UCLouvain

École polytechnique de Louvain

Learning stability guarantees for black-box hybrid systems

From arbitrary to constrained switching linear systems, a step towards complexity

Author: **Adrien Banse**
Supervisor: **Raphaël M. Jungers**
Readers: **Julien Hendrickx, Zheming Wang**
Academic year 2021–2022
Master [120] in Mathematical Engineering

Acknowledgements

My very first thanks goes to my supervisor, Raphaël M. Jungers. Prof. Jungers is an extremely motivating supervisor, and his interest and ambition are contagious. I am very grateful for the confidence he has had in me since the beginning of our exchanges, more than a year ago. I would also like to thank him for his availability, his enthusiasm, and his investment in the development of this thesis. I look forward to working with him and his team again.

I would also like to thank Zheming Wang. I asked him countless questions, and he always made himself available to answer them with kindness. I could never have written this thesis and the publications we submitted without his help and support. Zheming is undoubtedly a brilliant researcher, and I wish him much success in the future. I would also like to thank Julien Hendrickx for the time he spends reading this thesis.

A thought finally goes to my friends and family, who supported me in exam periods and during these five years. I especially thank my parents, nothing would be possible without their support; Emilie and Valentine, our "blocuses" at our grandparents' house contributed to the completion of my studies; Marine, we worked together on at least 11 courses (I counted!); and last but not least Adèle, for her love, her unfailing support and her precious advice.

Abstract

Hybrid systems are dynamical systems whose dynamics is characterized by continuous and discrete behaviours. In particular, *switching linear systems* are an important family of hybrid systems in which a *switching rule* plays a critical role. Consider a linear system whose controller is physically separated from its plant, and where the feedback has to go via a network. If the network fails, this linear system *switches* from an active to a failing status.

Although this hybrid behaviour makes their range of application extensive, switching linear systems are notoriously hard to analyze. However, they often generate rich data (harvested from devices such as cameras, lidars, etc.), to which the engineer has access in large quantities. This motivates a global research effort in developing *data-driven* tools to analyze such complex systems.

Recently, data-driven stability analysis techniques were developed for a subcategory of switching linear systems, where the switching rule is arbitrary. In this thesis, we take a step towards complexity by generalizing these results to a more general framework, where we consider that the switching rule is subject to logical rules. These systems are called *constrained switching linear systems*.

We leverage existing data-driven approaches and combine them with specific tools for analyzing the stability of these systems, such as *lifting* techniques and *multiple Lyapunov functions*. Using these concepts, we first demonstrate that we can derive probabilistic guarantees for the stability of constrained switching linear systems. We investigate two different approaches, and compare them with multiple examples. Second, we provide algorithmic tools to compute these guarantees.

Contents

Notations

Basics

$\mathbb{R}$ ..... Real numbers
$\mathbb{R}_+$ ..... Non-negative real numbers
$\mathbb{R}_{++}$ ..... Positive real numbers
$\mathbb{Z}$ ..... Integer numbers
$\mathbb{N}$ ..... Natural numbers
$[m]$ ..... Interval of natural numbers $\{1, \dots, m\}$
X^t ..... Set of all possible sequence $(x_n)_{n=1}^t$, $x \in X$
$X^{\mathbb{N}}$ ..... Set of all possible infinite sequence $(x_n)_{n\in\mathbb{N}}$, $x \in X$
$f : X \to Y : x \mapsto f(x)$ ..... f has domain X, codomain Y, and maps x to $f(x)$

Linear algebra

$\mathbb{R}^n$ ..... Vectors of dimensions n
$\mathbb{S}$ ..... Unit sphere $\{x \in \mathbb{R}^n : \|x\| = 1\}$
$\mathbb{R}^{n\times n}$ ..... Matrices of dimensions $n \times n$
$A \succ 0$, $A \succeq 0$ ..... A is positive definite, A is positive semi-definite
$\lambda_{\min}(A), \lambda_{\max}(A)$ ..... Minimal and maximal eigenvalues of matrix A
$\det(A)$ ..... Determinant of matrix A
$\mathcal{S}^n$ ..... Symmetric positive semi-definite $n \times n$ matrices
I_n ..... Identity $n \times n$ matrix

Automata theory

$\mathbf{G}(V, E)$ ..... Automaton $\mathbf{G}$ with set of nodes V and set of edges E
$\mathcal{L}_{\mathbf{G},l}$ ..... Set of all words of length l accepted by $\mathbf{G}$
$\mathcal{L}_{\mathbf{G}}$ ..... Set of all words accepted by $\mathbf{G}$, $\mathcal{L}_{\mathbf{G}} = \cup_{l\in\mathbb{N}}\mathcal{L}_{\mathbf{G},l}$

Acronyms

ASLS ..... Arbitrary switching linear system
CSLS ..... Constrained switching linear system
JSR ..... Joint spectral radius

CJSR ..... Constrained joint spectral radius
CQLF ..... Common Lyapunov function
MQLF ..... Multiple Lyapunov function

List of Figures

List of Tables

Introduction

Due to major technological upheavals, **the complexity of many dynamical systems has dramatically increased** in recent years. As examples, think of systems such as Smart Energy Grids, autonomous vehicles, Smart Cities, Networked Control Systems, embedded robots, etc. The academic community has coined this paradigm shift under the name of the *Cyber-Physical revolution* (see [Lee15, HJT12, KK12, Alu15, LS16]). In particular, *hybrid systems* are dynamical systems whose dynamics are characterized by continuous and discrete behaviours. These systems often appear in promising Cyber-Physical application, placing their analysis and control at the center of a global, multidisciplinary and extensive research effort [CL08, Tab09, GST12]. **The natural complexity of hybrid systems prevents the engineer from using classical control and analysis techniques**. Moreover, it has been proven that even in simple settings, some natural control problems are undecidable, NP-hard or with no algebraic solution [TB97, BT00a, BT00b]. In this thesis, we are interested in a particular type of hybrid systems, namely *switching systems*.

Switching systems [Jun09, LM99, SG, JP11, Lib03] are dynamical systems in which a *switching rule* plays a non-trivial role. **A switching system is a two-level hybrid system.** The first level drives the dynamics of the system, and depends on a *mode*. The second level, the coordinator, decides the *rules* of switching among the modes. More precisely, *switching linear systems* have been of great interest from the research community. Many tools have been developed to analyze their stability [Jun09, RG60, AJPR14, DRI02, LD06, PJ08].

Many model-based analysis concepts and techniques, such as the *joint spectral radius* have been applied to **more general models, on which the *switching signal* is subject to various constraints** [Dai11, PJ14, PEDJ16]. We call these systems *constrained switching linear systems* (*CSLS*s), in contrast to *arbitrary switching linear systems* (*ASLS*s), where the switching rule is not constrained. As a lot of constrained switching systems do not possess a *common Lyapunov function*, **the notion of *mutliple Lyapunov function* has been shown useful to analyze stability of such systems** [YMH98, Bra98, LA09]. This notion will be central in this thesis. Networked Controlled Systems, for example, can be modelled as constrained switching systems [DHvdWH11, JHK16]. An illustration is provided in Figure 1.

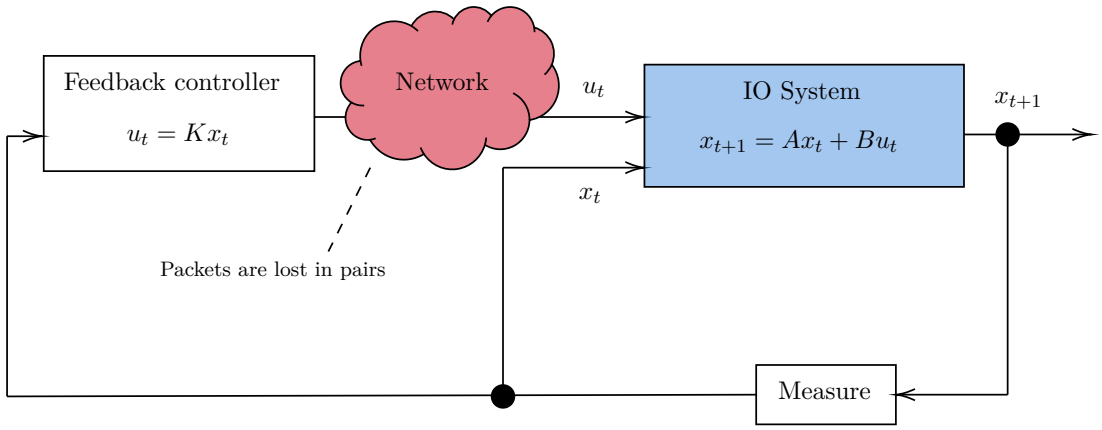


Figure 1: Modelling of a Networked Control System as a constrained switching linear system. Suppose one wants to stabilize a system $x_{t+1} = Ax_t + Bu_t$. Assume A is instable. We implement a linear state feedback $u_t = Kx_t$. However, the controller is physically separated from the plant in this application, and the feedback has to be communicated to the system via a network. Suppose that this network can fail. If the packet is correctly sent, the system satisfies $x_{t+1} = (A+BK)x_t$, otherwise it satisfies $x_{t+1} = Ax_t$. Moreover, suppose that if a packet is lost, then the following packet is also lost, so that packets are lost by group of 2. The latter is the switching rule. The IO system drives the dynamics, it is the first level of the switching system. The network drives the switching rule, it it the second level of the dynamical system.

In many practical applications, **the engineer cannot rely on having a model**, but rather has to **analyze the underlying system in a *data-driven* fashion**. In this case, researchers talk about *black-box* system, in opposition to model-based, *white-box* systems. Most classical data-driven method (e.g. [KK17, HGGL98, CLS03]) are limited to linear systems and rely on classical identification and frequency-domain approaches. These methods may not be well suited for Cyber-Physical systems because of the natural complexity of the latter. Novel data-driven stability analysis methods [KBJT19, BJW21, RWJ21] have been recently develop based on *scenario optimization* [CG16, CG18].

Our work, and our findings, follow the previous work of [KBJT19, BJW21, RWJ21]. Inspired by generalization to constrained switching linear systems such as [Dai11, PJ14, PEDJ16], **we will provide data-driven stability analysis techniques in a more general framework**, i.e. for constrained systems. More precisely, we will answer the following question: **from a finite set of observations, is one able to derive mathematical probabilistic guarantees for the stability of a constrained system?**

A summary of the state of the art for stability analysis of switching linear systems, as

well as the place of our contribution takes is provided in Table 1.

	ASLS	CSLS
White-box	[Jun09, RG60, DRI02, LD06, PJ08]	[Dai11, PJ14, PEDJ16]
Black-box	[KBJT19, BJW21, RWJ21]	**[BWJ22a, BWJ22b]**

Table 1: Non-exhaustive list of the state of the art for model-based and data-driven stability analysis of switching linear systems. In this thesis, we answer the question of the stability analysis of black-box CSLSs. Our contributions are the subject of two publications [BWJ22a, BWJ22b].

Outline and contributions

Our work takes place in the context of the Cyber-Physical revolution. Our goals are twofold. First of all, our contribution is theoretical. We show that it is possible to generalize recently obtained data-driven results on ASLSs to a more general framework. In this context, we develop two methods providing probabilistic stability certificates for CSLSs. Our goal is also to provide practical tools for the engineer to analyze such systems. We thus provide in addition to this document software tools to compute these guarantees. These tools[1] were implemented in the promising programming language `Julia` [BKSE12]. The outline is presented in Figure 2. We detail it hereinafter.

Part I collects all necessary existing results about ASLSs and CSLSs. In Chapter 1, main formal definitions and basic results about stability of switching linear systems are provided. In particular, central concepts from Lyapunov theory are introduced, such as *common Lyapunov quadratic functions* and *multiple Lyapunov Functions* (*CQLF*s and *MQLF*s for short), respectively for stability analysis of ASLSs and CSLSs. In Chapter 2, we present existing methods providing data-driven stability guarantees for ASLSs. After formally stating the problem and the setting for ASLSs in Section 2.1, we present results based on the *scenario approach* [KBJT19, BJW21] in Section 2.2. Then, another method based on a *sensitivity analysis approach* [RWJ21] is presented in Section 2.3.

Our main theoretical findings are stated in Part II. We develop two methods to derive probabilistic stability guarantees for CSLS:

- We propose a *lifting* approach in Chapter 3. As explained in Section 3.2, lifting CSLSs to simpler ASLSs allows to reduce the computation of guarantees for CSLSs to a simpler problem, for which data-driven results already exist. In this context, we can use a powerful tool for stability analysis of ASLSs, namely **common Lyapunov functions**, and more precisely CQLFs. We build our method on top of existing results. More precisely, we generalize results based on the scenario approach presented in Section 2.2. A general theorem is presented in Section 3.2.
- In Chapter 4, we propose another method which requires a slight change of setting. We assume that harvested data provides more information than in the first case.

[1] Algorithmic tools are available at `https://github.com/adrienbanse/DataDrivenCSLS.jl`.

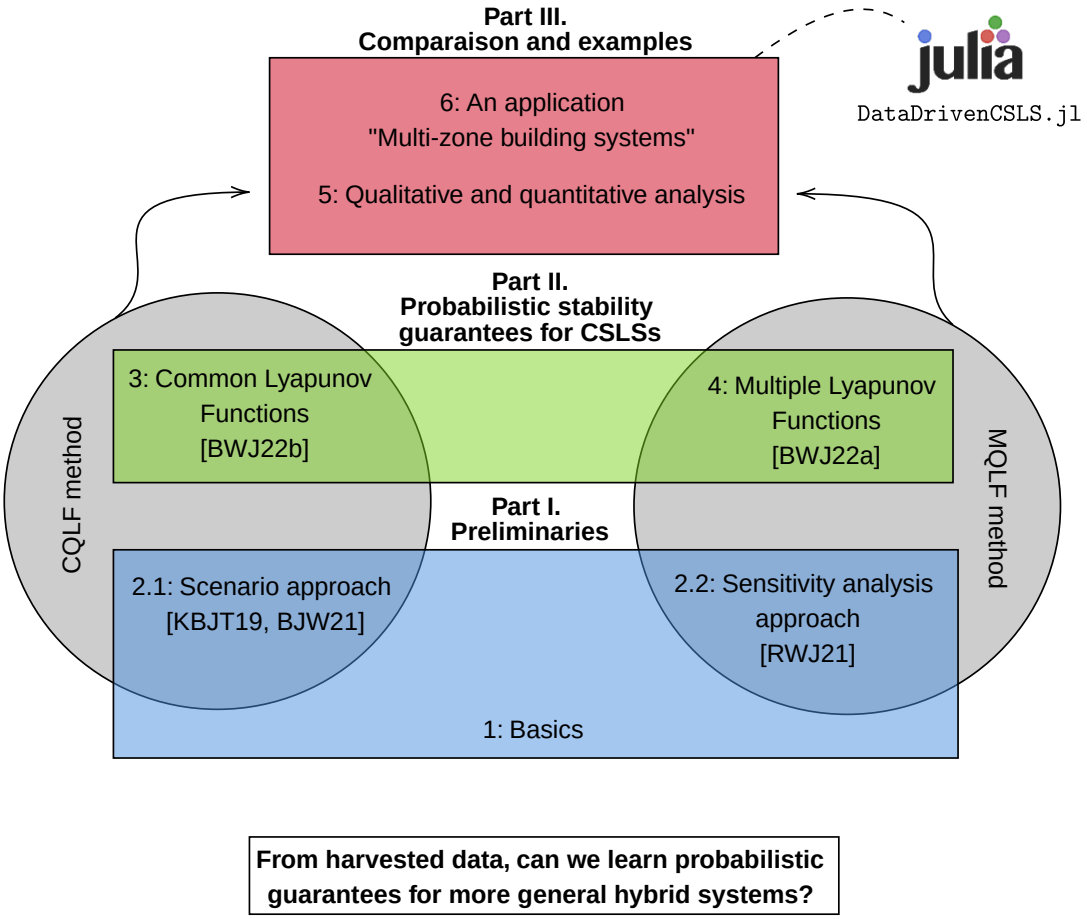


Figure 2: Thesis outline from bottom to top. The thesis is divided into three main parts. Existing model-based results about ASLSs and CSLSs are stated in Part I, as well as existing data-driven results for ASLSs. The main theoretical contributions can be found in Part II, when methods for CSLSs are described. Part III contains qualitative and quantitative comparaisons between methods, as well as a concrete example. A `Julia` toolbox detailed in Appendix B is used to obtain Part III results.

However, since this method does not require to lift a CSLS to an ASLS, it is not required for the considered CSLS to possess a common Lyapunov function anymore. The latter observation thus makes this method less conservative. Indeed, we leverage the sensitivity analysis approach introduced in Section 2.3 to **multiple Lyapunov functions**. We first present the problem setting in Section 4.1. We then show in Section 4.2 that, before any probabilistic guarantee on the stability of a CSLS, our method can provide deterministic instability guarantees. In Section 4.3 and Section 4.4, we show that we can generalize results of [RWJ21] to CSLSs.

Finally, Part III collects comparisons between both methods, as well as an applicative example, namely *multi-zone building systems* [FLC16, CFM+19, BSH21]. In Chapter 5, we provide qualitative and quantitative comparisons between the two methods. In particular, we first provide a theoretical comparison in Section 5.1. Then, we verify these assertions numerically in Section 5.2. We also show how to tune the parameters of our technique and study their impact on bounds introduced in Part II. Finally, in Chapter 6, we apply our methods to a specific multi-zone building system. We first define the problem, and how it can be modeled with a CSLS in Section 6.1. We finally provide a data-driven stability analysis of it in Section 6.2, using methods developed in this thesis. Part III contains simulations. They all are performed with the `Julia` toolbox implemented for this purpose. An overview of the code is available in Appendix B.

Publications

- Adrien Banse, Zheming Wang, and Raphaël M. Jungers. Learning stability guarantees for data-driven constrained switching linear systems. *25th International Symposium on Mathematical Theory of Networks and Systems* (Bayreuth, Germany). Submitted.
- Adrien Banse, Zheming Wang, and Raphaël M. Jungers. Learning stability guarantees for data-driven constrained switching linear systems. *41st Benelux Meeting on Systems and Control* (Brussels, Belgium). Accepted.
- Adrien Banse, Zheming Wang, and Raphaël M. Jungers. Black-box stability analysis of hybrid systems with sample-based multiple Lyapunov functions. *61st IEEE Conference on Decision and Control* (Cancùn, Mexico). Submitted.

Part I

Preliminaries

Chapter 1

Basics

In this chapter, we introduce the concepts needed to tackle white-box stability analysis of *switching linear systems*.

First, we give a definition of a switching linear system, as well as a definition of its stability. Then, we formally define ASLSs, where the switching sequence is unconstrained. We introduce the *joint spectral radius* (*JSR* for short), a quantity characterizing the stability of ASLSs [RG60]. We explain how it is related to *Lyapunov theory*, and we present a model-based method using *common quadratic Lyapunov functions* to approximate the JSR of an ASLS [Jun09]. Finally, we define CSLSs, where concepts from *automata theory* are needed [LM95]. In the same way as for ASLSs, we introduce the *constrained joint spectral radius* (*CJSR* for short) [Dai11], and we present a model-based method using *multiple quadratic Lyapunov functions* to approximate the CJSR of a given CSLS [PEDJ16].

1.1 Switching linear systems

Discrete-time *switching systems* are hybrid systems whose dynamics is given by the following equation:

$$x_{t+1} = f_{\sigma(t)}(x_t), \tag{1.1}$$

where $x_t \in \mathbb{R}^n$ is the *state* at time $t \in \mathbb{N}$, $\sigma(t) \in [m]$ is the *mode* at time $t \in \mathbb{N}$ and, for any mode σ, $f_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ is some function deciding the dynamics of the systems. We denote $(\sigma(0), \sigma(1), \dots)$ as the *switching sequence*.

In this thesis we are interested in *switching linear systems*, defined as follows.

Definition 1.1.1. *A* switching linear system *is a switching system as defined in* (1.1) *where, for all modes $\sigma \in [m]$, f_σ is a linear map i.e., $f_\sigma : \mathbb{R}^n \to \mathbb{R}^n : x \mapsto f_\sigma(x) = A_\sigma x$, with $A_\sigma \in \mathbb{R}^{n \times n}$.*

The dynamics of a switching linear system thus takes the following form:

$$x_{t+1} = A_{\sigma(t)} x_t. \tag{1.2}$$

We denote by $\mathbf{\Sigma} = \{A_i\}_{i\in[m]}$ the set of all matrices defining the system. Now suppose that the structure of the system is such that, if $\sigma(t) = 3$, then $\sigma(t+1)$ is necessarily equal to 3, and $\sigma(t+2) \in [m] \setminus \{3\}$. It means that, if for some time t the system has mode 3, then it has mode 3 exactly twice in a row. Such *logical rules* on the switching sequence defines $\mathcal{L} \subseteq [m]^{\mathbb{N}}$, the set of all admissible switching sequences. The set $\mathcal{L}$ is further defined in the following sections. In this work, a switching linear system is entirely defined by $\mathbf{\Sigma}$ and $\mathcal{L}$.

All along this thesis, we are interested in the *asymptotic stability* of switching linear systems. It is defined as follows:

Definition 1.1.2. *A switching linear system defined by $\mathbf{\Sigma}$ and $\mathcal{L}$ is* asymptotically stable *if, for all switching sequence $(\sigma(0), \sigma(1), \dots) \in \mathcal{L}$, and for all $x_0 \in \mathbb{R}^n$,*

$$\lim_{t\to\infty} A_{\sigma(t)} \dots A_{\sigma(0)} x_0 = 0. \tag{1.3}$$

For the sake of simplicity, from now, we refer to asymptotic stability as stability. Example 1.1.1 and Figure 1.1 highlight the fact that a switching linear system defined by $\mathbf{\Sigma} \subset \mathbb{R}^{n\times n}$ and some logical rules $\mathcal{L} \subseteq [m]^{\mathbb{N}}$ can be instable, even if each one of the underlying linear systems is stable. The latter observation makes us think that we need more advanced concepts in order to characterize the stability of switching linear systems.

Example 1.1.1 ([Jun09, Figure 1.1 and Figure 1.2])**.** *Let $\mathbf{\Sigma} = \{A_0, A_1\}$ with*

$$A_0 = \frac{2}{3}\begin{pmatrix} \cos(1.5) & \sin(1.5) \\ -2\sin(1.5) & 2\cos(1.5) \end{pmatrix} \text{ and } A_1 = \frac{2}{3}\begin{pmatrix} 2\cos(1.5) & 2\sin(1.5) \\ -\sin(1.5) & \cos(1.5) \end{pmatrix}. \tag{1.4}$$

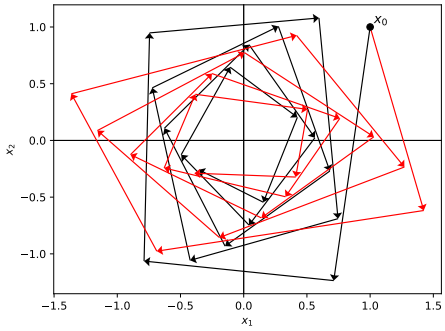
Consider the linear systems S_0 defined by $x_{t+1} = A_0 x_t$ and S_1 defined by $x_{t+1} = A_1 x_t$. Both are stable. Consider the switching linear system defined by $\mathbf{\Sigma}$ and $\mathcal{L}$, where

$$\mathcal{L} = \{(2), (2,1), (2,1,2), (2,1,2,1) \dots\}, \tag{1.5}$$

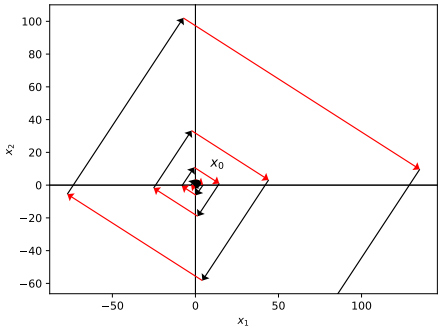
i.e. the set of switching sequences starting with $\sigma(0) = 2$ and where, for each $t > 0$, $\sigma(t) = 1$ implies $\sigma(t+1) = 2$ and $\sigma(t) = 2$ implies $\sigma(t+1) = 1$. This switching linear system seems to be instable as we can see on Figure 1.1(b) although S_0 and S_1 are both stable as we can see on Figure 1.1(a).

