}\left(\frac{f_\sigma}{\rho}(x)\right) \leq V_{s_1}(x). \tag{11}$$

Recall that, by the definition of simulation, the function R satisfies $\forall (s_2, d_2, \sigma) \in E_2 : (R(s_2), R(d_2), \sigma) \in E_1$. Therefore, we can write

$$\forall (s_2, d_2, \sigma) \in E_2, \forall x \in \mathbb{R}^n : V_{R(d_2)}\left(\frac{f_\sigma}{\rho}(x)\right) \leq V_{R(s_2)}(x).$$

Hence, the set of CLFs $\{U_s = V_{R(s)}, s \in S_2\}$ satisfies

$$\forall (s_2, d_2, \sigma) \in E_2, \forall x \in \mathbb{R}^n : U_{d_2}\left(\frac{f_\sigma}{\rho}(x)\right) \leq U_{s_2}(x). \tag{12}$$

Since (12) holds for all ρ such that (11) holds, we conclude that for any $F \in \mathcal{F}$,

$$\rho_{\mathcal{V}}(F; \mathbf{G}_2) \leq \rho_{\mathcal{V}}(F; \mathbf{G}_1).$$

This being true for any arbitrary choice of $\mathcal{F}$ and $\mathcal{V}$, we conclude the first part of the proof.

2. $\Rightarrow$ **1.**: We now suppose that regardless of the family $\mathcal{F}$ and of the template $\mathcal{V}$, $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$, or equivalently for any $F \in \mathcal{F}$,

$$\rho_{\mathcal{V}}(F; \mathbf{G}_2) \leq \rho_{\mathcal{V}}(F; \mathbf{G}_1), \tag{13}$$

for $\rho_{\mathcal{V}}(F; \mathbf{G})$ in (3). This implies in particular that if there is a set of CLFs from the template $\mathcal{V}$ admissible for F and $\mathbf{G}_1$, an admissible set exists for F and $\mathbf{G}_2$. From there, we have to prove that $\mathbf{G}_1$ simulates $\mathbf{G}_2$, that is, we have to exhibit a function $R : S_2 \to S_1$ for any $(s_2, d_2, \sigma) \in E_2$, $(R(s_2), R(d_2), \sigma) \in E_1$.
To do so, we first apply Theorem 3.6 to $\mathbf{G}_1$ and obtain a (linear) switching system $F = \{f_\sigma, \sigma \in \langle M \rangle\}$ and CLFs $\{V_{s_1}, s_1 \in S_1\}$ such that the following holds:

$$\forall (s_1, d_1, \sigma_1) \in E_1, \forall x \in \mathbb{R}^n : V_d(f_\sigma(x)) \leq V_s(x), \tag{14}$$

$$\forall (s_1, d_1, \sigma_1) \in \bar{E}_1, \exists \bar{x} \in \mathbb{R}^n : V_d(f_\sigma(\bar{x})) > V_s(\bar{x}), \tag{15}$$

where $\bar{E}_1 = (S_1 \times S_1 \times \langle M \rangle) \backslash E_1$.
Second, we consider the particular family $\mathcal{F} = \{F\}$ and template $\mathcal{V} = \{V_s, s \in S_1\}$, for the ordering relation $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$. That is, our new template is the finite set of CLFs *that satisfy* (14), (15) *for* $\mathbf{G}_1$. We aim at exhibiting a simulation between $\mathbf{G}_1$ and $\mathbf{G}_2$ using these.
Since the template $\mathcal{V}$ is restricted to a finite set of CLFs, each of which being part of the solution to (14) and (15), there is a natural way of building a relation between the nodes of $\mathbf{G}_2$ and the nodes of $\mathbf{G}_1$. Indeed, any admissible set of CLFs for $\mathbf{G}_2$ and F can be expressed as

$$\{U_{s_2} = V_{R(s_2)},\ s_2 \in S_2\};$$

for some function $R : S_2 \to S_1$. For this, for any node s_2 of G_2 just select an arbitrary node $R(s_2)$ of G_1 such that the above equation is satisfied. Since our template only contains functions that are part of the solution for G_1, there must be one node $R(s_2)$ satisfying this. Hence, we can write

$$\forall (s_2, d_2, \sigma) \in E_2, \forall x \in \mathbb{R}^n : V_{R(d_2)}(f_\sigma(x)) \leq V_{R(s_2)}(x).$$