1.2 Arbitrary switching linear systems

Suppose that there is no logical rule on the switching sequence of a switching linear system, then we speak about *arbitrary switching linear systems.*



(a) S_0 (in black) and S_1 (in red). Both dynamical systems are stable.



(b) The switching linear system defined by $\mathbf{\Sigma} = \{A_0, A_1\}$ and $\mathcal{L} = \{(2), (2,1), (2,1,2), \dots\}$. This dynamical system is instable.

Figure 1.1: Illustration of the stability property for switching linear systems.

1.2.1 Definition and joint spectral radius

Definition 1.2.1. *An* arbitrary switching linear system *(*ASLS *for short) is a switching linear system as defined in Definition 1.1.2 whose switching sequences* $(\sigma(0), \sigma(1), \dots) \in \mathcal{L}$*, with* $\mathcal{L} = [m]^{\mathbb{N}}$*.*

We naturally denote by $S(\mathbf{\Sigma})$ the ASLS defined by the set of matrices $\mathbf{\Sigma}$.

The *joint spectral radius* of a set of matrices $\mathbf{\Sigma}$, introduced in [RG60] is defined as follows:

Definition 1.2.2. *Given a set of matrices* $\mathbf{\Sigma} = \{A_i\}_{i\in[m]}$*, the* joint spectral radius *(*JSR *for short)* $\rho(\mathbf{\Sigma})$ *is defined as*

$$\rho(\mathbf{\Sigma}) := \lim_{t\to\infty} \max\left\{ \|A_{\sigma(t-1)} \dots A_{\sigma(0)}\|^{1/t} : (\sigma(0), \dots, \sigma(t-1)) \in [m]^t \right\}. \tag{1.6}$$

As shown in Proposition 1.2.1, the JSR of $\mathbf{\Sigma}$ entirely characterizes the stability of the ASLS $S(\mathbf{\Sigma})$:

Proposition 1.2.1 ([Jun09, Corollary 1.1])**.** *Given a set of matrices* $\mathbf{\Sigma}$*, the ASLS* $S(\mathbf{\Sigma})$ *is stable if and only if* $\rho(\mathbf{\Sigma}) < 1$*.*

1.2.2 JSR Approximation with common Lyapunov functions

We can now focus on the approximation of the JSR with *common Lyapunov functions*. First we present an alternative definition of the JSR:

Proposition 1.2.2 ([Jun09, Proposition 2.6])**.** *For any bounded set* $\mathbf{\Sigma}$ *such that* $\rho(\mathbf{\Sigma}) \neq 0$*, the JSR can be defined as*

$$\rho(\mathbf{\Sigma}) = \inf_{\|\cdot\|} \max\{\|A\| : A \in \mathbf{\Sigma}\}. \tag{1.7}$$

Proposition 1.2.2 implies that, for any norm $\|\cdot\|$, $\rho(\mathbf{\Sigma}) \leq \max\{\|A\| : A \in \mathbf{\Sigma}\}$. Hence, in order to find an upper bound on the JSR, one can consider a large enough set of norms. Then, for each considered norm $\|\cdot\|$, compute $\max\{\|A\| : A \in \mathbf{\Sigma}\}$, and take the tightest.

We now explain why the latter assertion is linked to Lyapunov theory. We recall that a *Lyapunov function* is a function $V : \mathbb{R}^n \to \mathbb{R}$ satisfying the following conditions [Jun09]:

1. $V(x_t)$ tends to 0 as t tends to infinity,
2. and $V(x_t) \to 0$ implies $x_t \to 0$,

with x_t the state of some discrete-time dynamical system. According to Lyapunov theory, in our context, existence of a Lyapunov function is a necessary and sufficient condition for the stability of the system. If there is a norm $\|\cdot\|$ such that $\|A\| < 1$ for all $A \in \mathbf{\Sigma}$, then $\|\cdot\|$ is a Lyapunov function. Indeed, if $\|A\| < 1$ for all $A \in \mathbf{\Sigma}$, then $\max\{\|A\| : A \in \mathbf{\Sigma}\} < 1$, and, for any initial state $x_0 \in \mathbb{R}^n$,

$$\begin{aligned} \lim_{t\to\infty} \|x_t\| &= \lim_{t\to\infty} \|A_{\sigma(t-1)} \dots A_\sigma x_0\| \\ &\leq \lim_{t\to\infty} \|A_{\sigma(t-1)}\| \dots \|A_\sigma\| \|x_0\| \\ &\leq \lim_{t\to\infty} \left(\max\{\|A\| : A \in \mathbf{\Sigma}\}\right)^t \|x_0\| \\ &= 0, \end{aligned} \tag{1.8}$$

which is condition 1. Moreover, by continuity of linear maps, and since $\|x_t\| = 0$ implies $x_t = 0$ by definition of a norm, condition 2 holds.

Summarizing, if one wants to approximate the JSR, one can apply the methodology explained above. If a norm is found with $\max\{\|A\| : A \in \mathbf{\Sigma}\} < 1$, then $\rho(\mathbf{\Sigma}) < 1$ according to Proposition 1.2.2, and the system is stable according to both Proposition 1.2.1 and Lyapunov theory.

We now introduce *ellipsoidal norms*:

Definition 1.2.3 ([Jun09, Definition 2.8])**.** *Let $P \in \mathcal{S}^n$, the* vector ellipsoidal norm *is defined as*

$$\|\cdot\|_P : \mathbb{R}^n \to \mathbb{R} : x \mapsto \|x\|_P = \sqrt{x^T P x}, \tag{1.9}$$

and, for any $A \in \mathbb{R}^{n\times n}$, the induced matrix norm

$$\|A\|_P = \max_{\|x\|_P = 1} \|Ax\|_P \tag{1.10}$$

is the matrix ellipsoidal norm*.*

Note that, when it is clear from the context, we drop the words vector or matrix to respectively refer to vector and matrix ellipsoidal norms. In the search for a norm as explained above, we first propose to restrict ourselves to this particular familiy of norms. If there is $P \in \mathcal{S}^n$ such that $\max\{\|A\|_P : A \in \mathbf{\Sigma}\} < 1$, we say that $\|\cdot\|_P$ is a *common quadratic Lyapunov function* (or *CQLF* for short) for the ASLS $S(\mathbf{\Sigma})$. The following

proposition shows that searching for P can be done by solving *linear matrix inequalities* (*LMI*s):

Proposition 1.2.3 ([Jun09, Proposition 2.8])**.** *For any set of matrices $\mathbf{\Sigma}$, if there is $\gamma > 0$ and $P \in \mathcal{S}^n$ satisfying the following set of LMIs:*

$$\forall A \in \mathbf{\Sigma} : A^T P A \preceq \gamma^2 P, \tag{1.11}$$

then $\rho(\mathbf{\Sigma}) \leq \gamma$.

Indeed, finding $P \in \mathcal{S}^n$ such that LMIs (1.11) hold and finding $P \in \mathcal{S}^n$ such that $\|A\|_P \leq \gamma$ for all $A \in \mathbf{\Sigma}$ are equivalent [Jun09].

The tightest value of γ in LMIs (1.11) belongs to the interval $[\rho(\mathbf{\Sigma}), n^{1/2}\rho(\mathbf{\Sigma})]$ [AS98, BNT05]. In addition, [BNT05] provides another bound $\gamma \in [\rho(\mathbf{\Sigma}), m^{1/2}\rho(\mathbf{\Sigma})]$. Hence Theorem 1.2.4 holds.

Theorem 1.2.4 ([Jun09, Theorem 2.11])**.** *Consider a set of matrices $\mathbf{\Sigma}$. Let γ^* be the smallest value of γ such that there is $P \in \mathcal{S}^n$ such that, for all $A \in \mathbf{\Sigma}$, LMIs $A^T P A \preceq \gamma^2 P$ hold. Then*

$$\max\left\{m^{-1/2}, n^{-1/2}\right\}\gamma^* \leq \rho(\mathbf{\Sigma}) \leq \gamma^*. \tag{1.12}$$

In summary, in order to approximate the JSR of a given ASLS $S(\mathbf{\Sigma})$, we will try to find the minimal γ such that LMIs (1.11) hold for the set $\mathbf{\Sigma}$, and apply Theorem 1.2.4 and Proposition 1.2.1 to find a sufficient condition on stability or instability of the considered ASLS.

1.3 Constrained switching linear systems

A constrained switching linear system is a switching linear system whose switching sequences are constrained by logical rules, implying $\mathcal{L} \subset [m]^{\mathbb{N}}$. We model these rules by an *automaton*. Therefore, we need first to present some results from *automata theory*. Then, we will formally define a *constrained switching linear system*, and we will present a model-based method to approximate the *constrained joint spectral radius* of the latter, a generalization of the JSR.

1.3.1 Automata and languages

First, we define some concepts from *automata theory*. We give the definition of a *labelled graph*:

Definition 1.3.1. *A* labelled graph *consists of a finite set V of* vertices *together with a finite set $E \subseteq V \times V \times \mathcal{S}$ of* edges*, where $\mathcal{S}$ is a finite set of* labels*. It is noted $\mathbf{G}(V, E)$. Any edge $(u, v, \sigma) \in E$* starts *at vertex u,* terminates *at vertex v and* carries *the label σ.*

Without using the same notations, this definition is inspired from [LM95, Definition 2.2.1] and [LM95, Definition 3.1.1]. Without any loss of generality, from now, we assume that the set of labels $\mathcal{S} = [m]$, where m is the number of labels. Now we define the notion of *irreducibility*:

Definition 1.3.2 ([LM95, Definition 2.2.13])**.** *A labelled graph is* irreducible *if, for each ordered pair of vertices u and v, there is a* path *starting at u and terminating at v.*

In the latter, a *path* is defined as a sequence of edges. We now have defined every notion necessary to define an *automaton*:

Definition 1.3.3. *An* automaton *is an irreducible labelled graph. It is noted $\mathbf{G}(V, E)$.*

Example 1.3.1. *Consider the automaton $\mathbf{G}(V, E)$ depicted in Figure 1.2. It is defined by $V = \{a, b\}$, $\mathcal{A} = [m]$ with $m = 2$, and*

$$E = \{(a, b, 1), (b, a, 1), (a, a, 2), (b, b, 2)\}. \tag{1.13}$$

One can verify that a path exists from a to b e.g., $((a, b, 1))$ or $((a, a, 2), (a, b, 1))$, and that a path exists from b to a e.g., $((b, a, 1))$ or $((b, b, 2), (b, a, 1))$.

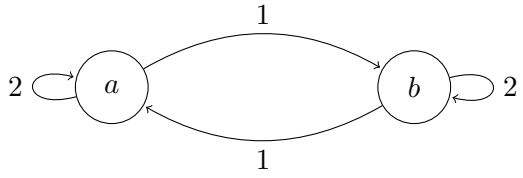


Figure 1.2: Example of an automaton.

A *word* of length $l \in \mathbb{N}$ *accepted* by an automaton $\mathbf{G}$ is a sequence of labels $(\sigma(0), \dots, \sigma(l-1))$ such that there exists a path $((u_i, v_i, \sigma(i)))_{i=0,\dots,l-1}$ in $\mathbf{G}$. The *language of* $\mathbf{G}$ *restricted to length* l, noted $\mathcal{L}_{\mathbf{G},l}$, is the set of all words of length l accepted by $\mathbf{G}$. Finally, the *language* of $\mathbf{G}$, noted $\mathcal{L}_{\mathbf{G}}$ is the set of all words accepted by $\mathbf{G}$, i.e. $\mathcal{L}_{\mathbf{G}} = \cup_{l \in \mathbb{N}} \mathcal{L}_{\mathbf{G},l}$.

All along this work, we will use the notion of *entropy* of an automaton. We give its definition:

Definition 1.3.4 ([LM95, Definition 4.1.1])**.** *Let $\mathbf{G}$ be an automaton. Its* entropy*, noted $h(\mathbf{G})$, is defined as*

$$h(\mathbf{G}) = \lim_{l \to \infty} \frac{\log_2 |\mathcal{L}_{\mathbf{G},l}|}{l}. \tag{1.14}$$

The entropy $h(\mathbf{G})$ is the growth rate of $|\mathcal{L}_{\mathbf{G},l}|$, the cardinality of its language restricted to length l.

1.3.2 Definition and constrained joint spectral radius

We are now able to define a *constrained switching linear system*:

Definition 1.3.5. *A* constrained switching linear system *(*CSLS *for short) is a switching linear system as defined in Definition 1.1.2 whose switching sequences* $(\sigma(0), \sigma(1), \dots) \in \mathcal{L}$ *with* $\mathcal{L} = \mathcal{L}_{\mathbf{G}}$ *for some automaton* $\mathbf{G}$*.*

We denote by $S(\mathbf{G}, \mathbf{\Sigma})$ the CSLS defined by the set of matrices $\mathbf{\Sigma}$ and whose switching sequences belong to the language accepted by the automaton $\mathbf{G}$.

This definition allows to model constraints on the switching sequence. We give an example of a CSLS, inspired from [PEDJ16, Section 4]:

Example 1.3.2. *Consider a plant that may experience control failures. Its dynamics is given by* $x_{t+1} = A_{\sigma(t)}x_t$ *where* $A_{\sigma(t)} = A + BK_{\sigma(t)}$*, with*

$$A = \begin{pmatrix} 0.47 & 0.28 \\ 0.07 & 0.23 \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{1.15}$$

$K_{\sigma(t)}$ *is defined as follows.* $K_1 = \begin{pmatrix} k_1 & k_2 \end{pmatrix}$*, with* $k_1 = -0.245$ *and* $k_2 = 0.135$*, corresponds to the mode where the controller works as expected.* $K_2 = \begin{pmatrix} 0 & k_2 \end{pmatrix}$ *and* $K_3 = \begin{pmatrix} k_1 & 0 \end{pmatrix}$ *respectively correspond to the modes when the first and the second part of controller fails. And* $K_4 = \begin{pmatrix} 0 & 0 \end{pmatrix}$ *corresponds to the mode when both parts fail. We consider as a contraint that the same part of the controller never fails twice in a row. This is modelled by the automaton* $\mathbf{G}$*, depicted in Figure 1.3.*

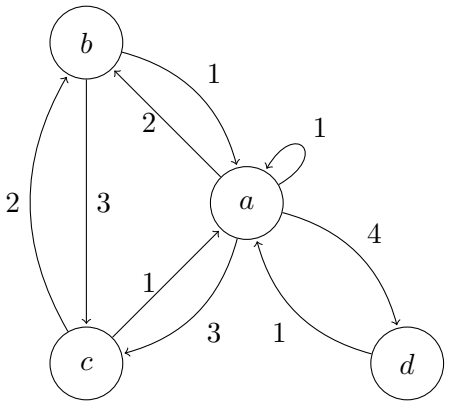


Figure 1.3: Automaton $\mathbf{G}$ for Example 1.3.2. No mode can fail twice in a row.

The *constrained joint spectral radius* of an automaton $\mathbf{G}$ and a set of matrices $\mathbf{\Sigma}$, introduced in [Dai11] is defined as follows:

Definition 1.3.6. *Given an automaton* $\mathbf{G}$ *and a set of matrices* $\mathbf{\Sigma} = \{A_i\}_{i \in [m]}$*, the*

constrained joint spectral radius *(*CJSR *for short)* $\rho(\mathbf{G}, \mathbf{\Sigma})$ *is defined as*

$$\rho(\mathbf{G}, \mathbf{\Sigma}) := \lim_{t\to\infty} \max \left\{ \|A_{\sigma(t-1)} \dots A_{\sigma(0)}\|^{1/t} : (\sigma(0), \dots, \sigma(t-1)) \in \mathcal{L}_{\mathbf{G},t} \right\}. \tag{1.16}$$

In the same way as for ASLSs, as showed in Proposition 1.3.1, the CJSR $\rho(\mathbf{G}, \mathbf{\Sigma})$ entirely characterizes the stability of the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$:

Proposition 1.3.1 ([Dai11, Corollary 2.8])**.** *Given an automaton* $\mathbf{G}$ *and a set of matrices* $\mathbf{\Sigma}$*, the CSLS* $S(\mathbf{G}, \mathbf{\Sigma})$ *is stable if and only if* $\rho(\mathbf{G}, \mathbf{\Sigma}) < 1$*.*

Remark 1.3.1. *An ASLS defined by* m *matrices is a CSLS where the automaton* $\mathbf{G}$ *only has one node and* m *loops, as illustrated in Figure 1.4. We call these automata are called* flowers *of order* m*.*

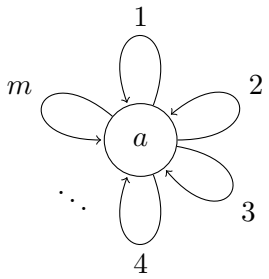


Figure 1.4: Flower of order m. If $\mathbf{G}$ is a flower of order m, the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ is equivalent to the ASLS $S(\mathbf{\Sigma})$, with $|\mathbf{\Sigma}| = m$.

1.3.3 CJSR approximation with multiple Lyapunov functions

As explained in Section 1.2.2, for any stable ASLS, there exists a common norm $\|\cdot\|$ such that $\|A\| < 1$ for all $A \in \mathbf{\Sigma}$. It is however not the case for CSLSs, as shown by the following example:

Example 1.3.3 ([PEDJ16, Example 1])**.** *Consider the one-dimensional CSLS* $S(\mathbf{G}, \mathbf{\Sigma})$ *with* $\mathbf{G}$ *as depicted in Figure 1.5, and* $\mathbf{\Sigma} = \{A_1, A_2\} = \{2, 1/8\}$*. Then there is no common norm such that* $2 = \|A_1\| < 1$*. However, given that*

$$\rho(\mathbf{G}, \mathbf{\Sigma}) = \lim_{t\to\infty} \|(A_1 A_2)^t\|^{1/(2t)} = \sqrt{A_1 A_2} = 1/2 < 1, \tag{1.17}$$

the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ *is stable.*

To circumvent this, the notion of *multinorms* in the context of CSLSs was introduced (see e.g. [PEDJ16]):

Definition 1.3.7 ([PEDJ16, Definition 1])**.** *A set of* multinorms *for a CSLS* $S(\mathbf{G}(V, E), \mathbf{\Sigma})$ *is a set of* $|V|$ *norms* $\{\|\cdot\|_u, u \in V\}$*.*

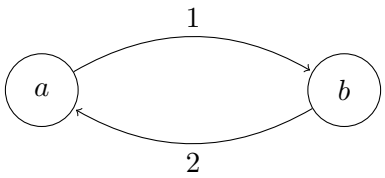


Figure 1.5: Automaton $\mathbf{G}$ for Example 1.3.3. The CSLS defined by $\mathbf{G}$ and $\mathbf{\Sigma} = \{2, 1/8\}$ is stable but there is no norm such that $\|A\| < 1$ for all $A \in \mathbf{\Sigma}$.

Similarly as for ASLSs, we then give a new definition for the CJSR:

Definition 1.3.8 ([PEDJ16, Proposition 2.2])**.** *Consider an automaton $\mathbf{G}(V, E)$ and a set of matrices $\mathbf{\Sigma}$. Then the CJSR can be defined as:*

$$\rho(\mathbf{G}, \mathbf{\Sigma}) = \inf_{\{\|\cdot\|_u,\, u \in V\}} \max \left\{ \frac{\|A_\sigma x\|_v}{\|x\|_u} : x \in \mathbb{R}^n,\, (u, v, \sigma) \in E \right\}. \tag{1.18}$$

We now do a similar analysis as for ASLSs. Proposition 1.3.8 implies that, for any set of multinorms $\{\|\cdot\|_u,\, u \in V\}$,

$$\rho(\mathbf{G}, \mathbf{\Sigma}) \leq \max \left\{ \frac{\|A_\sigma x\|_v}{\|x\|_u} : x \in \mathbb{R}^n,\, (u, v, \sigma) \in E \right\}. \tag{1.19}$$

If there is a set of multinorms $\{\|\cdot\|_u,\, u \in V\}$ such that $\|A_\sigma x\|_v / \|x\|_u < 1$ for all $x \in \mathbb{R}^n$ and for all $(u, v, \sigma) \in E$, then the set $\{\|\cdot\|_u,\, u \in V\}$ is a set of *multiple Lyapunov functions*, and the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ is stable. One can thus apply a very similar methodology as for ASLSs, but with family of multinorms. If a set of multinorms is found with $\max \{\|A_\sigma x\|_v / \|x\|_u : x \in \mathbb{R}^n,\, (u, v, \sigma) \in E\} < 1$, then $\rho(\mathbf{G}, \mathbf{\Sigma}) < 1$ according to Proposition 1.3.8, and the system is stable according to both Proposition 1.3.1 and Lyapunov theory.

Now, considering a CSLS $S(\mathbf{G}, \mathbf{\Sigma})$, we can restrict our search to multiple ellipsoidal norms (see Definition 1.2.3). It means, for all sets $\{P_u \in \mathcal{S}^n,\, u \in V\}$, take the set for which $\max \{\|A_\sigma x\|_{P_v} / \|x\|_{P_u} : x \in \mathbb{R}^n,\, (u, v, \sigma) \in E\}$ is the smallest. If this quantity is strictly smaller than 1, then $\{\|\cdot\|_{P_u},\, u \in V\}$ is a set of *multiple quadratic Lyapunov functions* (*MQLFs* for short). As for ASLSs, the following proposition shows that searching for $\{P_u \in \mathcal{S}^n,\, u \in V\}$ can be done by solving LMIs:

Proposition 1.3.2. *For any automaton $\mathbf{G}$ and set of matrices $\mathbf{\Sigma}$, if there is $\gamma > 0$ and $\{P_u \in \mathcal{S}^n,\, u \in V\}$ satisfying the following set of LMIs:*

$$\forall (u, v, \sigma) \in E : A_\sigma^T P_v A_\sigma \preceq \gamma^2 P_u, \tag{1.20}$$

then $\rho(\mathbf{G}, \mathbf{\Sigma}) \leq \gamma$.

Indeed one can see that the same equivalence as for the abritrary case holds (see Proposition 1.2.3).

We can generalize a part of Theorem 1.2.4 for the constrained case:

Theorem 1.3.3 ([PEDJ16, Theorem 3.1])**.** *Consider a set of matrices $\mathbf{\Sigma}$ and an automaton $\mathbf{G}(V, E)$. Let γ^* be the smallest value of γ such that there is a set $\{P_u \in \mathcal{S}^n,\, u \in V\}$ such that, for all $(u, v, \sigma) \in E$, LMIs $A_\sigma^T P_v A_\sigma \preceq \gamma^2 P_u$ hold. Then*

$$n^{-1/2}\gamma^* \leq \rho(\mathbf{G}, \mathbf{\Sigma}) \leq \gamma^*. \tag{1.21}$$

Finally, a more recent result presented in [LPJ19] states that the entropy (see Definiton 1.3.4) of the underlying automaton plays a role in the approximation of the CJSR:

Theorem 1.3.4 ([LPJ19, Theorem 2])**.** *Consider a set of matrices $\mathbf{\Sigma}$ and an automaton $\mathbf{G}(V, E)$. Let γ^* be the smallest value of γ such that there is a set $\{P_u \in \mathcal{S}^n,\, u \in V\}$ such that, for all $(u, v, \sigma) \in E$, LMIs $A_\sigma^T P_v A_\sigma \preceq \gamma^2 P_u$ hold. Then*

$$2^{-h(\mathbf{G})/2}\gamma^* \leq \rho(\mathbf{G}, \mathbf{\Sigma}) \tag{1.22}$$

Corollary 1.3.1 thus holds.

Corollary 1.3.1 ([LPJ19, Corollary 1])**.** *Consider a set of matrices $\mathbf{\Sigma}$ and an automaton $\mathbf{G}(V, E)$. Let γ^* be the smallest value of γ such that there is a set $\{P_u \in \mathcal{S}^n,\, u \in V\}$ such that, for all $(u, v, \sigma) \in E$, LMIs $A_\sigma^T P_v A_\sigma \preceq \gamma^2 P_u$ hold. Then*

$$\max\{2^{-h(\mathbf{G})/2}, n^{-1/2}\}\gamma^* \leq \rho(\mathbf{G}, \mathbf{\Sigma}) \leq \gamma^* \tag{1.23}$$

In summary, in order to approximate the CJSR of a given CSLS $S(\mathbf{G}(V, E), \mathbf{\Sigma})$, we will try to find the minimal γ such that LMIs (1.20) hold for the set of edges E, and apply Corollary 1.3.1 and Proposition 1.3.1 to find a sufficient condition on stability or instability of the considered CSLS.

Chapter 2

Existing approaches for data-driven ASLSs

In this chapter, we present existing approaches for approximating the JSR of an ASLS in a *data-driven fashion*. It means, for an ASLS $S(\mathbf{\Sigma})$, finding minimal γ such that LMIs (1.11) hold is not possible since $\mathbf{\Sigma}$ is not known. Some works have shown that it is possible to use different approaches to approximate the solution with a certain level of confidence. First, [BJW21] uses results from *scenario approach* (see [CG18] for an introduction) to approximate the tightest γ. Then, [RWJ21] tries a different approach by applying a *sensitivity analysis*. In Part II, we will draw on these two approaches to learn stability guarantees for black-box CSLSs by approximating their CJSR.

2.1 Problem statement

Consider an ASLS $S(\mathbf{\Sigma})$, where $\mathbf{\Sigma} = \{A_1, \dots, A_m\} \subseteq \mathbb{R}^{n\times n}$, where m is the number of matrices. For a fixed length $l \geq 1$, consider the set $\mathbf{\Sigma}^l := \{\mathbf{A} = A_{\sigma(l-1)} \dots A_{\sigma(0)} : (\sigma(0), \dots, \sigma(l-1)) \in [m]^l\}$. From the definition of the JSR (see Definition 1.2.2), it is straighforward that $\rho(\mathbf{\Sigma}^l) = \rho(\mathbf{\Sigma})^l$ [Jun09, Proposition 2.5]. Now, let $\Delta = \mathbf{\Sigma}^l \times \mathbb{S}$, and $\mathbb{P}$ be the *uniform distribution* on Δ. We seek to solve the following program:

$$\mathcal{P}(\Delta): \min_{\substack{P\in\mathbb{R}^{n\times n}\\ \gamma>0}} \gamma \tag{2.1a}$$

$$\text{s.t. } P \in \mathcal{S}^n, \tag{2.1b}$$

$$\forall(\mathbf{A}, x) \in \Delta, \quad (\mathbf{A}x)^T P(\mathbf{A}x) \leq \gamma^{2l} x^T P x. \tag{2.1c}$$

Remark 2.1.1. *Note that, following the definition of $\Delta = \mathbf{\Sigma}^l \times \mathbb{S}$, we restrict x to the unit sphere $\mathbb{S}$ in constraint* (2.1c)*. We can do this thanks to* homogeneity *of ASLS: for any $x \in \mathbb{R}^n$, $\mu > 0$, and $A \in \mathbf{\Sigma}$, it holds that $A(\mu x) = \mu A x$.*

We denote by $\gamma^*(\Delta)$ and $P^*(\Delta)$ the solutions of program $\mathcal{P}(\Delta)$. From Theorem 1.2.4, it yields $\rho(\mathbf{\Sigma}^l) \leq \gamma^*(\Delta)^l$, and thus $\rho(\mathbf{\Sigma}) \leq \gamma^*(\Delta)$. In a data-driven fashion, for any fixed length l, $\mathbf{\Sigma}$ is not known, hence Δ neither. We introduce the set of N *observations* as

$$\omega_N := \{(\mathbf{A}_i, x_i),\, i = 1, \dots, N\} \subset \Delta. \tag{2.2}$$

It means, we have access to N l-steps trajectories of the considered ASLS. Now, we can introduce $\mathcal{P}(\omega_N)$, the data-driven version of program $\mathcal{P}(\Delta)$:

$$\mathcal{P}(\omega_N) : \min_{\substack{P \in \mathbb{R}^{n\times n} \\ \gamma > 0}} (\gamma, \|P\|_F) \tag{2.3a}$$

$$\text{s.t. } P \in \mathcal{X} := \mathcal{S}^n \cap \{M \in \mathbb{R}^{n\times n} : I \preceq M \preceq CI\}, \tag{2.3b}$$

$$\forall (\mathbf{A}, x) \in \omega_N, \quad (\mathbf{A}x)^T P (\mathbf{A}x) \leq \gamma^{2l} x^T P x, \tag{2.3c}$$

for a large $C \geq n$. We denote by $\gamma^*(\omega_N)$ and $P^*(\omega_N)$ the solutions of program $\mathcal{P}(\omega_N)$.

There are multiple differences between programs $\mathcal{P}(\Delta)$ and $\mathcal{P}(\omega_N)$. The main difference is that there is an infinite number of LMIs 2.1c in $\mathcal{P}(\Delta)$, while there are N of them in $\mathcal{P}(\omega_N)$ (see constraint (2.3c)). Also, a *tie-breaking rule*[1] is defined in $\mathcal{P}(\omega_N)$. This tie-breaking rule allows for improving the probabilistic approximation that is presented in [BJW21] (see [KBJT19] for details). Finally, the feasible set for P is restricted from $\mathcal{S}^n$ in constraint (2.1b) to $\mathcal{S}^n \cap \{M \in \mathbb{R}^{n\times n} : I \preceq M \preceq CI\}$ in constraint (2.3b). This is done to ensure its compacity, guaranteeing the existence of a solution.

For a given set of observations ω_N, the problem tackled for ASLSs in [BJW21] and [RWJ21] is the **inference of an upper bound on** $\rho(\mathbf{G}, \mathbf{\Sigma})$ (denoted here by $\gamma^*(\Delta) \in \mathbb{R}$) **from** $\gamma^*(\omega_N) \in \mathbb{R}$ **and the matrix** $P^*(\omega_N) \in \mathbb{R}^{n\times n}$**, the solution of the sampled optimization problem** (2.3).

2.2 The scenario approach

In this subsection, we present the solution introduced in [KBJT19, BJW21]. This method is based on the scenario approach, a technique which allows to obtain solutions for optimization problems with only a sample of constraints [CG16, CG18, Cal10]. We first introduce some notions related with the problem tackled in this section.

Definition 2.2.1. *Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^d$, with non-empty interior and such that $0 \notin \mathcal{X}$. Let ω be a set, and $\{a_\delta\}_{\delta\in\omega}$, $\{b_\delta\}_{\delta\in\omega}$ be two collections of vectors in $\mathbb{R}^d$ such that $b_\delta^T x > 0$ for all $x \in \mathcal{X}$ and $\delta \in \omega$. A* quasi-linear *optimization problem has the form*

$$\min_{x\in\mathbb{R}^d,\, \lambda \geq 0} (\lambda, c(x)) \;\; s.t. \quad x \in \mathcal{X}, \tag{2.4a}$$

$$\forall \delta \in \omega,\, a_\delta^T x \leq \lambda b_\delta^T x, \tag{2.4b}$$

[1] We note $\min(f(x), g(x))$ the multiobjective optimization problem where $g(x)$ is used as a *tie-breaking rule*. That is, the objective is to minimize the function $f(x)$, and, in case there are several optimizers, the solution is the one which minimizes $g(x)$.

where $c : \mathcal{X} \to \mathbb{R}$ is a strongly convex function.

We now define the notion of *Barabanov matrix*:

Definition 2.2.2 ([BJW21, Definition 7])**.** *A matrix $A \in \mathbb{R}^{n\times n}$ is said to be* Barabanov *if there exists a symmetric matrix $P \succ 0$ and $\gamma \geq 0$ such that $A^T P A = \gamma^2 P$.*

We will also need two classical functions, which are known as *incomplete regularized beta function* and *inverse incomplete regularized beta function.*

Definition 2.2.3 ([Cro11, Equation (6.6.1)])**.** *The* incomplete beta function *is defined as:*

$$B : \mathbb{R}_{++} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \to \mathbb{R}_+ : (x; a, b) \mapsto B(x; a, b) = \int_0^x t^{a-1}(1-t)^{b-1} dt. \tag{2.5}$$

Definition 2.2.4 ([Cro11, Equation (6.6.2)])**.** *The* incomplete regularized beta function *is defined as:*

$$I : \mathbb{R}_{++} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \to \mathbb{R}_+ : (x; a, b) \mapsto I(x; a, b) = \frac{B(x; a, b)}{B(1; a, b)}. \tag{2.6}$$

Definition 2.2.5 ([MB73])**.** *The* inverse incomplete regularized beta function *is denoted by $I^{-1}(y; a, b)$, and is the function whose output is x such that $I(x; a, b) = y$.*

Finally, we define the notion of *spherical cap*:

Definition 2.2.6 ([Li11])**.** *The* spherical cap *on $\mathbb{S}$ of direction c and measure ε is defined as $\mathcal{C}(c, \varepsilon) := \{x \in \mathbb{S} : c^T x > \|c\|\delta(\varepsilon)\}$, where $\delta(\varepsilon)$ is defined as*

$$\delta(\varepsilon) = \sqrt{1 - I^{-1}(2\varepsilon; (n-1)/2, 1/2)}. \tag{2.7}$$

An illustration of Definition 2.2.6 is given in Figure 2.1 for $n = 2$. A plot of the function $\delta(\varepsilon)$ for different system dimensions is also given in Figure 2.2.

Authors of [BJW21] note that the data-driven program (2.3), $\mathcal{P}(\omega_N)$ is a quasi-linear program. They show that the scenario approach *chance-constrained* result [Cal10, Theorem 3.3] for convex optimization problems holds for quasi-linear problems (see [BJW21, Theorem 6]). It yields Theorem 2.2.1.

Theorem 2.2.1 ([BJW21, Corollary 12])**.** *Consider a set of m matrices $\mathbf{\Sigma} \subset \mathbb{R}^{n\times n}$, a set of N observations $\omega_N \in \Delta^N$, where, for a fixed length l, $\Delta = \mathbf{\Sigma}^l \times \mathbb{S}$ with $N \geq d := n(n+1)/2$. Suppose that $\mathbf{\Sigma}^l$ contains no Barabanov matrices. Consider program (2.3), $\mathcal{P}(\omega_N)$ with solutions $\gamma^*(\omega_N)$ and $P^*(\omega_N)$. Then, for a given level of confidence $\beta \in (0, 1)$, with probability at least β,*

$$\rho(\mathbf{\Sigma}) \leq \frac{\gamma^*(\omega_N)}{\delta\left(\frac{\varepsilon(\beta,N)\tilde{\kappa}(P^*(\omega_N))}{2(1/m^l)}\right)^{1/l}} \tag{2.8}$$

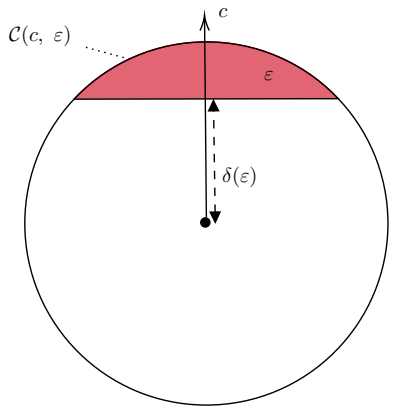


Figure 2.1: Illustration of the quantitiy $\delta(\varepsilon)$ for a given ε, and the spherical cap $\mathcal{C}(c, \varepsilon)$ for $n = 2$ and an arbitrary vector $c \in \mathbb{R}^2$.

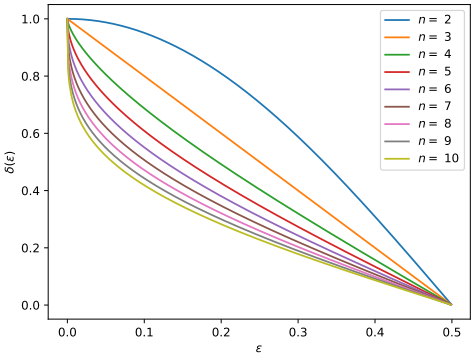


Figure 2.2: The function $\delta(\varepsilon)$ as defined in Equation 2.7 for all possible values of ε and for dimensions $n \in \{2, \dots, 10\}$.

where $\tilde{\kappa}(A) = \sqrt{\det(A)/\lambda_{\min}(A)^n}$ *for any* $A \in \mathbb{R}^{n\times n}$, *and*

$$\varepsilon(\beta, N) = I^{-1}(1 - \beta; d, N - d + 1). \tag{2.9}$$

Theorem 2.2.1 thus provides a first method to derive **probabilistic guarantee on the stability of a given ASLS**.

Remark 2.2.1. *In this thesis, we will not focus on the proof of Theorem 2.2.1, which is based on the notion of quasi-linear optimization problems and [KBJT19]. However, it is important to note that the sampling is uniform in the setting for ASLS (see Section 2.1). The factor* $1/m^l$ *in Equation* (2.8) *is the* ***minimal probability*** *to draw any matrix in* $\mathbf{\Sigma}^l$, *i.e.* $1/|\mathbf{\Sigma}^l|$. *Moreover, as it is explained in [KBJT19, Remark 16], it is equivalent to find a lower bound on the set of points that satisfy the constraints, and to take the complement of the latter. As a consequence, an equivalent bound as the one stated in Equation* (2.8) *is*

$$\rho(\mathbf{\Sigma}) \leq \frac{\gamma^*(\omega_N)}{\delta\left(\frac{1-(1-\varepsilon(\beta,N)m^l)\tilde{\kappa}'(P^*(\omega_N))}{2}\right)^{1/l}}, \tag{2.10}$$

with $\tilde{\kappa}'(A) = \sqrt{\det(A)/\lambda_{\max}(A)^n}$. *This alternative bound can be used when the bound* (2.8) *is infinite or weaker. This time, it is important to note that the factor* m^l *in Equation* (3.14) *is the inverse of the* ***maximal probability*** *to draw any matrix in* $\mathbf{\Sigma}^l$. *This is the same probability thanks to uniformity in the setting of [BJW21].*

2.3 The sensitivity analysis approach

We now present the solution to the problem stated in Section 2.1 proposed in [RWJ21], based on a sensitivity analysis. First we present a key result:

Lemma 2.3.1 ([RWJ21, Lemma 1])**.** *Consider the program* (2.1)*, $\mathcal{P}(\Delta)$ for the ASLS $S(\mathbf{\Sigma})$ with optimal cost $\gamma^*(\Delta)$. There exists a set $\omega \subset \Delta$ with $|\omega| = n(n+1)/2+1$ such that $\gamma^*(\omega) = \gamma^*(\Delta)$, where $\gamma^*(\omega)$ is the solution of the program* (2.3)*, $\mathcal{P}(\omega)$.*

Lemma 2.3.1 provides a result on the cardinality of the set of *support constraints* of the problem. Support constraints are constraints needed for the sampled problem $\mathcal{P}(\omega_N)$ to have the same solution as the full problem $\mathcal{P}(\Delta)$. As it will be explained in Remark 4.3.1, Lemma 2.3.1 can be improved.

Now, for a given number of observations N, Proposition 2.3.2 gives a bound on the conservativism of the sampled problem (2.3), $\mathcal{P}(\omega_N)$, with respect to the white-box problem (2.1), $\mathcal{P}(\Delta)$. It is a consequence of Lemma 2.3.1.

Proposition 2.3.2 ([RWJ21, Proposition 2])**.** *Consider the program* (2.1)*, $\mathcal{P}(\Delta)$ for the ASLS $S(\mathbf{\Sigma})$ with optimal cost $\gamma^*(\Delta)$. Let $\omega_N = \{(\mathbf{A}_i, x_i),\, i = 1, \dots, N\} \subset \Delta$ be a set of N samples. Suppose $N \geq n(n+1)/2+1$. Then, for all $\varepsilon \in (0,1]$, with probability at least*

$$\beta(\varepsilon, m, N) = 1 - \left(\frac{n(n+1)}{2} + 1\right)\left(1 - \frac{\varepsilon}{m^l}\right)^N, \tag{2.11}$$

there exists a set $\omega'_N := \{(\mathbf{A}_i, x'_i),\, i = 1, \dots, N\} \subset \Delta$ such that $\gamma^*(\omega'_N) = \gamma^*(\Delta)$ with $\|x_i - x'_i\| \leq \sqrt{2 - 2\delta(\varepsilon)}$, where $\gamma^*(\omega'_N)$ is the solution of program (2.3)*, $\mathcal{P}(\omega'_N)$.*

Finally, Proposition 2.3.2 allows to derive Theorem 2.3.3.

Theorem 2.3.3 ([RWJ21, Proposition 3])**.** *Consider a set of m matrices $\mathbf{\Sigma} \subset \mathbb{R}^{n\times n}$, a set of N observations $\omega_N \in \Delta^N$, where, for a fixed length l, $\Delta = \mathbf{\Sigma}^l \times \mathbb{S}$ with $N \geq d := n(n+1)/2+1$. Consider program* (2.3)*, $\mathcal{P}(\omega_N)$ with solutions $\gamma^*(\omega_N)$ and $P^*(\omega_N)$. Then, for a given level of confidence $\beta \in (0,1]$, let*

$$\varepsilon = m^l\left(1 - \sqrt[N]{\frac{2(1-\beta)}{n(n+1)+2}}\right). \tag{2.12}$$

Then, with probability at least β,

$$\rho(\mathbf{\Sigma}) \leq \left(\gamma^*(\omega_N)^l + \left[\gamma^*(\omega_N)^l + \mathcal{A}(\mathbf{\Sigma}^l)\right] d(\varepsilon)\kappa(P^*(\omega_N))\right)^{1/l}, \tag{2.13}$$

where $d(\varepsilon) = \sqrt{2 - 2\delta(\varepsilon)}$, and $\mathcal{A}(\mathbf{\Sigma}^l) = \max_{\mathbf{A}\in\mathbf{\Sigma}^l} \|\mathbf{A}\|$.

With the same setting as for the first method based on scenario approach, Theorem 2.3.3 based on sensitivity analysis provides an **alternative method to derive probabilistic guarantees on the stability of an ASLS**.

Remark 2.3.1. *The white-box quantity* $\mathcal{A}(\mathbf{\Sigma}^l) = \max_{\mathbf{A}\in\mathbf{\Sigma}^l} \|\mathbf{A}\|$ *is not known since* $\mathbf{\Sigma}$ *is not known. Based on chance-constrained results and scenario approach, [RWJ21] gives a closed form for it (see [RWJ21, Equation (15) and Theorem 1]), but we do not present it in this thesis. We will provide our own closed form for CSLS in the next chapters (see Section 4.4).*

Part II

Probabilistic stability guarantees for CSLSs

Chapter 3

Common Lyapunov functions

In this chapter, we present a first method to solve data-driven stability guarantees learning for CSLSs.

By generalizing results based on the scenario approach for ASLSs (see Section 2.2, [BJW21]), we will show that we are able to **construct an approximate common Lyapunov function** for the considered CSLS, from sampled data. We will also show that, under some assumptions, a smaller entropy of the underlying automaton allows for a better probabilistic guarantee.

This chapter is based on [BWJ22b] and is organized as follows. In Section 3.1, we formally present the problem. In Section 3.2 we propose a lifting result allowing us to reduce the computation of the CJSR to the JSR of a certain set of matrices. We are then able to extend data-driven results from [BJW21] to CSLS. Then, in Section 3.3, we investigate further the obtained generalization. Finally, in Section 3.4, we conclude about the derived method.

3.1 Problem statement

Consider a CSLS $S(\mathbf{G}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = \{A_1, \dots, A_m\} \subseteq \mathbb{R}^{n\times n}$, where m is the number of matrices, and where $\mathbf{G}(V, E)$ is some automaton constraining the switching sequence as explained in Section 1.3. For a fixed length $l \geq 1$, consider the set of possible products

$$\mathbf{\Pi}_l = \{A_{\sigma(l-1)} \dots A_{\sigma(0)} : (\sigma(0), \dots, \sigma(l-1)) \text{ is a word accepted by } \mathbf{G}\}. \tag{3.1}$$

We generalize the white-box problem (2.1) to l-steps CSLSs. Let $\Delta = \mathbf{\Pi}_l \times \mathbb{S}$. We seek to solve the following program:

$$\mathcal{P}(\Delta): \min_{\substack{P\in\mathbb{R}^{n\times n}\\ \gamma>0}} \gamma \tag{3.2a}$$

$$\text{s.t. } P \in \mathcal{S}^n, \tag{3.2b}$$

$$\forall(\mathbf{A}, x) \in \Delta, \quad (\mathbf{A}x)^T P(\mathbf{A}x) \leq \gamma^{2l} x^T P x. \tag{3.2c}$$

Note that program (3.2) is the same as program (2.1)[1], except that, following the definition of $\mathbf{\Pi}_l$ given in Equation (3.1), $\Delta = \mathbf{\Pi}_l \times \mathbb{S}$ is now constrained with $\mathbf{G}$ for $l > 1$. We denote by $\gamma^*(\Delta)$ and $P^*(\Delta)$ the solutions of program $\mathcal{P}(\Delta)$. We will show in Section 3.2 that, if $\gamma^*(\Delta) < 1$, $\|\cdot\|_{P^*(\Delta)}$ is a common Lyapunov function for the CSLS.