Furthermore, we claim that such a function R has to satisfy the condition

$$(s_2, d_2, \sigma) \in E_2 \Rightarrow (R(s_2), R(d_2), \sigma) \in E_1,$$

thus establishing the fact that $\mathbf{G}_1$ simulates $\mathbf{G}_2$. Indeed, assume by contradiction that there is $(s_2, d_2, \sigma) \in E_2$ such that $(R(s_2), R(d_2), \sigma) \notin E_1$. Then, by (15), we have that

$$\exists \bar{x} \in \mathbb{R}^n : V_{R(d_2)}(f_\sigma(\bar{x})) = U_{d_2}(f_\sigma(\bar{x})) > U_{s_2}(\bar{x}) = V_{R(s_2)}(\bar{x}),$$

and the set $U_s, s \in S_2$ cannot be admissible in the first place.
In conclusion, we have shown that if the ordering $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}}$ holds for any family $\mathcal{F}$ and template $\mathcal{V}$, then it must be so that $\mathbf{G}_1$ simulates $\mathbf{G}_2$. □

3.3 Beyond simulation

As discussed in Subsection 3.1, there can be particular families $\mathcal{F}$ and templates $\mathcal{V}$ such that $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$ when $\mathbf{G}_1$ does not simulate $\mathbf{G}_2$. This subsection discusses such cases through examples.

Let us begin with some simple observations that highlight the importance of the family of systems $\mathcal{F}$ and template $\mathcal{V}$ considered when discussing ordering.

Example 3.7.

- *Consider the family $\mathcal{F}$ of switching systems which map any state to the origin in 1 step (i.e., $\forall x, \forall F \in \mathcal{F}, \forall f_\sigma \in F, f_\sigma(x) = 0$). It is trivial to show that for any pair of graph $\mathbf{G}_1$ and $\mathbf{G}_2$ and any template $\mathcal{V}$, $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$.*
- *Consider a CLF template of the form $\mathcal{V} = \{V\}$, i.e., that contains a unique CLF. One can show that for any pair of graphs $\mathbf{G}_1$ and $\mathbf{G}_2$ on M modes and any family $\mathcal{F}$, $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$.*

Clearly, in the simple cases above, the concept of simulation does not play a role in ordering.

In the next example, we present ordering phenomena that arise when the template $\mathcal{V}$ has particular algebraic structures.

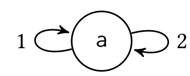
Example 3.8. *Consider the graphs $\mathbf{G}_1 = (S_1, E_1)$ and $\mathbf{G}_2 = (S_2, E_2)$ in Figure 3. It is easy to see that $\mathbf{G}_1$ simulates $\mathbf{G}_2$ with the function $R : S_2 \to S_1$ such that $R(\mathsf{b}) = R(\mathsf{c}) = \mathsf{a}$. Therefore for any $\mathcal{F}$, $\mathcal{V}$, $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$.*
Of course, the fact that $\mathbf{G}_1$ produces more conservative certificates than $\mathbf{G}_2$ is intuitive: after-all, $\mathbf{G}_2$ does have more nodes, and hence a PCLF on $\mathbf{G}_2$ seems inherently more complex. Moreover, $\mathbf{G}_2$ does not simulate $\mathbf{G}_1$, and as a consequence of Theorem 3.5, there are systems $\mathcal{F}$ and templates $\mathcal{V}$, such that $\mathbf{G}_2$ performs better than $\mathbf{G}_1$.
However, if we take $\mathcal{V}$ to be e.g. the set of quadratic positive definite function, then we can show that $\mathbf{G}_2 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_1$ holds for any family $\mathcal{F}$ as well. Indeed, one can check that whenever a set of quadratic functions $\{U_\mathsf{b}, U_\mathsf{c}\}$ is admissible for $\mathbf{G}_2$, then $\{V_\mathsf{a} = U_\mathsf{b} + U_\mathsf{c}\}$ is admissible for $\mathbf{G}_1$. Indeed, for example, for the edge $(\mathsf{a}, \mathsf{a}, 1) \in E_1$, we see that

$$\forall x \in \mathbb{R}^n : (U_\mathsf{b} + U_\mathsf{c})(f_1(x)) \leq (U_\mathsf{b} + U_\mathsf{c})(x)$$

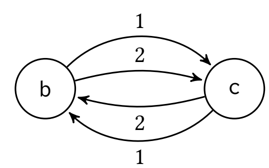
because in $\mathbf{G}_2$, from the edges $(\mathsf{b}, \mathsf{c}, 1) \in E_2$, $(\mathsf{c}, \mathsf{b}, 1) \in E_2$, we have

$$\forall x \in \mathbb{R}^n : U_\mathsf{c}(f_1(x)) \leq U_\mathsf{b}(x),$$
$$\forall x \in \mathbb{R}^n : U_\mathsf{b}(f_1(x)) \leq U_\mathsf{c}(x),$$

and summing these inequalities leads to the desired result.