In a data-driven fashion, for any fixed length l, $\mathbf{\Sigma}$ and $\mathbf{G}(V, E)$ are not known, hence Δ neither. We introduce the set of N observations as

$$\omega_N := \{(\mathbf{A}_i, x_i),\, i = 1, \ldots, N\} \subset \Delta. \tag{3.3}$$

It means, we have access to N l-steps observations. We assume that observations are sampled in the following way. x_i is drawn randomly, uniformly and independently from $\mathbb{S}$. And, for each sample $i \in [N]$, the l matrices are generated from the automaton $\mathbf{G}(V, E)$ in the following way. An initial state u_0 is drawn randomly and uniformly from V. Then a random walk of length l is performed on $\mathbf{G}$, where, from $u_j \in V$, the next state u_{j+1} is drawn randomly, uniformly and independently from the set of its out-neighbours $\{u_{j+1} \in V : (u_j, u_{j+1}, \sigma_i(j) \in E\}$ where $\sigma_i(j)$ is the label corresponding to the edge linking u_j and u_{j+1}. The sequence of nodes $(u_0, \ldots, u_j, u_{j+1}, \ldots, u_l)$ form a switching sequence $\sigma_i(0), \ldots, \sigma_i(l-1)$, which maps to the matrices $A_{\sigma_i(0)}, \ldots, A_{\sigma_i(l-1)}$. It yields $\mathbf{A}_i = A_{\sigma_i(0)} \ldots A_{\sigma_i(l-1)}$. An illustration of this sampling strategy is provided in Figure 3.1.

Now, we can introduce $\mathcal{P}(\omega_N)$, the data-driven version of program $\mathcal{P}(\Delta)$:

$$\mathcal{P}(\omega_N): \min_{\substack{P\in\mathbb{R}^{n\times n}\\ \gamma>0}} (\gamma, \|P\|_F) \tag{3.4a}$$

$$\text{s.t. } P \in \mathcal{X} := \mathcal{S}^n \cap \{M \in \mathbb{R}^{n\times n} : I \preceq M \preceq CI\}, \tag{3.4b}$$

$$\forall(\mathbf{A}, x) \in \omega_N, \quad (\mathbf{A}x)^T P(\mathbf{A}x) \leq \gamma^{2l} x^T P x, \tag{3.4c}$$

for a large $C \geq n$. We denote by $\gamma^*(\omega_N)$ and $P^*(\omega_N)$ the solutions of program $\mathcal{P}(\omega_N)$. Again, program (3.4) is the same as program (2.3), except that ω_N is now constrained with $\mathbf{G}$.

For a given set of observations ω_N, the problem tackled for CSLSs is the inference of $\gamma^*(\Delta)$ from $\gamma^*(\omega_N)$ and $P^*(\omega_N)$. According to Proposition 1.3.3, we will show that $\gamma^*(\Delta)$ is an upper bound on $\rho(\mathbf{G}, \mathbf{\Sigma})$.

[1]For example, homogeneity, as defined in Remark 2.1.1, holds for CSLS.

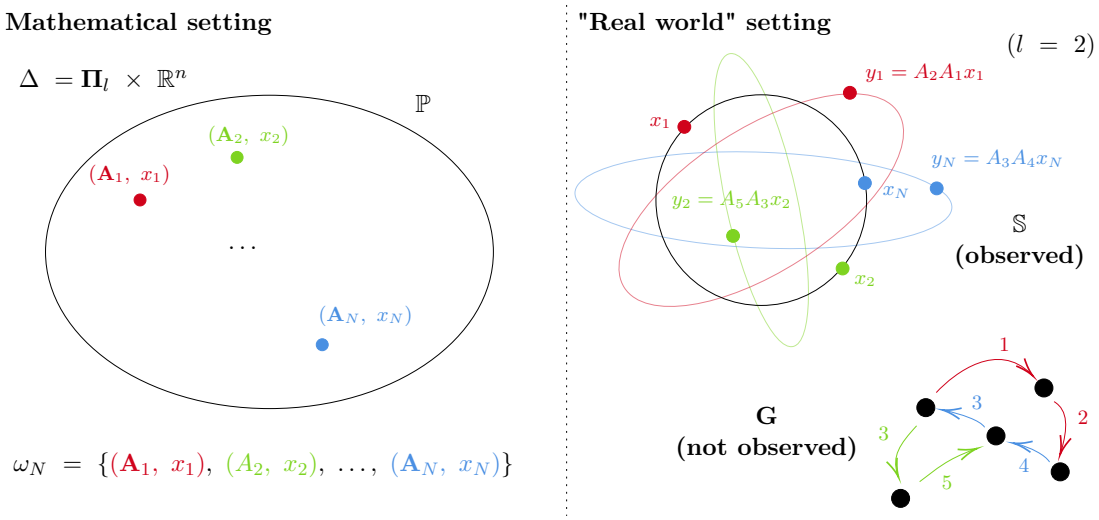


Figure 3.1: Illustration of the sampling strategy for the CQLF method. On the left, the sampling as it is mathematically defined in Equation (3.3): an observation is a point in Δ, sampled with some predefined measure $\mathbb{P}$. On the right, the origin of this mathematical definition. One sample N couples of points (x, y) in $\mathbb{S}^2$ where $y = \mathbf{A}x$, where $\mathbf{A} = A_{\sigma(l-1)} \ldots A_{\sigma(0)}$ where $(\sigma(l-1), \ldots, \sigma(0))$ is a word of length l is accepted by $\mathbf{G}$.

3.2 Lifting l-steps CSLSs to ASLSs

In this section, we present a lifting technique allowing us to reduce the computation of the CJSR to the computation of a simpler JSR. We show that the CJSR of a given CSLS can be bounded by the classical JSR of the set of all admissible product of a given length $l > 0$, i.e. $\mathbf{\Pi}_l$ defined in Equation (3.1).

Proposition 3.2.1. *For all $l > 0$, given an automaton $\mathbf{G}$ and a set of matrices $\mathbf{\Sigma}$, the CJSR $\rho(\mathbf{G}, \mathbf{\Sigma})$ and the JSR $\rho(\mathbf{\Pi}_l)$ satisfy*

$$\rho(\mathbf{G}, \mathbf{\Sigma})^l \leq \rho(\mathbf{\Pi}_l). \tag{3.5}$$

Moreover, the equality holds asymptotically, i.e.

$$\rho(\mathbf{G}, \mathbf{\Sigma}) = \lim_{l \to \infty} \rho(\mathbf{\Pi}_l)^{1/l}. \tag{3.6}$$

Proposition 3.2.1 provides a lifting result, stating that the simpler quantity $\rho(\mathbf{\Pi}_l)^{1/l}$ allows to approximate the true CJSR $\rho(\mathbf{G}, \mathbf{\Sigma})$. We call the ASLS $S(\mathbf{\Pi}_l)$ the *l-lifting* of the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$.

Proof. First we prove inequality (3.5). Since for all $t > 0$ and $l > 0$, $\mathbf{\Pi}_{lt} \subseteq \mathbf{\Pi}_l^t$, the definition of the JSR (see Definition 1.2.2) yields

$$\begin{aligned} \rho(\mathbf{\Pi}_l) &= \lim_{t\to\infty} \max\left\{\|\mathbf{A}\|^{1/t} : \mathbf{A} \in \mathbf{\Pi}_l^t\right\} \\ &\geq \lim_{t\to\infty} \max\left\{\|\mathbf{A}\|^{1/t} : \mathbf{A} \in \mathbf{\Pi}_{lt}\right\}. \end{aligned} \tag{3.7}$$

Letting $k := lt$ and using the definition of the CJSR (see Definition 1.3.6), the inequality becomes

$$\begin{aligned} \rho(\mathbf{\Pi}_l) &\geq \lim_{t\to\infty} \max\left\{\|\mathbf{A}\|^{l/k} : \mathbf{A} \in \mathbf{\Pi}_k\right\} \\ &= \left(\lim_{k\to\infty} \max\left\{\|\mathbf{A}\|^{1/k} : \mathbf{A} \in \mathbf{\Pi}_k\right\}\right)^l \\ &= \rho(\mathbf{G}, \mathbf{\Sigma})^l, \end{aligned} \tag{3.8}$$

which is the desired result. Now we prove asymptotic equality (3.6). First, given another definition of the JSR (see Definition 1.2.2) for any $l > 0$,

$$\begin{aligned} &\rho(\mathbf{\Pi}_l) \leq \max\{\|\mathbf{A}\| : \mathbf{A} \in \mathbf{\Pi}_l\} \\ \iff\ &\rho(\mathbf{\Pi}_l)^{1/l} \leq \max\left\{\|\mathbf{A}\|^{1/l} : \mathbf{A} \in \mathbf{\Pi}_l\right\}. \end{aligned} \tag{3.9}$$

Taking the limit in both sides, it gives $\lim_{l\to\infty} \rho(\mathbf{\Pi}_l)^{1/l} \leq \rho(\mathbf{G}, \mathbf{\Sigma})$. Since $\rho(\mathbf{\Pi}_l)^{1/l} \geq \rho(\mathbf{G}, \mathbf{\Sigma})$ for any $l > 0$, it gives the desired result. □

Even though other reductions of the CJSR computation problems to a simpler JSR have already been proposed in the litterature (see e.g. [Dai11] and [PEDJ16]), to the best of our knowledge, Proposition 3.2.1 is new, and will be useful for our purposes. Indeed Proposition 1.2.3 and Proposition 3.2.1 yield that $\rho(\mathbf{G}, \mathbf{\Sigma}) \leq \rho(\mathbf{\Pi}_l)^{1/l} \leq \gamma^*(\Delta)$. It means that if $\gamma^*(\Delta) < 1$, then $\rho(\mathbf{G}, \mathbf{\Sigma}) < 1$ and $\|\cdot\|_{P^*(\Delta)}$ is a common Lyapunov function for $S(\mathbf{G}, \mathbf{\Sigma})$.

Remark 3.2.1. *This method is limited to CSLSs that admit a common quadratic Lyapunov function. We say that a CSLS $S(\mathbf{G}(V, E), \mathbf{\Sigma})$ admits a CQLF if it admits a set of MQLFs $\{\|\cdot\|_{P_u}, u \in V\}$ where $P_u = P$ for all $u \in V$. Example 3.2.1 illustrates this. As it is explained in Section 1.3.3 and with counter-example 1.3.3, any CSLS does not possess a CQLF. Indeed a CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ admits a CQLF if the underlying ASLS $S(\mathbf{\Sigma})$ admits a CQLF.*

Remark 3.2.2. *LMIs* (1.11) *can be interpreted in terms of inclusions of convex sets (see [Phi17, Remark 2.17]). For any scalar function f, let $\Omega_f = \{x : f(x) \leq 1\}$ be the* one-level set *of f. For $P \in \mathcal{S}^n$, LMIs* (1.11) *are equivalent to, for all $A \in \mathbf{\Sigma}$,*

$$A\Omega_{\|\cdot\|_P} \subseteq \Omega_{\|\cdot\|_P}, \tag{3.10}$$

where $A\Omega := \{Ax : x \in \Omega\}$.

Example 3.2.1. *Consider the CSLS defined by* $\mathbf{G}$ *and* $\mathbf{\Sigma}$*, where* $\mathbf{\Sigma} = \{A_1, A_2\}$ *with*

$$A_1 = \begin{pmatrix} 0 & 2 \\ -1/8 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 3/2 \\ -1/4 & 0 \end{pmatrix}, \tag{3.11}$$

and $\mathbf{G}$ *is as depicted in Figure 3.2.*

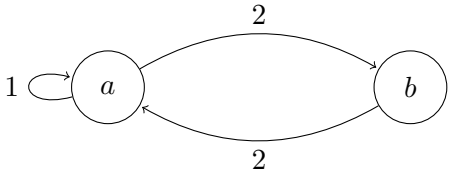


Figure 3.2: Automaton $\mathbf{G}$ for Example 3.2.1. If the systems switches to the second mode, it has to switch twice in a row.

Consider the matrix

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix}, \tag{3.12}$$

then there exists $\gamma \in (0, 1)$ *such that* $A_1^T P A_1 \preceq \gamma^2 P$ *and* $A_2^T P A_2 \preceq \gamma^2 P$*. Hence* $\|\cdot\|_P$ *is a CQLF for* $S(\mathbf{G}, \mathbf{\Sigma})$*. Following Remark 3.2.2, Figure 3.3 illustrates this.*

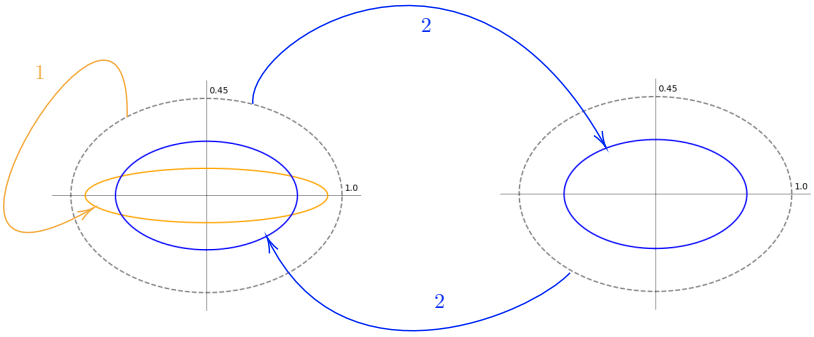


Figure 3.3: Illustration of a CQLF for CSLSs. In grey, $\Omega_{\|\cdot\|_P}$ the one-level set of the CQLF (the same for every node). In orange, $A_1\Omega_{\|\cdot\|_P}$. In blue, $A_2\Omega_{\|\cdot\|_P}$. One can indeed observe that $A_1\Omega_{\|\cdot\|_P} \subseteq \Omega_{\|\cdot\|_P}$ and $A_2\Omega_{\|\cdot\|_P} \subseteq \Omega_{\|\cdot\|_P}$, thus $\|\cdot\|_P$ is a CQLF and the considered CSLS is stable.

For a given length $l > 0$, we are now able to find an upper bound on the CJSR of a given CSLS by approximating a simpler JSR $\rho(\mathbf{\Pi}_l)$. As explained in Chapter 2, previous

results [KBJT19, BJW21] hold for data-driven approximation of the JSR. We will rely on these previous works. This raises the next question.

Remark 3.2.3. *If the existence of a CQLF for some CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ is equivalent to the existence of a CQLF for its underlying ASLS $S(\mathbf{\Sigma})$,* ***how is the problem of approximating the CJSR different from the problem of approximating the JSR?*** *Following Remark 2.2.1, there are two main differences:*

- *First, the measure on the matrices in $\mathbf{\Pi}_l$ is not necessarily uniform for $l > 1$. Example 3.2.2 illustrates this.*
- *Second, suppose the measure is uniform, the probability of sampling any matrix in $\mathbf{\Pi}_l$ is $1/|\mathbf{\Pi}_l|$. However, $|\mathbf{\Pi}_l|$ is not known, and can be smaller than m^l, as Example 3.2.3 highlights.*

Example 3.2.2. *Consider the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$, where $\mathbf{G}$ is as depicted in Figure 3.2, and $\mathbf{\Sigma}$ contains two matrices A_1 and A_2. Suppose that the considered length is $l = 2$. It yields $\mathbf{\Pi}_2 = \{A_1A_1, A_1A_2, A_2A_1, A_2A_2\}$. However, following the sampling strategy described in Section 3.1, there are five sampling scenarios. The latter are presented in Table 3.1.*

Random walk	With probability	Resulting matrix
(a, a, a)	$1/8$	$\mathbf{A}_1 = A_1A_1$
(a, a, b)	$1/8$	$\mathbf{A}_2 = A_1A_2$
(a, b, b)	$1/4$	$\mathbf{A}_3 = A_2A_2$
(b, a, a)	$1/4$	$\mathbf{A}_4 = A_2A_1$
(b, a, b)	$1/4$	$\mathbf{A}_5 = A_2A_2 = \mathbf{A}_3$

Table 3.1: Different random walks of length 2 in automaton $\mathbf{G}$ as depicted in Figure 3.2, with the probability of happening and the corresponding modes.

It yields that the probability of sampling $\mathbf{A}_1$ is 1/8, the probability of sampling $\mathbf{A}_2$ is 1/8, the probability of sampling $\mathbf{A}_3$ is 1/2, and the probability of sampling $\mathbf{A}_4$ is 1/4.

Example 3.2.3. *Consider a CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ with $\mathbf{G}$ as depicted in Figure 3.4, and $\mathbf{\Sigma} = \{A_1, A_2, A_3\}$. Suppose $l = 2$. The matrix $A_1A_3 \notin \mathbf{\Pi}_2$. It yields $|\mathbf{\Pi}_2| < m^l = 9$.*

Based on the reduction to ASLSs presented in Proposition 3.2.1, we present a generalization of Theorem 2.2.1 ([BJW21, Corollary 12]) for CSLSs taking into account non-uniformity.

Theorem 3.2.2. *Consider an automaton $\mathbf{G}$, a set of matrices $\mathbf{\Sigma} \subset \mathbb{R}^{n\times n}$, a fixed length $l > 0$ and a set of samples $\omega_N \subset \Delta$ obtained as explained in Section 3.1. Suppose that $N \geq d := n(n+1)/2$. Suppose $\mathbf{\Pi}_l$ contains no Barabanov matrices. Consider the problem 3.4, $\mathcal{P}(\omega_N)$ with solutions $\gamma^*(\omega_N)$ and $P^*(\omega_N)$. Then, for a given level of*

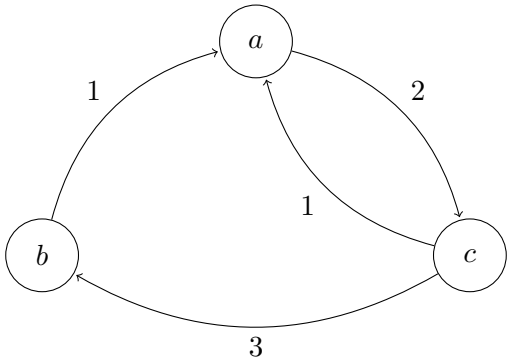


Figure 3.4: Automaton $\mathbf{G}$ for Example 3.2.3. The word $(1, 3)$ is not accepted by $\mathbf{G}$.

confidence $\beta \in (0, 1)$*, with probability at least* β*,*

$$\rho(\mathbf{G}, \mathbf{\Sigma}) \leq \frac{\gamma^*(\omega_N)}{\delta\left(\frac{\varepsilon(\beta,N)\tilde{\kappa}(P^*(\omega_N))}{2p_{l,\min}}\right)^{1/l}}, \tag{3.13}$$

where $p_{l,\min}$ *is the minimal probability among probabilities to draw each matrices in* $\mathbf{\Pi}_l$*, and other quantities are given in Theorem 2.2.1.*

Based on the lifting result exposed in Proposition 3.2.1, to the best of our knowledge, Theorem 3.2.2 provides the first result allowing to **obtain, from a set of observations, data-driven probabilistic guarantee on the stability of a given CSLS**.

Proof. The proof follows exactly the same lines as the one of Theorem 2.2.1, except that $1/m^l$ (the probability to draw any matrix in $\mathbf{\Sigma}^l$) is replaced with $p_{l,\min}$ in proof of [KBJT19, Corollary 11], following Remark 2.2.1. □

Remark 3.2.4. *In parallel to Remark 2.2.1, we are able to provide the following equivalent alternative bound, replacing* (3.13)*. With the same notations as in Equation 3.14,*

$$\rho(\mathbf{\Sigma}) \leq \frac{\gamma^*(\omega_N)}{\delta\left(\frac{1-(1-\varepsilon(\beta,N)/p_{l,\max})\tilde{\kappa}'(P^*(\omega_N))}{2}\right)^{1/l}}, \tag{3.14}$$

where $p_{l,\max}$ *is the maximal probability among probabilities to draw each matrices in* $\mathbf{\Pi}_l$*.*

Theorem 3.2.2 provides a general result. We will now build corollaries on top of this theorem to show how it can used in practice.

3.3 Further results in case of uniformity

Suppose that the probability measure on $\mathbf{\Pi}_l$ is uniform. Then, $p_{l,\min}$ in Equation (3.13) can be replaced by $1/|\mathbf{\Pi}_l|$. Let $\mathcal{L}_{\mathbf{G},l}$ be the language accepted by $\mathbf{G}$ restricted to length l. Since two different products of some length l (two products generated by different words of length l) can yield the same matrix, $|\mathbf{\Pi}_l| \leq |\mathcal{L}_{\mathbf{G},l}|$. Thus, one can bound the cardinality of $\mathcal{L}_{\mathbf{G},l}$ in order to bound the cardinality of $\mathbf{\Pi}_l$.

First, let us highlight the role of the entropy $h(\mathbf{G})$ (see Definition 1.3.4). Its definition allows to derive the following asymptotic result:

Corollary 3.3.1. *For $l \to \infty$, consider that the probability measure on $\mathbf{\Pi}_l$ is uniform. Then, with the same setting as in Theorem 3.2.2, for a given level of confidence $\beta \in (0,1)$, with probability at least β,*

$$\rho(\mathbf{G}, \mathbf{\Sigma}) \leq \lim_{l\to\infty} \frac{\gamma^*(\omega_N)}{\delta\left(\varepsilon(\beta, N)\tilde{\kappa}(P^*(\omega_N))2^{lh(\mathbf{G})}/2\right)^{1/l}}. \tag{3.15}$$

Proof. The definition of the entropy (see Definition 1.3.4) yields

$$\lim_{l\to\infty} |\mathcal{L}_{\mathbf{G},l}| = \lim_{l\to\infty} 2^{lh(\mathbf{G})}, \tag{3.16}$$

and for any $l > 0$, $p_{l,\min} = 1/|\mathbf{\Pi}_l| \geq 1/|\mathcal{L}_{\mathbf{G},l}|$ in Equation 3.13. □

Corollary 3.3.1 provides an asymptotic estimate of the probabilistic upper bound in Theorem 3.2.2, as a function of the entropy of the automaton $\mathbf{G}$. One can see that an automaton with small entropy allows for a better estimate of the CJSR, for a fixed number of samples N. This is illustrated in Figure 3.5.

In order to get some intuition about this result and the role of the entropy, let us consider the case where the considered CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ is such that the entropy of its automaton $h(\mathbf{G}) = 0$. This happens when all the nodes of $\mathbf{G}$ have an indegree and outdegree of one. Then, it holds that, for any value of $\beta \in (0,1)$,

$$\lim_{l\to\infty} \delta\left(\varepsilon(\beta, N)\tilde{\kappa}(P^*(\omega_N))2^{lh(\mathbf{G})}/2\right)^{-1/l} = 1, \tag{3.17}$$

as $\delta(\epsilon) \in (0,1)$ for any constant ϵ (see Definition 2.2.6). This yields that, with an arbitrarily large confidence level, and for any N, there is an arbitrarily small parameter $\nu(l) > 0$ depending on l such that

$$\rho(\mathbf{G}, \mathbf{\Sigma}) < (1 + \nu(l))\gamma^*(\omega_N). \tag{3.18}$$

Intuitively, it means that if $\mathbf{G}$ has a zero entropy, with a large enough length l, it is sufficient to have one observation to "capture" all the information necessary to derive a quasi-deterministic upper bound on the CJSR.