(a) Graph $\mathbf{G}_1$ for Example 3.8.



(b) Graph $\mathbf{G}_2$ for Example 3.8.

Figure 3: An example where the simulation criterion applies. $\mathbf{G}_1$ simulates $\mathbf{G}_2$, hence $\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2$ for any family $\mathcal{F}$ and template $\mathcal{V}$. For the template $\mathcal{V}$ of quadratic positive definite functions, $\mathbf{G}_2 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_1$ as well.

Our last example involves properties of both the family $\mathcal{F}$ and template $\mathcal{V}$ considered.

Example 3.9. *Consider the three graphs $\mathbf{G}_1 = (S_1, E_1)$ and $\mathbf{G}_2 = (S_2, E_2)$ in Figures 4a and 4b respectively. One can easily verify that no simulation exists between $\mathbf{G}_1$ and $\mathbf{G}_2$.*
Nevertheless, for the family $\mathcal{F}$ of linear switching systems *(i.e., such that $f_\sigma(x) = A_\sigma x$), where the dynamics are* invertible *(i.e. the matrices A_σ are* regular*), we can show that for any template $\mathcal{V}$* closed under composition with a linear function *(i.e., if $V \in \mathcal{V}$, $V \circ A \in \mathcal{V}$ for any regular matrix $A \in \mathbb{R}^{n \times n}$):*

$$\mathbf{G}_1 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_2, \tag{16}$$
$$\mathbf{G}_2 \leq_{\mathcal{F},\mathcal{V}} \mathbf{G}_1. \tag{17}$$

Indeed, for the first ordering relation (16)*, for any switching system F, let $\{V_\mathsf{b}, V_\mathsf{c}\}$ be an admissible set of CLFs for the graph $\mathbf{G}_1$. Then, one can show (see e.g. [30, Example 6.1]) that the set*

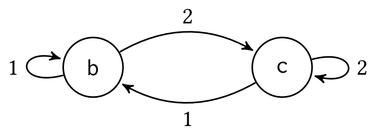
$$\{U_{\mathsf{b}'} = V_\mathsf{b} \circ f_1,\ U_{\mathsf{c}'} = V_\mathsf{c} \circ f_2\}$$

is admissible for the same system F *and the graph* $\mathbf{G}_2$*, conditional to the fact that* $V_{\mathsf{b}} \circ f_1$ *and* $V_{\mathsf{c}} \circ f_2$ *belong to* $\mathcal{V}$*. In particular, for e.g.* $(\mathsf{b}', \mathsf{c}', 1) \in E_2$*, we have*

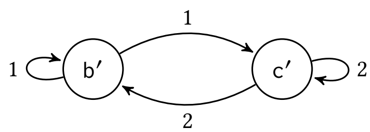
$$\forall x \in \mathbb{R}^n : U_{\mathsf{c}'}(f_1(x)) = V_{\mathsf{c}}(f_1(x)) \leq V_{\mathsf{b}}(x) = U_{\mathsf{b}}(x),$$

which holds true since $\{V_{\mathsf{b}}, V_{\mathsf{c}}\}$ *is admissible for* $\mathbf{G}_1$ *and* $(\mathsf{b}, \mathsf{c}, 1) \in E_1$*. A similar argument can be used for the second ordering relation* (17)*, see [30, Example 6.1].*

In conclusion, if one restricts the choices of $\mathcal{F}$ *and* $\mathcal{V}$ *to particular sets, the characterization of Theorem 3.5 is not valid anymore, since the ordering relation is not anymore equivalent to the simulation property.*



(a) $\mathbf{G}_1$ **for Example 3.9.**



(b) Graph $\mathbf{G}_2$ **for Example 3.9. This graph does not satisfy the simulation criterion with** $\mathbf{G}_1$ **at Figure 4a, but yet they are ordered with respect to the particular template of quadratic Lyapunov functions.**

Figure 4: Graphs for Example 3.9.