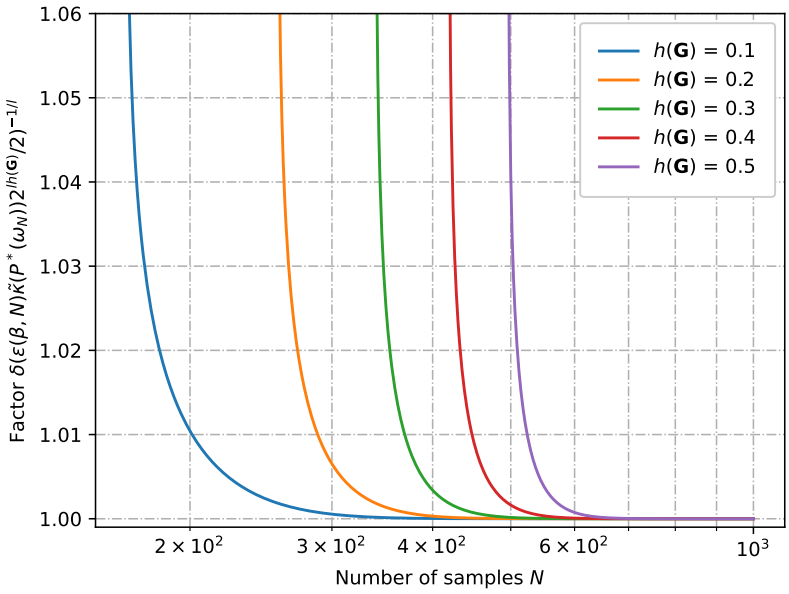


Figure 3.5: Shape of the factor $\delta\left(\varepsilon(\beta,N)\tilde{\kappa}(P^*(\omega_N))2^{lh(\mathbf{G})}/2\right)^{-1/l}$ in Corollary 3.3.1 with respect to the entropy, for a confidence level $\beta = 95\%$, a large l (here $l = 50$), and $n = 2$. One can see that this factor converges to 1 as N increases, and that a smaller entropy allows a faster convergence.

We can also verify that we can recover the arbitrary case. Suppose $\mathbf{G}$ is equivalent to the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ is the ASLS $S(\mathbf{\Sigma})$, where $|\mathbf{\Sigma}| = m$. It means, $\mathbf{G}$ is a flower of order m, as depicted in Figure 1.4. Then $2^{h(\mathbf{G})} = m$. As we assume uniformity, in this case, Corollary 3.3.1 is exactly Theorem 2.2.1.

We now show that, for any finite $l > 0$, we can derive a practical bound using classical results from graph theory. First, the following proposition holds:

Proposition 3.3.1. *Let A be the* adjacency matrix *of some automaton $\mathbf{G}(V, E)$. Let $\lambda_1 \leq \cdots \leq \lambda_{|V|}$ be the* eigenvalues *of A. Assume A is diagonalizable. Then, for any $l > 0$, $|\mathbf{\Pi}_l| \leq |V|\lambda_{|V|}^l$.*

Proof. Let G be the underlying graph of $\mathbf{G}$ without any labelling. Let $w_l(G)$ be the number of walks in the G. We know that $|\mathbf{\Pi}_l| \leq |\mathcal{L}_{\mathbf{G},l}| \leq w_l(G)$.

It remains to show that $w_l(G) \leq n\lambda_n^l$, with $n = |V|$. Let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be the corresponding eigenvectors. For $l \geq 0$, we can write the spectral form of A^l as $A^l = \sum_{i=1}^n \lambda_i^l \mathbf{v}_i \mathbf{v}_i^T$. Also,

$w_l(G) = \sum_{i,j=1}^{n} (A^l)_{ij} = \mathbf{1}^T A^l \mathbf{1}$ for $\mathbf{1}$ a vector of all ones. It follows that

$$w_l(G) = \mathbf{1}^T \left(\sum_{i=1}^{n} \lambda_i^l \mathbf{v}_i \mathbf{v}_i^T \right) \mathbf{1} = \sum_{i=1}^{n} \lambda_i^l \left(\mathbf{v}_i^T \mathbf{1} \right)^2. \tag{3.19}$$

Let $\alpha_i := \left(\mathbf{v}_i^T \mathbf{1} \right)^2$, we can write $w_l(G) = \sum_{i=1}^{n} \alpha_i \lambda_i^l$. We also know that $w_0(G) = \sum_{i=1}^{n} \alpha_i = n$. Since $\alpha_i \geq 0$ for any i, it follows that

$$w_l(G) = \sum_{i=1}^{n} \alpha_i \lambda_i^l = \lambda_n^l \sum_{i=1}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_n} \right)^l \leq \lambda_n^l \sum_{i=1}^{n} \alpha_i = n \lambda_n^l, \tag{3.20}$$

which gives the result. □

Proposition 3.3.1 directly gives the following corollary:

Corollary 3.3.2. *For $l > 0$, consider that the probability measure on $\mathbf{\Pi}_l$ is uniform. With the same setting as in Theorem 3.2.2, let A be the adjacency matrix of the automaton $\mathbf{G}(V, E)$. Let $\lambda_1 \leq \cdots \leq \lambda_{|V|}$ be the eigenvalues of A. Assume A is diagonalizable. Then, for a given level of confidence $\beta \in (0, 1)$, with probability at least β,*

$$\rho(\mathbf{G}, \mathbf{\Sigma}) \leq \frac{\gamma^*(\omega_N)}{\delta \left(\varepsilon(\beta, N) \tilde{\kappa}(P^*(\omega_N)) |V| \lambda_{|V|}^l / 2 \right)^{1/l}}. \tag{3.21}$$

Corollary 3.3.2 provides a probabilistic upper bound in Theorem 3.2.2, as a function of the largest eigenvalue of the adjacency matrix of $\mathbf{G}$. One can see that an automaton with a small largest eigenvalue allows for a better estimate of the CJSR, for a fixed number of samples N and length l.

3.4 Conclusions

In this chapter, we generalized data-driven results for ASLSs to CSLSs.

We first proposed a lifting result allowing us to reduce the computation of the CJSR of a given CSLS to the computation of a simpler JSR. We then raised two differences between the sampling strategies of ASLSs and CSLSs (see Remark 3.2.3). We then generalized the main theorem of [BJW21] to CSLSs in Theorem 2.2.1. In case of uniformity, we investigated further the obtained generalization. Finally, we showed that the entropy of the automaton plays a role, as a smaller entropy allows for a better guarantee for the stability.

Chapter 4

Multiple Lyapunov functions

In the precedent chapter, we found a method to solve data-driven stability guarantees for CSLSs. However, this method requires strong assumptions. Following Remark 3.2.1, it can only be applied to CSLSs that admit a common quadratic Lyapunov function. Moreover, further results derived in Section 3.3 come with a strong assumption of uniformity, which we know is not always respected as illustrated in Example 3.2.2. In this chapter, inspired by previous results for ASLSs using a sensitivity analysis approach (see [RWJ21] and Section 2.3), we present a method using **multiple Lyapunov functions**. It is thus not restricted to CSLSs that possess a CQLF anymore. However, this method comes with a strong assumption: we assume that **we observe the nodes of the automaton in addition to the states**, which is a big change of setting compared to the method presented in Chapter 3.

This chapter is based on [BWJ22a] and is organized as follows. In section 4.1, we formalize the new problem. In Section 4.2, we derive a deterministic lower bound for the CJSR. In Section 4.3, we show that we can generalize and even improve results of [RWJ21] for CSLSs. In Section 4.4, we solve a sub-problem needed to derive a practical bound, namely the estimation of the maximal norm of matrices in $\mathbf{\Sigma}$. Finally, in Section 4.5, we conclude about the derived method.

4.1 From CQLFs for ASLSs to MQLFs for CSLSs

Consider a CSLS $S(\mathbf{G}(V,E),\mathbf{\Sigma})$, where $\mathbf{\Sigma} = \{A_1,\dots,A_m\} \subseteq \mathbb{R}^{n\times n}$, where m is the number of matrices, and where $\mathbf{G}(V,E)$ is some automaton constraining the switching sequence as explained in Section 1.3. We generalize the white-box problem (2.1) to CSLSs, using MQLFs. Note that, for the sake of simplicity, we focus on 1-step Lyapunov functions, thus $l = 1$, where l is as defined in Section 2.1. Generalizing this method to longer traces is further work.

In this chapter, we assume that we observe both the nodes of the automaton and the

states. We redefine our set Δ. Let $\Delta = E \times \mathbb{S}$. We seek to solve the following program:

$$\mathcal{P}(\Delta): \min_{\substack{\{P_u \in \mathbb{R}^{n\times n},\, u\in V\}\\ \gamma>0}} \gamma \tag{4.1a}$$

$$\text{s.t. } \forall u \in V, \quad P_u \in \mathcal{S}^n, \tag{4.1b}$$

$$\forall ((u,v,\sigma),x) \in \Delta, \quad (A_\sigma x)^T P_v (A_\sigma x) \leq \gamma^2 x^T P_u x. \tag{4.1c}$$

We denote by $\gamma^*(\Delta)$ and $\{P_u^*(\Delta),\, u \in V\}$ the solutions of program 4.1, $\mathcal{P}(\Delta)$. For $u \in V$, $\gamma^*(\Delta)$ and matrices $P^*(\Delta)_u$ are equal to γ^* and P_u in Theorem 1.3.3. Thus, if $\gamma^*(\Delta) < 1$, the set of ellipsoidal norms $\{\|\cdot\|_{P^*(\Delta)_u},\, u \in V\}$ is a set of MQLFs.

Remark 4.1.1. *In parallel to Remark 3.2.1, this method is not limited to CSLSs that only admit a CQLF, it extends to all CSLSs that admit a set of MQLFs. We give an example as well as a visualization of a CSLS that admit a set of MQLFs in Example 4.1.1.*

Remark 4.1.2. *LMIs* (1.20) *can also be interpreted in terms of inclusions of convex sets (see Remark 3.2.2 and [Phi17, Remark 2.17]). Consider the CSLS* $S(\mathbf{G}(V,E),\mathbf{\Sigma})$ *with* $\mathbf{\Sigma} = \{A_1,\dots,A_m\}$. *For the set* $\{P_u \in \mathcal{S}^n,\, u \in V\}$, *LMIs* (1.20) *are equivalent to the following condition: for all edges* $(u,v,\sigma) \in E$,

$$A_\sigma \Omega_{\|\cdot\|_{P_u}} \subseteq \Omega_{\|\cdot\|_{P_v}}. \tag{4.2}$$

Example 4.1.1 ([Phi17, Example 2.16])**.** *Consider the CSLS defined by* $\mathbf{G}$ *and* $\mathbf{\Sigma}$, *where* $\mathbf{\Sigma} = \{A_1, A_2\}$ *with*

$$A_1 = \begin{pmatrix} 0 & 2 \\ -1/8 & 0 \end{pmatrix},\ A_2 = \begin{pmatrix} 0 & 1/8 \\ -2 & 0 \end{pmatrix}, \tag{4.3}$$

and $\mathbf{G}$ *is as depicted in Figure 3.2.*

Consider the matrices

$$P_a = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix}, \quad P_b = \begin{pmatrix} 40 & 0 \\ 0 & 1/8 \end{pmatrix}. \tag{4.4}$$

Then, there exists $\gamma \in (0,1)$ *such that* $A_1^T P_a A_1 \preceq \gamma^2 P_a$, $A_2^T P_a A_2 \preceq \gamma^2 P_b$, *and* $A_2^T P_b A_2 \preceq \gamma^2 P_a$. *Hence* $\{\|\cdot\|_{P_u},\, u \in V\}$ *is a set of MQLFs for* $S(\mathbf{G},\mathbf{\Sigma})$. *Following Remark 4.1.2, Figure 4.1 illustrates this.*

In a data-driven fashion, $\mathbf{\Sigma}$ is not known, and neither is the automaton $\mathbf{G}(V,E)$. As for the method based on CQLF, Δ is not known. However, in order to derive this method, **we assume that nodes are observed**. One observation consists in an ordered pair of points in an extended state space $\mathbb{R}^n \times V$. The i-th observation is a couple of initial and final states and nodes. It is noted $((x_i,u_i),(y_i,v_i)) \in (\mathbb{R}^n \times V)^2$, where $(u_i,v_i,\sigma_i) \in E$ for some label $\sigma_i \in \{1,\dots,m\}$, with $y_i = A_{\sigma_i} x_i$. For any $i \in [N]$, x_i and (u_i,v_i,σ_i) are drawn randomly, uniformly and independently from respectively $\mathbb{S}$ and E. We attract the attention of the reader on the fact that the sampled *mode* is not known.

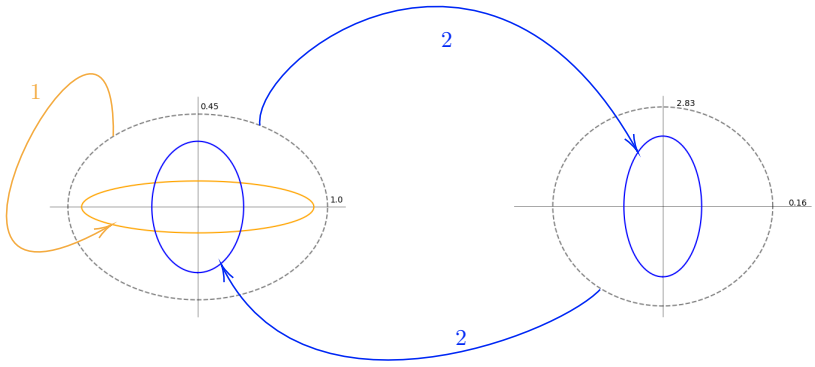


Figure 4.1: Illustration of a set of MQLFs for CSLSs. In grey, for each node $u \in \{a, b\}$, $\Omega_{\|\cdot\|_{P_u}}$. For each node $u = a, b$, in orange, $A_1\Omega_{\|\cdot\|_{P_u}}$ and, in blue, $A_2\Omega_{\|\cdot\|_{P_u}}$. One can indeed observe that $A_1\Omega_{\|\cdot\|_{P_a}} \subseteq \Omega_{\|\cdot\|_{P_a}}$, $A_2\Omega_{\|\cdot\|_{P_a}} \subseteq \Omega_{\|\cdot\|_{P_b}}$, and $A_2\Omega_{\|\cdot\|_{P_b}} \subseteq \Omega_{\|\cdot\|_{P_a}}$. It yields that $\{\|\cdot\|_{P_u}, u \in V\}$ is a set of MQLFs and the considered CSLS is stable. We draw the attention of the reader to the fact that, unlike in Figure 3.3, the two ellipsoids $\Omega_{\|\cdot\|_{P_u}}$ are not identical for $u \in V$.

We thus introduce the set of N observations as

$$\omega_N := \{((u_i, v_i, \sigma_i), x_i),\ i = 1, \dots, N\} \subset \Delta, \tag{4.5}$$

where x_i, u_i, v_i and σ_i are as described above. An illustration of this sampling strategy is provided in Figure 4.2.

Now, for a given set of observations ω_N, we introduce $\mathcal{P}(\omega_N)$, the data-driven version of program $\mathcal{P}(\Delta)$:

$$\mathcal{P}(\omega_N): \min_{\substack{\{P_u \in \mathbb{R}^{n\times n},\, u\in V\}\\ \gamma>0}} \gamma \tag{4.6a}$$

$$\text{s.t. } \forall u \in V, \quad P_u \in \mathcal{X} := \mathcal{S}^n \cap \{M \in \mathbb{R}^{n\times n} : I \preceq M \preceq CI\}, \tag{4.6b}$$

$$\forall ((u, v, \sigma), x) \in \omega_N, \quad (A_\sigma x)^T P_v (A_\sigma x) \leq \gamma^2 x^T P_u x, \tag{4.6c}$$

for a large $C \geq n$. We denote $\gamma^*(\omega_N)$ and $P^*(\omega_N)$ as the solutions of program $\mathcal{P}(\omega_N)$. The problem (4.6), $\mathcal{P}(\omega_N)$ differs from problem (4.1), $\mathcal{P}(\omega_N)$ in two ways: the LMIs expressed in constraint (4.6c) are restricted to ω_N, and compactness of the domain of the matrices P_u is imposed in constraint (4.6b)[1].

[1]We will need this to prove Proposition 4.3.1

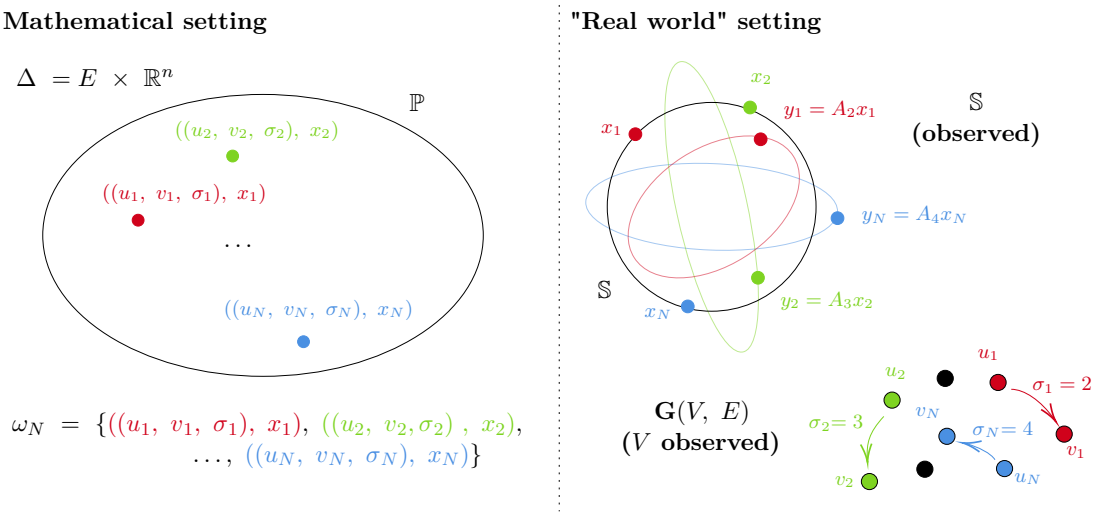


Figure 4.2: Illustration of the sampling strategy for the MQLF method. On the left, the sampling as it is mathematically defined in Equation (4.5): an observation is a point in Δ, sampled with measure $\mathbb{P} = \mathbb{P}_\sigma \times \mathbb{P}_\mathbb{S}$, where $\mathbb{P}_\sigma$ and $\mathbb{P}_\mathbb{S}$ are the uniform probability measure on respectively E and $\mathbb{S}$. On the right, the origin of this mathematical definition. One sample N couples of points $((u, x), (v, y))$ in $(V \times \mathbb{S})^2$ where $y = A_\sigma x$ and $(u, v, \sigma) \in E$. We invite the reader to compare this figure and its caption with Figure 3.1 and its caption in order to clearly understand the difference between the settings of the CQLF and MQLF methods.

As in Chapter 3, for a given set of observations ω_N, the problem tackled is the inference, with a user-defined confidence level, of $\gamma^*(\Delta)$ from $\gamma^*(\omega_N)$ and the set $\{P^*_u(\omega_N),\, u \in V\}$.

4.2 A deterministic lower bound

In the same fashion as in [KBJT19], we first derive a deterministic lower bound on the CJSR:

Proposition 4.2.1. *Let ω_N be a set of N observations from Δ as explained above. Consider the program (4.6), $\mathcal{P}(\omega_N)$ for the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ with optimal cost $\gamma^*(\omega_N)$. Then the following holds:*

$$\max\{2^{-h(\mathbf{G})/2}, n^{-1/2}\}\gamma^*(\omega_N) \leq \rho(\mathbf{G}, \mathbf{\Sigma}). \tag{4.7}$$

To the best of our knowledge, Proposition 4.2.1 provides the first **way to obtain from data deterministic sufficient conditions for the instability of a data-driven CSLS**.

Proof. Notice that $\mathcal{P}(\omega_N)$ defined in (4.6) is a relaxation of program 4.1, $\mathcal{P}(\Delta)$. As a consequence, we have $\gamma^*(\Delta) \geq \gamma^*(\omega_N)$. Following Corollary 1.3.1,

$$\rho(\mathbf{G}, \mathbf{\Sigma}) \geq \max\{2^{-h(\mathbf{G})/2}, n^{-1/2}\}\gamma^*(\Delta) \geq \max\{2^{-h(\mathbf{G})/2}, n^{-1/2}\}\gamma^*(\omega_N), \tag{4.8}$$

which is the desired result. □

We observe in Proposition 4.2.1 that, as for the first method (see Section 3.3 and Figure 3.5), the entropy also plays a role in the approximation of the CJSR. However, it does not play the same role, as Proposition 4.2.1 comes from a white-box result on the lower bound.

Remark 4.2.1. *Following [LPJ19, Corollary 1], one can show that the lower bound of Proposition 4.2.1 can be improved thanks to* Sums-of-squares approximation methods*, introduced in [PJ08] for the approximation of the* joint spectral radius *and generalized in [PEDJ16] for the CJSR. Generalizing this method to Sums-of-squares functions is thus considered as interesting further work.*

4.3 Generalization of the sensitivity analysis approach

In this section, we will generalize results for ASLSs introduced in [RWJ21] and explained in Section 2.3. First, we generalize and improve Lemma 2.3.1:

Proposition 4.3.1. *Consider the program* (4.1)*, $\mathcal{P}(\Delta)$ for the CSLS $S(\mathbf{G}(V, E), \mathbf{\Sigma})$ with optimal cost $\gamma^*(\Delta)$. There exists a set $\omega \subset \Delta$ with $|\omega| = |V|n(n+1)/2$ such that $\gamma^*(\omega) = \gamma^*(\Delta)$, where $\gamma^*(\omega)$ is the optimal cost of program $\mathcal{P}(\omega)$ as defined in Equation* (4.6)*.*

Proof. First, from the arguments in [RWJ21, Lemma 1], we claim that there exists $\omega \subset \Delta$ with $|\omega| = |V|n(n+1)/2 + 1$ such that $\gamma^*(\omega) = \gamma^*(\Delta)$. Now, we consider the problem $\mathcal{P}(\omega)$ as defined in (4.6). With a similar argument as the one in [BJW21, Theorem 2], we can conclude that the objective remains unchanged removing one of the points in ω. □

Remark 4.3.1. *There are two main differences between Proposition 4.3.1 and Lemma 2.3.1. First, the proposition is derived for CSLSs instead of ASLSs. Second, the cardinality of the set is the number of variables of the considered program* (4.6) ***minus one****, while it is the number of variables of the considered program* (2.3) *in Lemma 2.3.1.*

We now generalize Proposition 2.3.2. The following proposition provides a bound on the conservatism of the sampled problem (4.6), $\mathcal{P}(\omega_N)$, with respect to the white-box problem 4.1, $\mathcal{P}(\Delta)$ as a function of N, the number of sampled points:

Proposition 4.3.2. *Consider the program* (4.1)*, $\mathcal{P}(\Delta)$ for the CSLS $S(\mathbf{G}(V, E), \mathbf{\Sigma})$ with optimal cost $\gamma^*(\Delta)$. Let $\omega_N = \{(x_i, (u_i, v_i, \sigma_i)), i = 1, \ldots, N\}$ be a set of N samples*

from Δ as explained above. Suppose $N \geq |V|n(n+1)/2$. Then, for all $\varepsilon \in (0,1]$, with probability at least

$$\beta(\varepsilon, m, N) = 1 - |V| \left(\frac{n(n+1)}{2}\right) \left(1 - \frac{\varepsilon}{m|V|}\right)^N, \tag{4.9}$$

there exists a set $\omega'_N = \{(x'_i, (u_i, v_i, \sigma_i)), i = 1, \ldots, N\} \subset \Delta$ such that $\gamma^*(\omega'_N) = \gamma^*(\Delta)$ with $\|x_i - x'_i\| \leq \sqrt{2 - 2\delta(\varepsilon)}$.