4 CONCLUSION

Path-Complete methods provide powerful tools to analyze key aspects of switching and hybrid systems.

In this paper, we study the performance of Path-Complete methods for stability analysis. More precisely, we seek to order these certificates by answering, for each pair, the question *"is one less conservative than the other in general?"*.

Our main result, Theorem 3.5, provides a complete characterization of this ordering: a PCLF criterion is less conservative than a second one, regardless of the system under study and the template of Lyapunov functions used, if and only if the graph of the first one simulates that of the second one.

We believe that this result, together with our second result Theorem 3.6, is a key step for understanding the Path-Complete framework as a whole. The natural continuation of this work is to further characterize the ordering of PCLFs for specific families of switching systems and templates of Lyapunov functions. Section 3.3 highlights several simple examples showing that the comparison problem becomes richer in such more specific frameworks.

A first step in this direction would be to further study the criteria introduced in [2, 3, 31] for the case of linear switching systems and quadratic Lyapunov functions.

REFERENCES

[1] Amir Ali Ahmadi, Raphaël M Jungers, Pablo A Parrilo, and Mardavij Roozbehani. 2014. Joint Spectral Radius and Path-Complete Graph Lyapunov Functions. *SIAM Journal on Control and Optimization* 52, 1 (2014), 687–717.

[2] David Angeli, Nikolaos Athanasopoulos, Raphaël M Jungers, and Matthew Philippe. 2017. A linear program to compare path-complete Lyapunov functions. In *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 5888–5893.

[3] David Angeli, Nikolaos Athanasopoulos, Raphaël M Jungers, and Matthew Philippe. 2017. Path-complete graphs and common lyapunov functions. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*. ACM, 81–90.

[4] Nikolaos Athanasopoulos and Raphaël M Jungers. 2018. Combinatorial methods for invariance and safety of hybrid systems. *Automatica* 98 (2018), 130–140.

[5] Nikolaos Athanasopoulos and Raphaël M Jungers. 2018. On invariance and reachability on semialgebraic sets for linear dynamics. *IFAC-PapersOnLine* 51, 16 (2018), 43–48.

[6] Guillaume O. Berger, Fulvio Forni, and Raphaël M Jungers. 2018. Path-complete p-dominant switching linear systems. *arXiv preprint arXiv:1808.09757* (2018).

[7] Pierre-Alexandre Bliman and Giancarlo Ferrari-Trecate. 2003. Stability analysis of discrete-time switched systems through Lyapunov functions with nonminimal state. In *Proceedings of IFAC Conference on the Analysis and Design of Hybrid Systems*. 325–330.

[8] Vincent D. Blondel and John N. Tsitsiklis. 2000. The boundedness of all products of a pair of matrices is undecidable. *Systems & Control Letters* 41, 2 (2000), 135–140.

[9] Michael S. Branicky. 1998. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Automat. Control* 43, 4 (1998), 475–482.

[10] Christos G. Cassandras and Stephane Lafortune. 2009. *Introduction to discrete event systems*. Springer Science & Business Media.

[11] Jamal Daafouz and Jacques Bernussou. 2001. Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems & Control Letters* 43, 5 (2001), 355–359.

[12] M.C.F. Donkers, WPMH Heemels, Nathan Van de Wouw, and Laurentiu Hetel. 2011. Stability Analysis of Networked Systems Using a Switched Linear Systems Approach. *IEEE Trans. Automat. Control* 56 (2011), 2101–2115.

[13] Ray Essick, Ji-Woong Lee, and Geir E. Dullerud. 2014. Control of linear switched systems with receding horizon modal information. *IEEE Trans. Automat. Control* 59, 9 (2014), 2340–2352.

[14] Ray Essick, Matthew Philippe, Geir Dullerud, and Raphael M. Jungers. 2015. The minimum achievable stability radius of switched linear systems with feedback. In *IEEE 54th Annual Conference on Decision and Control (CDC)*. IEEE, 4240–4245.

[15] Antoine Girard and George J. Pappas. 2011. Approximate Bisimulation: A Bridge Between Computer Science and Control Theory. *European Journal of Control* 17 (2011), 568–578.

[16] Esteban A. Hernandez-Vargas, Richard H. Middleton, and Patrizio Colaneri. 2011. Optimal and MPC switching strategies for mitigating viral mutation and escape. In *IFAC World Congress*. 14857–14862.