Proof. The proof follows the same lines as the one of [RWJ21, Proposition 2] except for three points. First the number of variables of the problem is not the same. Second, given that the edges are sampled uniformly (see Section 4.1), the probability of sampling a certain label σ is at least $1/(m|V|)$, while it is $1/m$ in the unconstrained case. Third, Proposition 4.3.1 allows to improve the probability β according to Remark 4.3.1. □

We now apply a sensitivity analysis approach in order to obtain from Proposition 4.3.2 a probabilistic upper bound on $\gamma^*(\Delta)$ from $\gamma^*(\omega_N)$ and the matrices $P_u^*(\omega_N)$ for $u \in V$.

Theorem 4.3.3. *Consider the program 4.1, $\mathcal{P}(\Delta)$ for the CSLS $S(\mathbf{G}(V, E), \mathbf{\Sigma})$ with optimal cost $\gamma^*(\Delta)$. Let ω_N be a set of N samples from Δ as explained in Section 4.1, with $N \geq |V|n(n+1)/2$. Consider the sampled program 4.6, $\mathcal{P}(\omega_N)$ with solutions $\gamma^*(\omega_N)$ and $\{P_u^*(\omega_N),\, u \in V\}$. For any $\beta \in (0,1)$, let*

$$\varepsilon = m|V| \left(1 - \sqrt[N]{\frac{2(1-\beta)}{|V|n(n+1)}}\right). \tag{4.10}$$

Then, with probability at least β,

$$\gamma^*(\Delta) \leq \gamma^*(\omega_N) + \max_{((u,v,\sigma),x) \in \omega_N} \left\{ \sqrt{\frac{\lambda^u_{\max}}{\lambda^u_{\min}}} \gamma^*(\omega_N) + \sqrt{\frac{\lambda^v_{\max}}{\lambda^u_{\min}}} \mathcal{A}(\mathbf{\Sigma}) \right\} d(\varepsilon), \tag{4.11}$$

with $d(\varepsilon) = \sqrt{(2 - 2\delta(\varepsilon))}$, $\lambda^u_{\min}$ and $\lambda^u_{\max}$ respectively the minimal and maximal eigenvalue of $P_u^*(\omega_N)$, and

$$\mathcal{A}(\mathbf{\Sigma}) = \max_{A \in \mathbf{\Sigma}} \|A\|. \tag{4.12}$$

Proof. For the sake of readibility, let $\gamma = \gamma^*(\omega_N)$ and $P_u = P_u^*(\omega_N)$ for any $u \in V$. By definition, for any $((u, v, \sigma), x) \in \omega_N$,

$$\|A_\sigma x\|_{P_v} \leq \gamma \|x\|_{P_u}. \tag{4.13}$$

Consider now for any $P \in \mathcal{S}^n$ its *Cholesky decomposition* $P = L^T L$. Then the following holds:

$$\|x\|_P = \|Lx\| \leq \|L\| \|x\| \leq \sqrt{\lambda_{\max}(P)} \|x\|, \tag{4.14}$$

where $\lambda_{\max}(P)$ is the maximal eigenvalue of P. Let us now consider an arbitrary constraint $((u,v,\sigma),x) \in \Delta$, and define $y = x + \Delta x$ with $((u,v,\sigma),x) \in \omega_N$. Then, for any $((u,v,\sigma),x) \in \omega_N$, it holds that

$$\begin{aligned}
\|A_\sigma(x+\Delta x)\|_{P_v} &\leq \|A_\sigma x\|_{P_v} + \|A_\sigma \Delta x\|_{P_v} \\
&\leq \gamma \|x\|_{P_u} + \|A_\sigma \Delta x\|_{P_v} \\
&= \gamma \|(x+\Delta x) - \Delta x\|_{P_u} + \|A_\sigma \Delta x\|_{P_v} \\
&\leq \gamma \|x+\Delta x\|_{P_u} + \gamma \|\Delta x\|_{P_u} + \|A_\sigma \Delta x\|_{P_v} \\
&\leq \gamma \|x+\Delta x\|_{P_u} + \gamma \|\Delta x\| \sqrt{\lambda^u_{\max}} + \|A_\sigma\| \|\Delta x\| \sqrt{\lambda^v_{\max}} \\
&\leq \gamma \|x+\Delta x\|_{P_u} + \gamma \|\Delta x\| \sqrt{\lambda^u_{\max}} \frac{\|x+\Delta x\|_{P_u}}{\sqrt{\lambda^u_{\min}}} \\
&\qquad + \|A_\sigma\| \|\Delta x\| \sqrt{\lambda^v_{\max}} \frac{\|x+\Delta x\|_{P_u}}{\sqrt{\lambda^u_{\min}}} \\
&= \left[\gamma + \left(\sqrt{\frac{\lambda^u_{\max}}{\lambda^u_{\min}}} \gamma + \sqrt{\frac{\lambda^v_{\max}}{\lambda^u_{\min}}} \|A_\sigma\| \right) \|\Delta x\| \right] \|x+\Delta x\|_{P_u}.
\end{aligned} \tag{4.15}$$

For any $\beta \in (0,1)$, let ε be defined such as in Equation (4.10), then, given that $N \geq |V| n(n+1)/2$, Proposition 4.3.2 guarantees the existence of a set ω'_N with N points such that $\gamma^*(\omega'_N) = \gamma^*(\Delta)$ with probability at least β, and such that for any $((u,v,\sigma),x) \in \omega_N$, there exists Δx such that $((u,v,\sigma), x+\Delta x) \in \omega'_N$ and $\|\Delta x\| \leq d(\varepsilon)$. Hence, by definition and following Equation (4.15),

$$\begin{aligned}
\gamma^*(\Delta) &= \gamma^*(\omega'_N) \\
&\leq \gamma + \max_{((u,v,\sigma),x) \in \omega_N} \left\{ \sqrt{\frac{\lambda^u_{\max}}{\lambda^u_{\min}}} \gamma + \sqrt{\frac{\lambda^v_{\max}}{\lambda^u_{\min}}} \mathcal{A}(\mathbf{\Sigma}) \right\} d(\varepsilon),
\end{aligned} \tag{4.16}$$

with probability at least β. □

In order to get a fully data-driven bound as expressed in Equation (4.11), one still needs to approximate $\mathcal{A}(\mathbf{\Sigma})$, as $\mathbf{\Sigma}$ is an unknown quantity in the data-driven setting.

4.4 Estimation of the maximal norm

In this section, we derive a theorem that gives a probabilistic upper bound on $\mathcal{A}(\mathbf{\Sigma})$ as defined in Equation (4.12). First, note that the following holds [Jun09, Proposition 2.7]:

$$\mathcal{A}(\mathbf{\Sigma}) := \eta^*(\Delta) = \min_{\eta \geq 0} \text{ s.t. } \forall ((u,v,\sigma),x) \in \Delta, \quad \|A_\sigma x\| \leq \eta. \tag{4.17}$$

For a given set of observations ω_N, with the same idea as in Section 2.1, Section 3.1 and Section 4.1, let us infer the value of $\eta^*(\Delta) = \mathcal{A}(\mathbf{\Sigma})$ from the solution of its data-driven version

$$\eta^*(\omega_N) = \min_{\eta \geq 0} \text{ s.t. } \forall ((u,v,\sigma),x) \in \omega_N, \quad \|A_\sigma x\| \leq \eta, \tag{4.18}$$

with a user-defined confidence level.

Note that the quantity $\mathcal{A}(\mathbf{\Sigma})$ already appears in Theorem 2.3.3, although we decided not to present it in this thesis (see Remark 2.3.1). The reason is that authors of [RWJ21] use the general chance-constrained theorem [BJW21, Theorem 6], which requires a technical assumption [BJW21, Assumption 8] that can be violated in our setting. We give a proof for Theorem 4.4.1 allowing us to get rid of this assumption.

Theorem 4.4.1. *Let ω_N be a set of N samples from Δ as explained in Section 4.1. Consider the solutions $\eta^*(\Delta)$ and $\eta^*(\omega_N)$ defined in Equation* (4.17) *and Equation* (4.18) *respectively. For any $\beta' \in (0,1)$, let*

$$\varepsilon' = 1 - \sqrt[N]{1-\beta'}. \tag{4.19}$$

Then, with probability at least β',

$$\mathcal{A}(\mathbf{\Sigma}) = \eta^*(\Delta) \leq \frac{\eta^*(\omega_N)}{\delta(\varepsilon' m|V|/2)}. \tag{4.20}$$

Proof. Let the *violating set* $V(\eta) := \{((u,v,\sigma),x) \in \Delta : \|A_\sigma x\| > \eta\}$, and let $f : \mathbb{R} \to [0,1] : \eta \mapsto f(\eta) = \mathbb{P}[V(\eta)]$ be its measure. Note that f is decreasing. For any $\varepsilon' \in [0,1]$, we start by showing the following equation:

$$\mathbb{P}^N[\omega_N \subset \Delta : f(\eta^*(\omega_N)) \leq \varepsilon'] = 1 - (1-\varepsilon')^N. \tag{4.21}$$

Consider one sampled constraint $d \in \Delta$, and let $\eta_{\varepsilon'} \in \mathbb{R}$ be such that $f(\eta_{\varepsilon'}) = \varepsilon'$. Then $\mathbb{P}[d \in \Delta : f(\eta^*(\{d\})) > \varepsilon'] = \mathbb{P}[d \in \Delta : f(\eta^*(\{d\})) > f(\eta_{\varepsilon'})]$. Since f is decreasing and has $[0,1]$ as codomain, $\mathbb{P}[d \in \Delta : f(\eta^*(\{d\})) > f(\eta_{\varepsilon'})] = 1 - \varepsilon'$, hence the following holds:

$$\mathbb{P}[d \in \Delta : f(\eta^*(\{d\})) > \varepsilon'] = 1 - \varepsilon'. \tag{4.22}$$

Since samples in ω_N are i.i.d., the following holds:

$$\mathbb{P}^N[\omega_N \subset \Delta : f(\eta^*(\omega_N)) > \varepsilon'] = \left(\mathbb{P}[d \in \Delta : f(\eta^*(\{d\})) > \varepsilon']\right)^N = (1-\varepsilon')^N, \tag{4.23}$$

which is equivalent to Equation (4.21).

Now, define the projected violating set $\tilde{\mathbb{S}} \subseteq \mathbb{S}$ as follows:

$$\tilde{\mathbb{S}} = \{x \in \mathbb{S} : \exists (u,v,\sigma) \in E, \|A_\sigma x\| > \eta^*(\omega_N)\}. \tag{4.24}$$

For any $(u,v,\sigma) \in E$, we define:

$$\tilde{\mathbb{S}}_{(u,v,\sigma)} = \{x \in \mathbb{S} : \|A_\sigma x\| > \eta^*(\omega_N)\}. \tag{4.25}$$

Thus, $\tilde{\mathbb{S}} = \cup_{(u,v,\sigma)\in E} \tilde{\mathbb{S}}_{(u,v,\sigma)}$. In the worst case, the sets $\{\tilde{\mathbb{S}}_{(u,v,\sigma)}\}$ are disjoint. In this case, $\mathbb{P}_x[\tilde{\mathbb{S}}] = \sum_{(u,v,\sigma)\in E} \mathbb{P}_x[\tilde{\mathbb{S}}_{(u,v,\sigma)}]$ and

$$\begin{aligned} \mathbb{P}[V(\eta)] &= \sum_{(u,v,\sigma)\in E} \mathbb{P}_x[\tilde{\mathbb{S}}_{(u,v,\sigma)}]\mathbb{P}_E[\{(u,v,\sigma)\}] \\ &\geq \frac{1}{m|V|} \sum_{(u,v,\sigma)\in E} \mathbb{P}_x[\tilde{\mathbb{S}}_{(u,v,\sigma)}] = \frac{\mathbb{P}_x[\tilde{\mathbb{S}}]}{m|V|}, \end{aligned} \tag{4.26}$$

where $\mathbb{P}_x$ and $\mathbb{P}_E$ denote the uniform (probability) measure on $\mathbb{S}$ and E respectively. This means that $\mathbb{P}[V(\eta)] \leq \varepsilon'$ implies $\mathbb{P}_x[\tilde{\mathbb{S}}] \leq \varepsilon' m|V|$.

The rest of the proof follows the same lines as the proof of [KBJT19, Theorem 15]. □

The probabilistic approximation of $\mathcal{A}(\mathbf{\Sigma})$ as stated in Theorem 4.4.1 allows to directly derive the following corollary, which provides a pratical probabilistic bound on the CJSR of a given CSLS given a set of observations.

Corollary 4.4.1. *Consider the program* (4.1)*, $\mathcal{P}(\Delta)$ for the CSLS $S(\mathbf{G}(V,E),\mathbf{\Sigma})$ with optimal cost $\gamma^*(\Delta)$. Let ω_N be a set of N samples from Δ as explained in Section 4.1, with $N \geq |V|n(n+1)/2$. Consider the sampled program* (4.6)*, $\mathcal{P}(\omega_N)$ with solution $\gamma^*(\omega_N)$ and $\{P_u^*(\omega_N), u \in V\}$. For any $\beta, \beta' \in (0,1)$, let*

$$\varepsilon = m|V|\left(1 - \sqrt[N]{\frac{2(1-\beta)}{|V|n(n+1)}}\right), \tag{4.27}$$

and

$$\varepsilon' = 1 - \sqrt[N]{1-\beta'}. \tag{4.28}$$

Then, with probability at least $\beta + \beta' - 1$,

$$\rho(\mathbf{G},\mathbf{\Sigma}) \leq \gamma^*(\omega_N) + \max_{(x,(u,v,\sigma))\in\omega_N} \left\{ \sqrt{\frac{\lambda^u_{\max}}{\lambda^u_{\min}}}\gamma^*(\omega_N) + \sqrt{\frac{\lambda^v_{\max}}{\lambda^u_{\min}}}\frac{\eta^*(\omega_N)}{\delta(m|V|\varepsilon'/2)} \right\} d(\varepsilon), \tag{4.29}$$

with $d(\varepsilon) = \sqrt{2-2\delta(\varepsilon)}$, and, for $u \in V$, $\lambda^u_{\min}$ and $\lambda^u_{\max}$ respectively the minimal and maximal eigenvalue of $P_u^*(\omega_N)$

Based on a sensitivity analysis approach, Corollary 4.4.1 provides an **alternative way to obtain probabilistic stability certificate for data-driven CSLSs.**

Proof. Following Theorem 1.3.3, Equation (4.29) holds if Equation (4.11) and Equation (4.20) both hold. Theorem 4.3.3 states that Equation (4.11) holds with probability β, and Theorem 4.4.1 states that Equation (4.20) holds with probability β'. Thus

$$\begin{aligned}
&\mathbb{P}^N[\omega_N \subset \Delta : (4.11) \text{ and } (4.20) \text{ hold}] \\
&= 1 - \mathbb{P}^N[\omega_N \subset \Delta : (4.11) \text{ or } (4.20) \text{ does not hold}] \\
&\geq 1 - \mathbb{P}^N[\omega_N \subset \Delta : (4.11) \text{ does not hold}] - \mathbb{P}^N[\omega_N \subset \Delta : (4.20) \text{ does not hold}] \\
&\geq 1 - (1-\beta) - (1-\beta') \\
&= \beta + \beta' - 1,
\end{aligned} \tag{4.30}$$

which concludes the proof. □

4.5 Conclusions

In this chapter, we presented a second method to derive probabilistic guarantee for stability of CSLSs. This method, compared to the one presented in Chapter 3, does not require the CSLS to possess a common Lyapunov function, but requires that we are able to observe the nodes of the underlying automaton.

We leveraged the sensitivty analysis approach such as presented in Section 2.3. As for the first method presented in Chapter 3, we used the CJSR as a tool to approximate the black-box stability of CSLSs. In particular, we provided a deterministic lower bound on the CJSR, as well as a probabilistic upper bound on it. We also derived a probabilistic guarantee on the maximal norm among the norms of matrices driving the considered CSLS. The closed form of the approximation presented in Equation (4.29) shows that a we obtain tighter approximations of the CJSR for a large number of samples, but also for smaller confidence levels.

Part III

Comparisons and examples

Chapter 5

Qualitative and quantitative analysis

In this chapter, we will provide qualitative, as well as quantitative comparisons between the CQLF method presented in Chapter 3 and the MQLF method presented in Chapter 4.

The purpose is twofold. First, from the theory derived in Chapter 3 and Chapter 4, we want to characterize the limitations of each method, and precise when a method is more limited than another. In order to do this, we will distinguish four different cases. We will particularly focus on the importance of the trace length when using the CQLF method. We will provide simulations in order to illustrate and quantitatively verify these assertions. Second, we want to analyze further the two bounds. We will analyze the influence of certain parameters on the behaviours of both bounds.

5.1 Qualitative comparisons

In this section, we will provide a qualitative comparison between the CQLF and the MQLF method.

First, as a general comment, both methods come with their assumptions. Indeed, it may be difficult to use the CQLF method in practice. Indeed, Theorem 3.13 requires the knowledge of $p_{l,\min}$, the minimal probability among probabilities to draw matrices in $\mathbf{\Pi}_l$, the set of all possible matrices products of length l. Moreover, corollaries presented in Section 3.3 (Corollary 3.3.1 and Corollary 3.3.2) both come with strong assumptions and prior knowledge, namely the entropy and the largest eigenvalue of the adjacency matrix of the automaton. On the other hand, the MQLF method is based on the observation of the nodes of the automaton, while the latter is considered as unknown in the data-driven setting. Moreover, unlike the CQLF method, it was not designed to handle traces of length $l > 1$.

Now, before going further on the difference between the four different cases mentioned above, let us note the following inequalities[1].

Proposition 5.1.1. *Consider the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$, let $\mathbf{\Pi}_l$ be the set of accepted product of matrices of length l. Then, for any trace length l,*

$$\rho(\mathbf{\Sigma}) = \rho(\mathbf{\Pi}_1) \geq \rho(\mathbf{\Pi}_2)^{1/2} \geq \cdots \geq \rho(\mathbf{\Pi}_l)^{1/l} \geq \rho(\mathbf{G}, \mathbf{\Sigma}). \tag{5.1}$$

It means that a certain l-lifting $S(\mathbf{\Pi}_l)$ of a CSLS (as defined in Chapter 3) may allow or not to find a stability certificate. Let us introduce in Figure 5.1 four different cases.

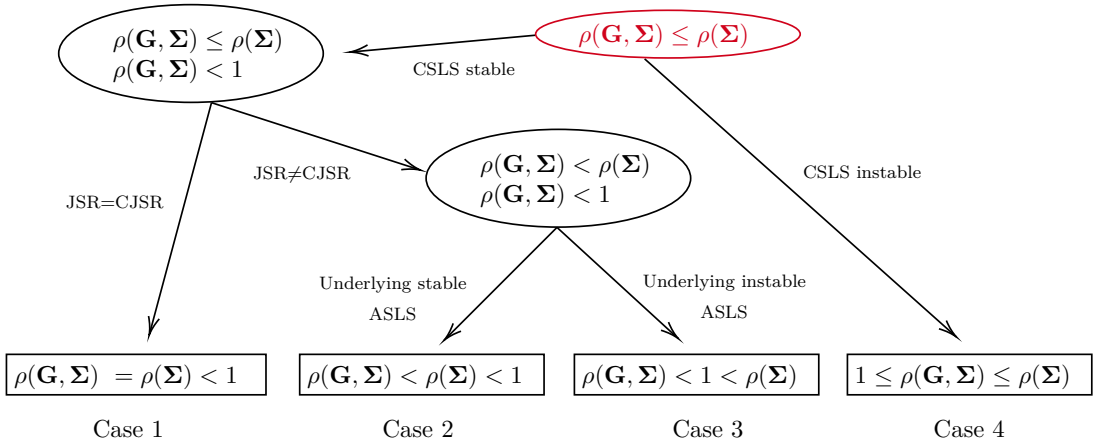


Figure 5.1: Four cases considered in Chapter 5.

In the first case of Figure 5.1, the 1-lift will provide the right approximation for the CJSR. **In this case, the MQLF method is not less limited than the CQLF method**. The choice of the method thus mainly depends on the assumptions and the prior knowledge, or on a heuristic.

In the second case, the choice of the method depends on the goal of its application. Suppose that the application is to find a stability certificate, then the CQLF method may provide one, for each value of l greater **or equal** than 1. We know that the model-based CQLF approximation of the JSR $S(\mathbf{\Sigma})$, i.e. $\gamma^*(\Delta)$ as defined in Section 3.1, belongs to the interval $[\rho(\mathbf{\Sigma}), \min\{m^{1/2}, n^{1/2}\}\rho(\mathbf{\Sigma})]$ (see Theorem 1.2.4). Thus, as long as $\rho(\mathbf{\Sigma}) < \max\{n^{-1/2}, m^{-1/2}\}$, it is guaranteed for N large enough to find a stability certificate **for all** l-liftings with the CQLF method, as $\gamma^*(\omega_N)$ tends to $\gamma^*(\Delta)$ for N going to infinity. However, suppose that the application is the approximation of the CJSR, and not only the stability certificate. Then, assuming that enough samples are harvested, the usage of the MQLF is less limited. An illustration of this case is provided in Figure 5.2.

[1]Proving this monoticity property remains an open problem, we conjectured it from our observations in the simulations.

In the third case, suppose that only a dataset with horizon 1 is available, then **it is impossible for the CQLF to provide a stability certificate**. In this case, there is an integer $\tilde{l} > 1$ such that

$$\rho(\mathbf{\Sigma}) = \rho(\mathbf{\Pi}_1) \geq \cdots \geq \rho(\mathbf{\Pi}_{\tilde{l}-1})^{1/(\tilde{l}-1)} \geq 1 > \rho(\mathbf{\Pi}_{\tilde{l}})^{1/\tilde{l}} \geq \cdots \geq \rho(\mathbf{G}, \mathbf{\Sigma}). \tag{5.2}$$

The usage of the CQLF method will thus depend on the dataset. Suppose that $\tilde{l}$ is very large, then it is possible that, in practice, a $\tilde{l}$-horizon dataset is available for a given application. The MQLF method, on the other hand, will directly tend to the approximation of the CJSR, i.e. $\gamma^*(\Delta)$, solution of program (4.1). An illustration of this case is provided in Figure 5.3

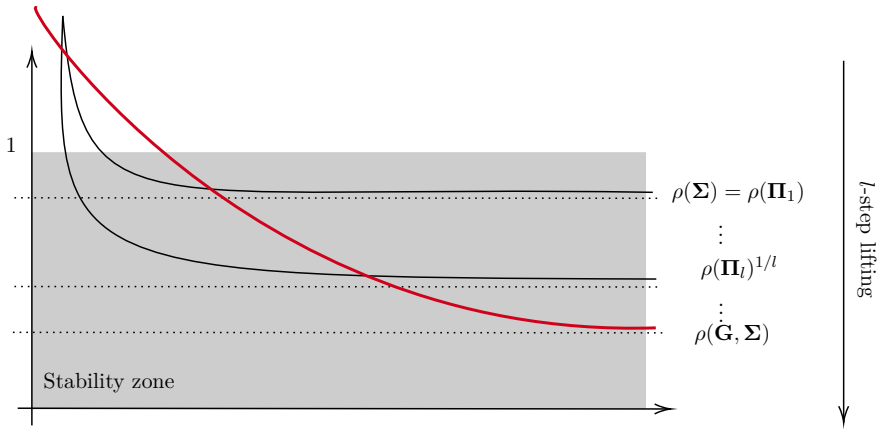


Figure 5.2: Illustration of the second case of the Figure 5.1. In red, the MQLF method. In black, the CQLF method for $l = 1$ and $l > 1$. One can obtain a stability certificate with any l-lifting, but it may not be a good approximation of the CJSR.