[17] Laurentiu Hetel, Jamal Daafouz, and Claude Iung. 2008. Equivalence between the Lyapunov–Krasovskii functionals approach for discrete delay systems and that of the stability conditions for switched systems. *Nonlinear Analysis: Hybrid Systems* 2, 3 (2008), 697–705.

[18] Mikael Johansson and Anders Rantzer. 1998. Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE transactions on automatic control* 43, 4 (1998), 555–559.

[19] Marc Jungers, Antoine Girard, and Mirko Fiacchini. 2018. Language constrained stabilization of discrete-time switched linear systems: an LMI approach. *IFAC-PapersOnLine* 51, 16 (2018), 25–30.

[20] Raphaël Jungers. 2009. The joint spectral radius. *Lecture Notes in Control and Information Sciences* 385 (2009).

[21] Victor Kozyakin. 2014. The Berger–Wang formula for the Markovian joint spectral radius. *Linear Algebra Appl.* 448 (2014), 315–328.

[22] Ji-Woong Lee and Geir E. Dullerud. 2006. Optimal disturbance attenuation for discrete-time switched and Markovian jump linear systems. *SIAM Journal on Control and Optimization* 45, 4 (2006), 1329–1358.

[23] Ji-Woong Lee and Geir E. Dullerud. 2006. Uniform Stabilization of discrete-time switched and Markovian jump Linear systems. *Automatica* 42 (2006), 205–218.

[24] Daniel Liberzon. 2012. *Switching in systems and control*. Springer Science & Business Media.

[25] Daniel Liberzon and Stephen A. Morse. 1999. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine* 19, 5 (1999), 59–70.

[26] Hai Lin and Panos J. Antsaklis. 2009. Stability and stabilizability of switched linear systems: a survey of recent results. *IEEE Transactions on Automatic control* 54, 2 (2009), 308–322.

[27] Sébastien Mariéthoz, Stefan Almér, Mihai Bâja, Andrea Giovanni Beccuti, Diego Patino, Andreas Wernrud, Jean Buisson, Hervé Cormerais, Tobias Geyer, Hisaya

Fujioka, Ulf T. Jönsson, Chung-Yao Kao, Manfred Morari, Georgios Papafotiou, Anders Rantzer, and Pierre Riedinger. 2010. Comparison of Hybrid Control Techniques for Buck and boost DC-DC Converters. *IEEE Transactions on Control Systems Technology* 18 (2010), 1126–1145.

[28] Pablo A. Parrilo and Ali Jadbabaie. 2008. Approximation of the joint spectral radius using sum of squares. *Linear Algebra Appl.* 428, 10 (2008), 2385–2402.

[29] Pierdomenico Pepe. 2018. Converse Lyapunov Theorems for Discrete-Time Switching Systems with Given Switches Digraphs. *IEEE Trans. Automat. Control* (2018).

[30] Matthew Philippe. 2017. *Path-Complete Methods and Analysis of Constrained Switching Systems*. Ph.D. Dissertation.

[31] Matthew Philippe, Nikolaos Athanasopoulos, David Angeli, and Raphaël M Jungers. 2017. On path-complete Lyapunov functions: geometry and comparison. *To appear in IEEE Transactions on Automatic Control* (2017).

[32] Matthew Philippe, Ray Essick, Geir E. Dullerud, and Raphaël M. Jungers. 2016. Extremal storage functions and minimal realizations of discrete-time linear switching systems. In *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*. 5533–5538.

[33] Matthew Philippe, Ray Essick, Geir E. Dullerud, and Raphaël M. Jungers. 2016. Stability of discrete-time switching systems with constrained switching sequences. *Automatica* 72 (2016), 242–250.

[34] Matthew Philippe and Raphaël M. Jungers. 2015. A sufficient condition for the boundedness of matrix products accepted by an automaton. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*. ACM, 51–57.

[35] Matthew Philippe, Gilles Millérioux, and Raphaël M Jungers. 2017. Deciding the boundedness and dead-beat stability of constrained switching systems. *Nonlinear Analysis: Hybrid Systems* 23 (2017), 287–299.

[36] Robert Shorten, Fabian Wirth, and Douglas Leith. 2006. A positive systems model of TCP-like congestion control: asymptotic results. *IEEE/ACM Transactions on Networking* 14, 3 (2006), 616–629.

[37] Robert Shorten, Fabian Wirth, Oliver Mason, Kai Wulff, and Christopher King. 2007. Stability criteria for switched and hybrid systems. *SIAM review* 49, 4 (2007), 545–592.