Finally, in the fourth case, **only the MQLF method may provide a deterministic instability certificate**. Indeed, it is not straightforward to derive a lower bound on the CJSR with the CQLF method, where it is the case for the MQLF method (see Section 4.2 and Proposition 4.2.1).

As explained above, even if the CQLF method proves to be more efficient than the MQLF method, **its limitations highly depend on the trace length l of the available observations**, and, depending on the situation, **may be more conservative**.

5.2 Quantitative comparisons

In this section, we will perform numerical experiments on examples. The goal is twofold. First, we want to verify that we indeed observe the qualitative remarks stated in Section 5.1.

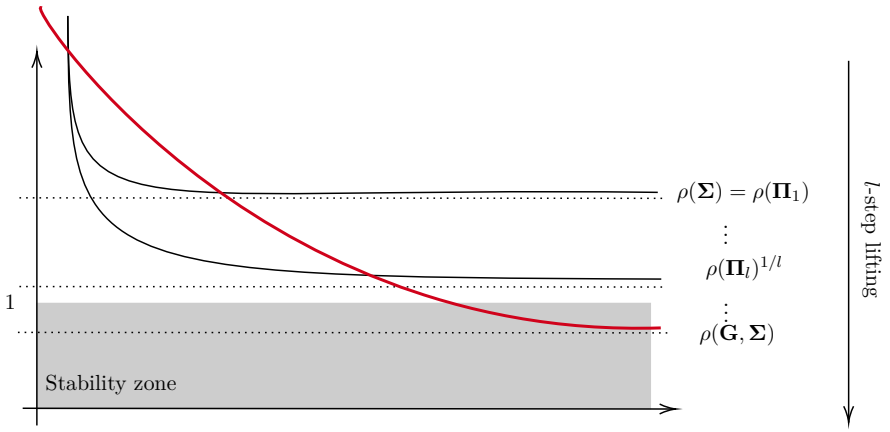


Figure 5.3: Illustration of the third case of the Figure 5.1. In red, the MQLF method. In black, the CQLF method for $l = 1$ and $l > 1$. Obtaining a stability certificate depends on the lifting with the CQLF method.

Then, we analyze further the bounds on different examples where we vary different parameters[2].

5.2.1 Illustrations of the qualitative comparison

First, note that, in this section, we use the general theorem for the CQLF bound (see Theorem 3.13), we assume $p_{l,\min}$ as known. For every comparaison below, unless otherwise explicitly stated, we choose the parameters given in Table 5.1[3].

Parameter	Value
Dimension n	2
Number of matrices m	2
Trace length l	1
Number of nodes $\lvert V \rvert$	2
Overall confidence level β	99%
"Sub-confidence" levels $(100\% + \beta)/2$	99.5%

Table 5.1: Parameters for the quantitative analysis, unless otherwise stated.

Note that, following the qualitative analysis, we chose $l = 1$ in order to highlight the limitations of the CQLF method. We present the probabilistic bounds computed with the CQLF and MQLF method with the parameters presented in Table 5.1 in Figure 5.4,

[2]These simulation were performed thanks to the `Julia` code attached to this thesis. For an overview of the code and implemented functions, see Appendix B.

[3]"Sub-confidence levels" states for β and β' in Theorem 4.3.3. Suppose we want to reach an overall confidence level of $\beta + \beta' - 1 = \overline{\beta}$ with $\beta = \beta'$, then $\beta = \beta' = (1 + \overline{\beta})/2$.

Figure 5.5, Figure 5.6, and Figure 5.7. The considered examples (described below) indeed allow to verify the assertions made in Section 5.1.

We describe examples corresponding to the four cases that have been analyzed in Section 5.1 and as illustrated in Figure 5.1. We start with Example 5.2.1, corresponds to the first case, where the CSLS is stable, and the JSR of the underlying JSR is the same as the CJSR of the CSLS.

Example 5.2.1. *Consider the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$, with the set $\mathbf{\Sigma} = \{A_1, A_2\}$ defined as*

$$A_1 = \begin{pmatrix} -0.1165 & -0.4876 \\ -0.4866 & 0.7275 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -0.9524 & 0.3378 \\ -0.0104 & 0.4850 \end{pmatrix}, \tag{5.3}$$

and $\mathbf{G}$ as depicted in Figure 3.2. The matrices were chosen such that $\rho(\mathbf{\Sigma}) = 0.95$ (verified with the `Julia` toolbox presented in Appendix B). Also, using [PEDJ16, Lemma 3.7], the sequence of edges $((a, b, 2), (b, a, 2))$ is a cycle and $\sqrt{\rho(A_2^2)} = 0.95$ yields $\rho(\mathbf{G}, \mathbf{\Sigma}) \geq 0.95$. Thus $\rho(\mathbf{G}, \mathbf{\Sigma}) = 0.95$. $l = 1$ implies that the minimal probability among probabilities to draw is matrix is the probability to draw A_1, which is $p_{1,\min} = 1/4$.

We now present Example 5.2.2, corresponding to the second case presented in Figure 5.1. In this case, $\rho(\mathbf{G}, \mathbf{\Sigma}) < \rho(\mathbf{\Sigma}) < 1$.

Example 5.2.2. *Consider the CSLS $S(\mathbf{G}, \mathbf{\Sigma})$, with the set $\mathbf{\Sigma} = \{A_1, A_2\}$ defined as*

$$A_1 = \begin{pmatrix} -0.0613 & -0.2566 \\ -0.2561 & 0.3828 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -0.9023 & 0.3201 \\ -0.0098 & 0.4595 \end{pmatrix}, \tag{5.4}$$

and $\mathbf{G}$ as depicted in Figure 5.8.

Matrices were generated such that $\rho(A_1) = 0.5$ and $\rho(A_2) = 0.9$. The JSR of the underlying ASLS $\rho(\mathbf{\Sigma}) = 0.9$ happens with the word $(\sigma(0), \sigma(1), \dots) = (1, 1, \dots)$. This JSR was verified with our `Julia` toolbox (see Appendix B). However, the CJSR $\rho(\mathbf{G}, \mathbf{\Sigma}) = 0.5$. An upper bound $\rho(\mathbf{G}, \mathbf{\Sigma}) \lesssim 0.5$ was found with the `Julia` white-box toolbox, and [PEDJ16, Lemma 3.7] yields $\rho(\mathbf{G}, \mathbf{\Sigma}) \geq \rho(A_1) = 0.5$. The minimal probability $p_{1,\min} = 1/4$ and corresponds to the sampling of the second mode.

We continue by presenting Example 5.2.3, corresponding to the third case of Figure 5.1. This case corresponds to the case where $\rho(\mathbf{G}, \mathbf{\Sigma}) < 1$, but the JSR of the underlying ASLS $\rho(\mathbf{\Sigma}) \geq 1$.

Example 5.2.3. *Consider the same CSLS as the one presented in Example 5.2.2 except that A_2 is defined as*

$$A_2 = \begin{pmatrix} -1.5038 & 0.5334 \\ -0.0164 & 0.7659 \end{pmatrix}. \tag{5.5}$$

Thus $\rho(A_2) = 1.5 \geq 1$, and with a very similar reasoning as for Example 5.2.2, $\rho(\mathbf{G}, \mathbf{\Sigma}) = 0.5$ and $\rho(\mathbf{\Sigma}) = 1.5$. $p_{1,\min}$ remains $1/4$.

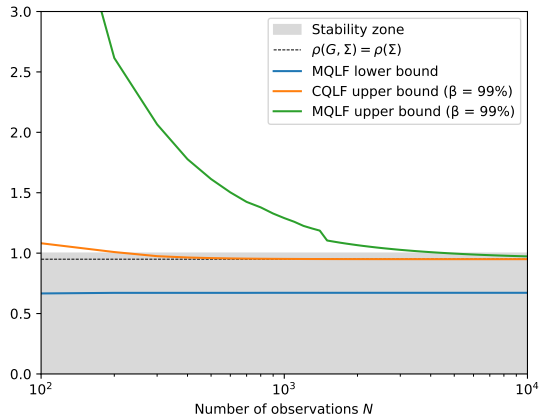


Figure 5.4: Bounds found by the two methods for the first case example (Example 5.2.1), and the parameters described in Table 5.1, from $N = 100$ to $N = 10000$. Both methods tend to the same value, despite that $l = 1$. The CQLF method seems to converge faster, and finds a 99% stability certificate faster.

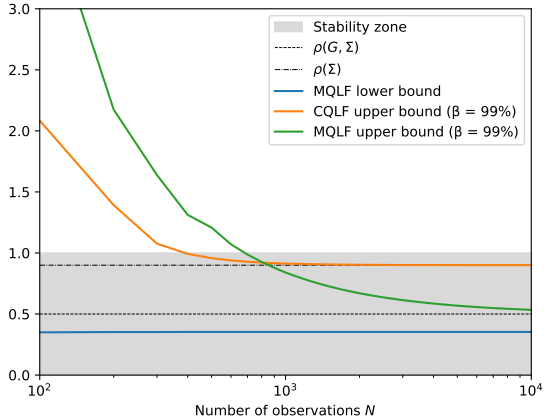


Figure 5.5: Bounds found by the two methods for the second case example (Example 5.2.2), and the parameters described in Table 5.1, from $N = 100$ to $N = 10000$. If the goal is to find a stability certificate, then the CQLF method is faster, even for $l = 1$. If the goal is to approximate the CJSR, the MQLF method is less limited, the CQLF method would require a larger trace length l.

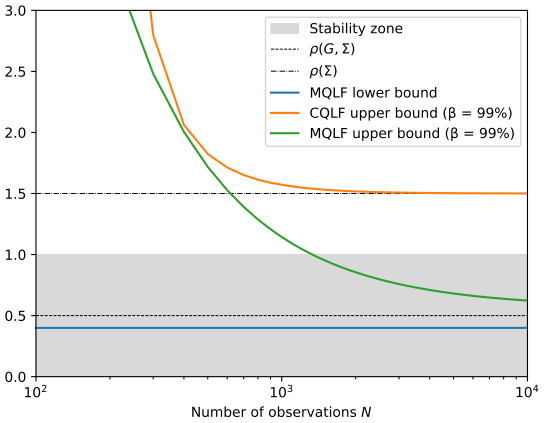


Figure 5.6: Bounds found by the two methods for the third case example (Example 5.2.3), and the parameters described in Table 5.1, from $N = 100$ to $N = 10000$. The MQLF method finds 99% stability certificate, unlike CQLF method with $l = 1$.

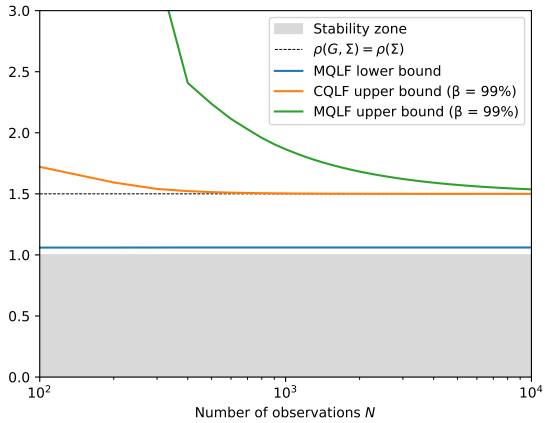


Figure 5.7: Bounds found by the two methods for the fouth case example (Example 5.2.4), and the parameters described in Table 5.1, from $N = 100$ to $N = 10000$. A deterministic instability certificate is directly found by the MQLF method.

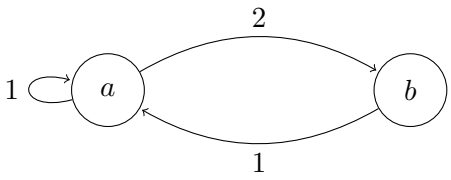


Figure 5.8: Automaton $\mathbf{G}$ for the second case considered in Chapter 5. If the second mode is chosen then it is only chosen once.

We finish by presenting Example 5.2.4, corresponding to the case where the CSLS is instable.

Example 5.2.4. *The instable CSLS considered in this example is the same as the CSLS considered in Example 5.2.1 except that the matrices are*

$$A_1 = \begin{pmatrix} -0.1839 & -0.7699 \\ -0.7684 & 1.1487 \end{pmatrix}, \tag{5.6}$$

and A_2 as defined in Equation (5.5)*. A similar reasoning as for Example 5.2.1 allows to conclude that $\rho(\mathbf{G}, \mathbf{\Sigma}) = \rho(\mathbf{\Sigma}) = 1.5$, and $p_{l,\min}$ remains $1/4$.*

We present an additional numerical result in Figure 5.9 in order to confirm the expected behaviours as theoretically predicted in Figure 5.2 and Figure 5.3, for trace lengths $l > 1$. Example 5.2.5 describes the CSLS corresponding to Figure 5.9.

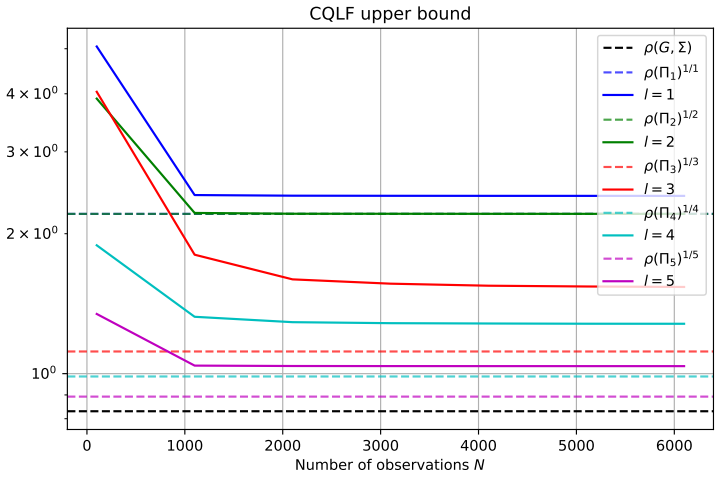


Figure 5.9: Illustration of the limitation of the CQLF method for small trace lengths l. Results of the CQLF upper bound for a CSLS for which the value $\tilde{l} = 4$ such as defined in Equation (5.2): $\rho(\mathbf{\Pi}_4)^{1/4} < 1 \leq \rho(\mathbf{\Pi}_3)^{1/3}$. With $l < 4$, it is impossible for the CQLF method to provide a stability probabilistic certificate.

Example 5.2.5. *Consider a CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ with $\mathbf{G}$ as depicted in Figure 5.10, $|\mathbf{\Sigma}| = m = 5$, and where the matrices are such that $\rho(\mathbf{G}, \mathbf{\Sigma}) \approx 0.8300$, and that the l-liftings yield the values in Table 5.2 for $l = 1, \dots, 5$. These values were computed with our* `Julia` *white-box toolbox (see Appendix B for more informations).*

Trace length l	$\rho(\mathbf{\Pi}_l)^{1/l}$	$\lvert\rho(\mathbf{\Pi}_l)^{1/l} - \rho(\mathbf{G}, \mathbf{\Sigma})\rvert / \rho(\mathbf{G}, \mathbf{\Sigma})$
1	2.2063	1.6580
2	2.2063	1.6580
3	1.1160	0.3445
4	0.98611	0.1880
5	0.8925	0.0751

Table 5.2: l-liftings approximations of the CJSR and the corresponding relative error for $l = 1, \dots, 5$.

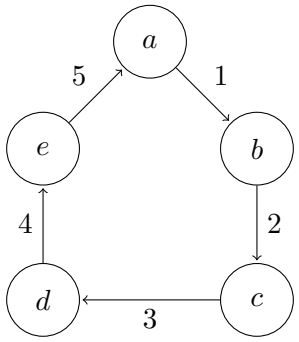


Figure 5.10: Automaton $\mathbf{G}$ for Example 5.2.5. Cycle with 5 nodes.

Numerical experiments verify the assertions made in Section 5.1. These simulations show the limitations of the CQLF in certain cases, depending on the order of the lifting, and give a first overview on the speed of each method.

5.2.2 Further analysis

In this section, we will study the influence of some parameter present in Table 5.1 on the results of each method. We will thus study the influence of the dimension and the number of nodes. Unless otherwise stated, parameters remain the same as those exposed in Table 5.1. Examples for each experiment are described in Appendix C.

Dimension The results of the simulations with dimensions ranging from $n = 2$ to $n = 5$ of the probabilistic upper bound, deterministic lower bound with MQLF method and probabilistic upper bound with CQLF method are respectively depicted in Figure 5.11, Figure 5.12 and Figure 5.13. One can observe that for both methods, a smaller dimension allows faster for a better approximation. The shape of the lower bound is a direct

consequence of Proposition 4.2.1. Concerning the CQLF method, this has already been analyzed in [KBJT19, Section 5.1, Figure 5, Figure 6 and Figure 7]. The same analysis can be made here. We can also observe that the CQLF method yields a much better upper bound for the same values of N. This is particularly the case since we are in the first case of Figure 5.1 (see Section 5.1 and Appendix C).

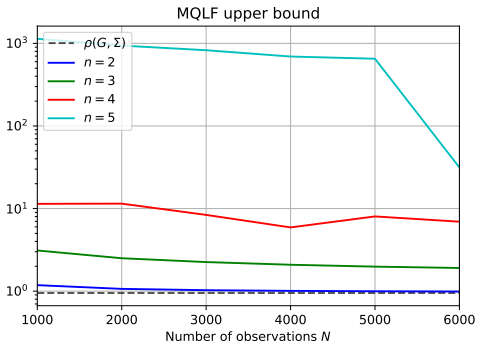


Figure 5.11: Influence of the dimension on the MQLF probabilistic upper bound. A lower dimension allows for a better performance.

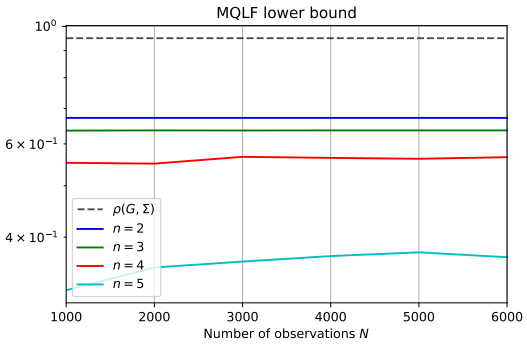


Figure 5.12: Influence of the dimension on the MQLF deterministic lower bound. A lower dimension improves the approximation.

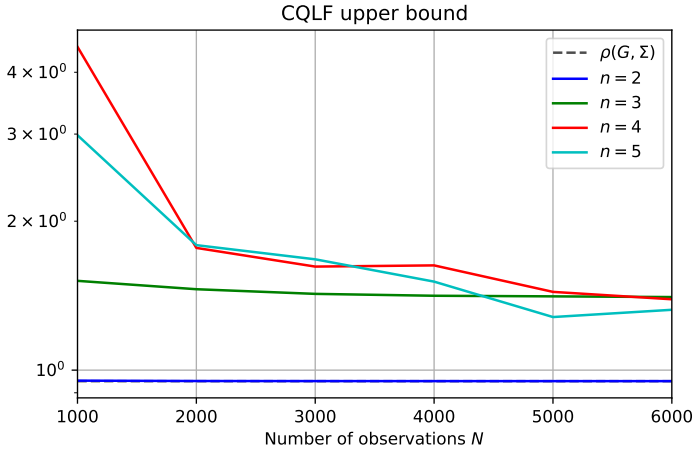


Figure 5.13: Influence of the dimension on the CQLF probabilistic upper bound. One can observe the difference between $n = 2$ and larger dimensions.

Number of nodes Results of the simulations with numbers of nodes ranging from $|V| = 2$ to $|V| = 5$ of the probabilistic upper bound and the deterministic lower bound of the MQLF method are respectively depicted in Figure 5.14, Figure 5.15. Automata for these examples are described in Appendix C. We did not perfom experiments for

the CQLF method as the quantity $|V|$ does not influence this bound. We can observe that smaller automata (automata with less nodes) allow for a better probabilistic upper bound, and thus a better approximate of the CJSR. As expected, the lower bound is not impacted by the number of nodes.

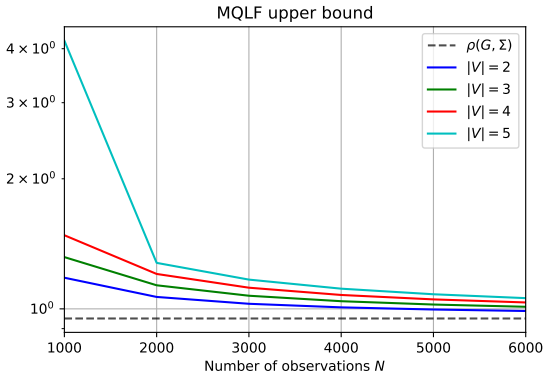


Figure 5.14: Influence of the number of modes on the MQLF probabilistic upper bound. An automaton with less nodes allows for a better approximate of the CJSR.

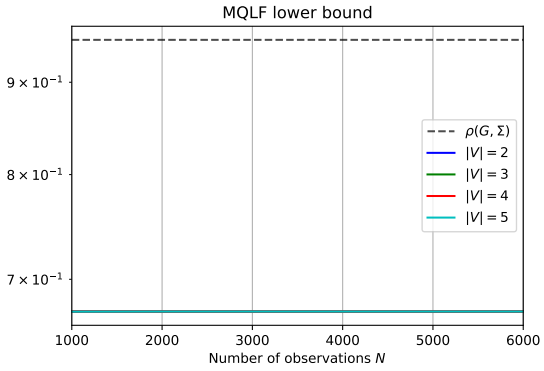


Figure 5.15: Influence of the number of modes on the MQLF deterministic lower bound. The number of nodes does not impact the lower bound.

Chapter 6

An application: multi-zone building systems

In this chapter, we present an application inspired from [WBJ21], namely *multi-zone building systems* (see e.g. [FLC16, CFM+19, BSH21]). We will consider a classical *Resistance-Capacitance* model that allows to model heat exchange between multiple zones in a building, with the presence of an *air conditioning* device. Some zone have doors, that can be arbitrarily opened and closed. We will assume that the temperature is controlled via the air conditoning device (via a model-based LQR controller, see Appendix A) to quickly reach a target temperature.

We then suppose that the controller fails, and that the latter happens with some constraints. We also assume that the model and the constraints are unknown. Although the physics tell us that the temperature never explodes, we want to compute the convergence rate of this closed-loop system. The problem is the latter: **from harvested data, with a user-defined confidence level, can we predict how fast the temperature will converge to the target temperature?** It is equivalent to derive a probabilistic approximation of the CJSR of the considered system. We will use tools derived in Chapter 3 and Chapter 4 in order to answer.

6.1 Presentation of the numerical example

In this section, we formally present the dynamical system tackled in this chapter.

6.1.1 Resistance-Capacitance model

Consider a building with n zones. Each zone is equiped with an *air conditioning* (*AC* for short) device capable of sending a given air flow at a given temperature. The thermal *Resistance-Capacitance* model (*RC* model for short) for this is the following

[FLC16, BSH21]:

$$C_i\dot{T}_i(t) = \frac{T^{\mathrm{o}} - T_i(t)}{R_i^{\mathrm{o}}} + \sum_{j\in[n]\setminus\{i\}} \frac{T_j(t) - T_i(t)}{R_{ji}} + \dot{m}_i C_p(T_i^{\mathrm{s}}(t) - T_i(t)) + q_i, \tag{6.1}$$

for all zones $i \in [n]$. The quantities involved in the model (6.1) are explained and given with their units Table 6.1.

Symbol	Unit	Physical quantity
C_i	$[\mathrm{kJK}^{-1}]$	Thermal capacitance of zone i
$T_i(t)$	[K]	Temperature of zone i at time t
T^{o}	[K]	Temperature of outside environment
R_{ij}	$[\mathrm{KkW}^{-1}]$	Thermal resistance between zone i and j
R_i^{o}	$[\mathrm{KkW}^{-1}]$	Thermal resistance between zone i and outside environment
$\dot{m}_i$	$[\mathrm{kgs}^{-1}]$	Flow rate into zone i
C_p	$[\mathrm{kJkg}^{-1}\mathrm{K}^{-1}]$	Specific heat capacity of air
$T_i^{\mathrm{s}}(t)$	[K]	Temperature of air sent by AC device to zone i at time t
q_i	[kJ/s]	Thermal disturbance from internal loads

Table 6.1: Quantities involved in Equation (6.1).

A representation of this RC model is provided in Figure 6.1.

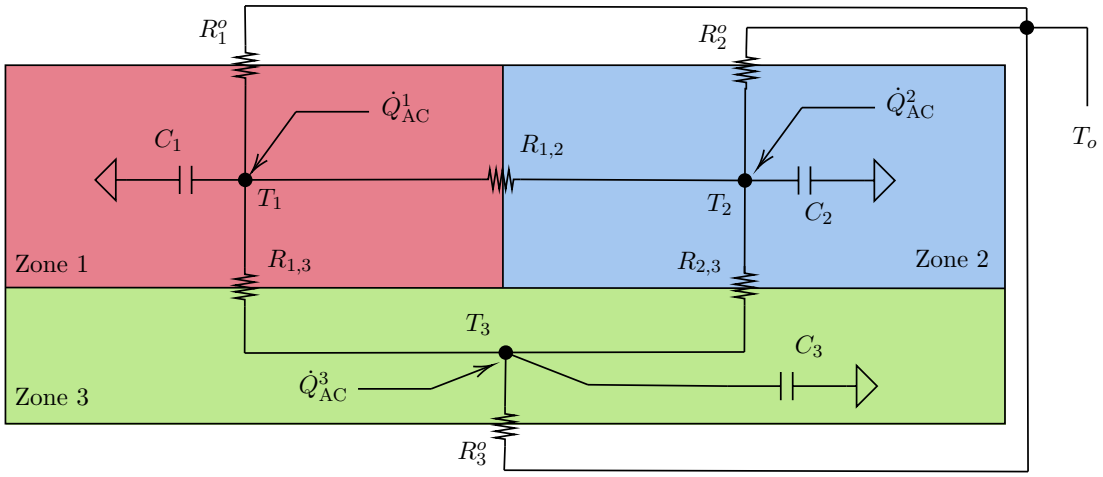


Figure 6.1: Representation of the RC model corresponding to a multi-zone building system with three zones. For $i = 1, 2, 3$, $\dot{Q}^i_{\mathrm{AC}} = \dot{m}_i C_p(T_i^{\mathrm{s}} - T_i) + q_i$.

As we focus on discrete-time switching systems in this thesis, we discretize the system. Euler's forward method (see [CFM+19] for more information about discretizations for

building systems) yields

$$\frac{C_i}{\tau}(T_i(k+1)-T_i(k)) = \frac{T^{\mathrm{o}}-T_i(k)}{R_i^{\mathrm{o}}} + \sum_{j\in[n]\setminus\{i\}} \frac{T_j(k)-T_i(k)}{R_{ji}} + \dot{m}_i C_p (T_i^{\mathrm{s}}(k)-T_i(k)) + q_i, \tag{6.2}$$

for all zones $i \in [n]$, where τ is the time step for the forward method, i.e. $t = k\tau$ for $k \in \mathbb{Z}$. It yields that the considered system satistifes the following recurrence:

$$\begin{aligned} T_i(k+1) &= T_i(k) \\ &+ \frac{\tau}{C_i}\left(\frac{T^{\mathrm{o}}-T_i(k)}{R_i^{\mathrm{o}}} + \sum_{j\in[n]\setminus\{i\}} \frac{T_j(k)-T_i(k)}{R_{ji}} + \dot{m}_i C_p (T_i^{\mathrm{s}}(k)-T_i(k)) + q_i\right), \end{aligned} \tag{6.3}$$

for all zones $i \in [n]$.

The considered numerical values are written in Table 6.2.

Quantity	Value
C_i, $i = 1, 2, 3$	1375 [kJK^{-1}]
T^{o}	305.15 [K]
R_{12}	1.5 [KkW^{-1}]
R_i^{o}, $i = 1, 2$	3 [KkW^{-1}]
R_3^{o}	2.7 [KkW^{-1}]
$\dot{m}_i$, $i = 1, 2, 3$	0.14 [kgs^{-1}]
C_p	1.012 [kJkg^{-1}K^{-1}]
q_i, $i = 1, 2$	0.1 [kJ/s]
q_3	0.12 [kJ/s]

Table 6.2: Numerical values considered for the multi-zone building system example.

6.1.2 Input of the model

The only quantity that we can make vary in this model is $T_i^{\mathrm{s}}(k)$ for all i, the temperature given by the AC device. Suppose we want to reach a target temperature of T^{t} [K] for all zone i. By letting $x_i(k) = T_i(k) - T^{\mathrm{t}}$, we can re-write Equation (6.3) as

$$\begin{aligned} x_i(k+1) &= x_i(k) \\ &+ \frac{\tau}{C_i}\left(\frac{(T^{\mathrm{o}}-T^{\mathrm{t}})-x_i(k)}{R_i^{\mathrm{o}}} + \sum_{j\in[n]\setminus\{i\}} \frac{x_j(k)-x_i(k)}{R_{ji}} + \dot{m}_i C_p (T_i^{\mathrm{s}}(k)-T_i(k)) + q_i\right), \end{aligned} \tag{6.4}$$

or equivalently

$$\begin{aligned} x_i(k+1) = x_i(k) + \frac{\tau}{C_i} \sum_{j \in [n] \setminus \{i\}} \frac{x_j(k) - x_i(k)}{R_{ji}} - \frac{\tau}{C_i} \frac{x_i(k)}{R_i^{\mathrm{o}}} \\ + \frac{\tau}{C_i} \left(\frac{(T^{\mathrm{o}} - T^{\mathrm{t}})}{R_i^{\mathrm{o}}} + \dot{m}_i C_p (T_i^{\mathrm{s}}(k) - T_i(k)) + q_i \right), \end{aligned} \tag{6.5}$$

for all zones $i \in [n]$.

Suppose we can measure the temperature $T_i(t)$, then we can consider that, in order to control the dynamical system written defined in Equation (6.1), for all zone $i \in [n]$, the input of the model is $u_i(k) := \dot{m}_i C_p (T_i^{\mathrm{s}}(k) - T_i(k)) + (T^{\mathrm{o}} - T^{\mathrm{t}})/R_i^{\mathrm{o}} + q_i$, and the model becomes

$$x_i(k+1) = x_i(k) + \frac{\tau}{C_i} \sum_{j \in [Z] \setminus \{i\}} \frac{x_j(k) - x_i(k)}{R_{ji}} - \frac{\tau}{C_i} \frac{x_i(k)}{R_i^{\mathrm{o}}} + \frac{\tau}{C_i} u_i(k), \tag{6.6}$$

for all zones $i \in [n]$. This model is an n-dimensional discrete-time input-output linear system and can be written in the form

$$x(k+1) = Ax(k) + Bu(k), \tag{6.7}$$

with $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$.

6.1.3 Introduction of an arbitrary switching rule

We consider the building depicted in Figure 6.2.

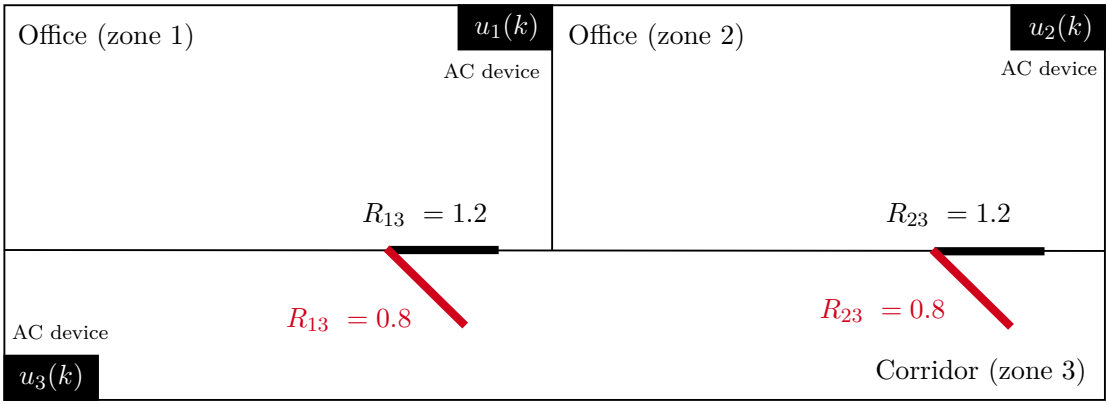


Figure 6.2: Building considered as an example. It has three zones: two offices and one corridor. The two offices have a door leading to the corridor. The fact that the door is open or closed influences the parameters $R_{13} = R_{31}$ and $R_{12} = R_{21}$.

If the door of zone (office) $i = 1, 2$ is open, then the thermal resistance between office i and the corridor R_{i3} will drop from 1.2 [KkW^{-1}] to 0.8 [KkW^{-1}]. We suppose that the

doors can be opened and closed arbitrarily. It yields that the system is now a switching system with an input. It has the form

$$x(k+1) = A_{\tilde{\sigma}(k)}x(k) + Bu(k), \tag{6.8}$$

where $A_{\tilde{\sigma}}$ can take four different values, respectively corresponding to the cases (closed, closed), (open, closed), (closed, open) and (open, open). In this order, they are noted $A_{(\mathrm{c,\ c})}$, $A_{(\mathrm{o,\ c})}$, $A_{(\mathrm{c,\ o})}$ and $A_{(\mathrm{o,\ o})}$. Thus,

$$A_{\tilde{\sigma}(k)} \in \tilde{\mathbf{\Sigma}} = \{A_1, A_2, A_3, A_3\} = \{A_{(\mathrm{c,\ c})}, A_{(\mathrm{o,\ c})}, A_{(\mathrm{c,\ o})}, A_{(\mathrm{o,\ o})}\}. \tag{6.9}$$

The switching is arbitrary, it means that each element $\tilde{\sigma}(k)$ in a switching sequence $(\tilde{\sigma}(0), \tilde{\sigma}(1), \dots)$ takes on arbitrary values in $[\tilde{m}]$, where $\tilde{m} = 4$.

In order to reach $T_i(k) = T^{\mathrm{t}}$ for all zones $i = 1, 2, 3$, we consider static linear feedback $u(k) = Kx(t)$ for $K \in \mathbb{R}^{n\times n}$ in the infinite horizon LQR problem. For more information about the design of such a controller, we refer the reader to Appendix A. System (6.8) becomes

$$x(k+1) = (A_{\tilde{\sigma}(k)} + BK)x(k). \tag{6.10}$$

We attract the attention of the reader on the fact that, **for now, the setting is entirely white-box and arbitrary**. Constraints and data-driven analysis come in the following sections.

6.1.4 Controller failures

The controller measures at each time step the differences between the real temperature and the target temperature for each zone, i.e. the variable $x_i(t)$ for $i = 1, 2, 3$. Suppose that the measures for the two offices can fail, and may measure the temperature as if it had already reached the right temperature, i.e. $x_i(t) = 0$. It is equivalent to consider that, in addition to the real matrix K designed by solving the infinite horizon LQR problem, the matrix K can takes on three other values: $K^{(0)}$ $K^{(1)}$, and $K^{(2)}$, respectively corresponding to the cases where no controller work, only the first controller works, and only the second controller works. The matrices $K^{(1)}$ and $K^{(2)}$ are the matrix K where respectively the second and the first column is set to zeros, and $K^{(0)}$ is K where the two first columns are set to zeros (see Appendix D for clarifications).

Following the definition of the considered system, we now consider sixteen different matrices. Indeed, for each four possible switches of system (6.10), there are four different ways for the controller to work. We define the new set of matrices as

$$\begin{aligned}\mathbf{\Sigma} = \{&A^{(0)}_{(\mathrm{c,\ c})}, A^{(0)}_{(\mathrm{o,\ c})}, A^{(0)}_{(\mathrm{c,\ o})}, A^{(0)}_{(\mathrm{o,\ o})},\\ &A^{(1)}_{(\mathrm{c,\ c})}, A^{(1)}_{(\mathrm{o,\ c})}, A^{(1)}_{(\mathrm{c,\ o})}, A^{(1)}_{(\mathrm{o,\ o})},\\ &A^{(2)}_{(\mathrm{c,\ c})}, A^{(2)}_{(\mathrm{o,\ c})}, A^{(2)}_{(\mathrm{c,\ o})}, A^{(2)}_{(\mathrm{o,\ o})},\\ &A_{(\mathrm{c,\ c})}, A_{(\mathrm{o,\ c})}, A_{(\mathrm{c,\ o})}, A_{(\mathrm{o,\ o})}\}\end{aligned} \tag{6.11}$$

And the dynamics of the new considered switching linear system is

$$x(k+1) = A_{\sigma(k)}x(k), \tag{6.12}$$

where $A_{\sigma(k)} \in \mathbf{\Sigma}$.

We now consider the following constraint. Suppose that each time that the door of an office is open, its AC controller fails, but with a time latency of τ, which is one time step in the discretized model. For example, suppose that both doors are closed for some k, both controller work fine. At time $k+1$, the first door opens, the first AC controller still works since it has a latency of one time step. At time $k+2$, the doors are opened arbitrarily, but it is guaranteed that the first controller does not work. We can model these constraints with the automaton $\mathbf{G}$ depicted in Figure 6.3.

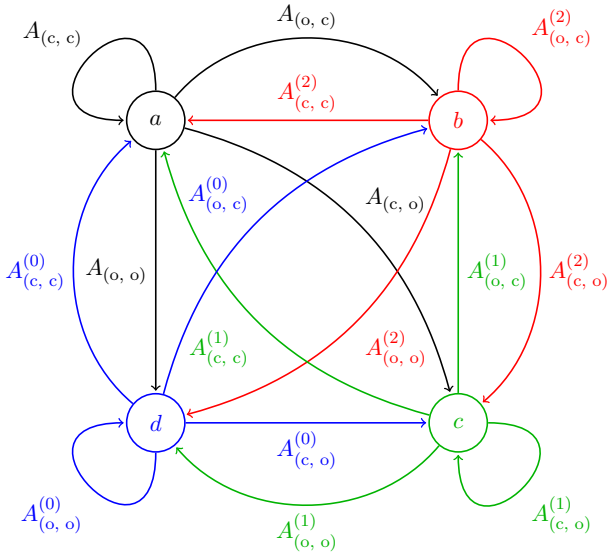
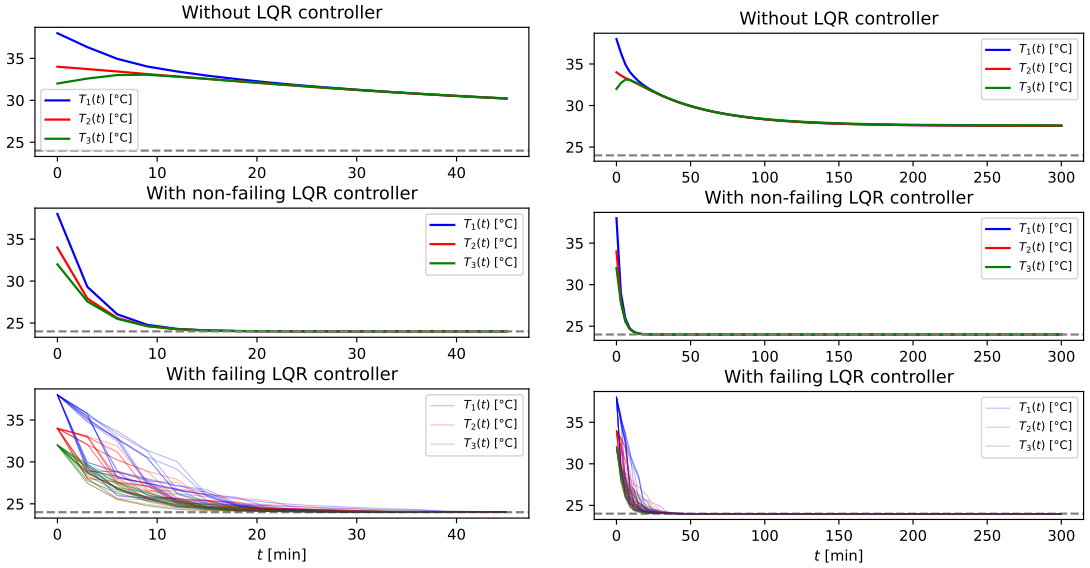


Figure 6.3: Automaton $\mathbf{G}$ modelling the constraints of considered multi-zone building system example. If one door is open, then it is guaranteed that, at the next time step, the corresponding controller does not work. For the sake of clarity, colors corresponding to the different controller failures have been assigned.

The considered CSLS is thus $S(\mathbf{G}, \mathbf{\Sigma})$. As an illustration of the impact of the controller in this case, we can find on Figure 6.4 an overview of the dynamics of the temperatures for each zone in three cases: with no controller, with a non-failing LQR controller (corresponding to the dynamics (6.10)), and with a failing LQR controller, corresponding to the considered CSLS $S(\mathbf{G}, \mathbf{\Sigma})$.



(a) Temperatures for $k \in \{1, \dots, 15\}$.

(b) Temperatures for $k \in \{1, \dots, 100\}$.

Figure 6.4: Temperatures for three different controllers. Note that, in the case of failing controller, we performed 20 simulations. In grey, the target temperature $T^{\mathrm{t}} = 24$ [C°]. Even in the constrained case, the temperatures seem to converge to the target temperature.

As we can observe on Figure 6.4, even with AC failures, the temperature seems to converge to the target temperature following a reasonable growth rate. Now we will answer the question: "**Is it possible for someone who does not have access to the model, to derive probabilistic guarantees on the growth rate of the true system?**".

6.2 Data-driven growth rate approximation

In order to analyze the stability of $S(\mathbf{G}, \mathbf{\Sigma})$ in a data-driven fashion, we will use the methods derived in Chapter 3 and Chapter 4. Indeed, the following propositions hold. First, if l-steps CQLF are used:

Proposition 6.2.1. *Suppose that a CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ admits a l-steps CQLF $\|\cdot\|_P$, $P \in \mathcal{S}^n$ such that, for a fixed $l > 0$, and for all $\mathbf{A} \in \mathbf{\Pi}_l$, there exists $\gamma > 0$ which satisfies*

$$\mathbf{A}^T P \mathbf{A} \preceq \gamma^{2l} P, \tag{6.13}$$

then, for any $x(0) \in \mathbb{R}^n$ and any $t \in \mathbb{N}$, there exists a constant $C > 0$ such that

$$\|x(lt)\|_2 \leq C\gamma^{lt}\|x(0)\|_2. \tag{6.14}$$

And, if MQLFs are used:

Proposition 6.2.2. *Suppose that a CSLS $S(\mathbf{G}(V,E),\mathbf{\Sigma})$ admits a set of MQLF $\{\|\cdot\|_{P_u}, u \in V\}$, $P \in \mathcal{S}^n$ such that, for a fixed $l > 0$, and for all $(u,v,\sigma) \in E$, there exists $\gamma > 0$ which satisfies*

$$A_\sigma^T P_v A_\sigma \preceq \gamma^2 P_u, \tag{6.15}$$

then, for any $x(0) \in \mathbb{R}^n$ and any $t \in \mathbb{N}$, there exists a constant $C > 0$ such that

$$\|x(t)\|_2 \leq C\gamma^t \|x(0)\|_2 \tag{6.16}$$

Proofs for Proposition 6.2.1 and Proposition 6.2.2 are given in Appendix E. Now let γ^*_{CQLF} and γ^*_{MQLF} be the probabilistic upper bounds on the CJSR obtained with respectively the l-steps CQLF and MQLF methods. It yields that, with a certain level of confidence, for all $x(0) \in \mathbb{R}^n$ and $t \in \mathbb{N}$, there exists a constant C such that

$$\|x(lt)\|_2 \leq C\left(\min\left\{\gamma^*_{\text{CQLF}}, \gamma^*_{\text{MQLF}}\right\}\right)^{lt} \|x(0)\|_2 \tag{6.17}$$

γ^*_{CQLF} and γ^*_{MQLF} thus give guarantees on the growth rate of $\|x(t)\|_2$.

Going back to the building system, suppose we have access to a dataset of $N = 50000$ observations with $l = 1$. We want to find a 95%-sure approximation of the CJSR $\rho(\mathbf{G},\mathbf{\Sigma})$ with unknwon $\mathbf{G}$ and $\mathbf{\Sigma}$. We thus compute the probabilistic upper bounds from the CQLF and the MQLF methods[1], and the deterministic lower bound from the MQLF method. Results for $N = 50000$ can be found in Table 6.3. An illustration of the evolution of the bounds for $N \in [1000, 50000]$ can be found in Figure 6.5.

	Deterministic lower bound	95%-sure upper bound
CQLF method	/	**0.8838**
MQLF method	0.4879	6.3491

Table 6.3: Bounds found by the CQLF and MQLF methods for the considered building systems with $N = 50000$ observations.

One can thus say with a confidence level of 95% that there exists a constant $C > 0$ such that the following holds:

$$\|x(t)\|_2 \leq C(0.8838)^t \|x(0)\|_2. \tag{6.18}$$

In this case, this result tells us how fast the temperature will converge in each zones to the target temperature with an unknown failing AC devide. More generally, **our methods have provided a probabilistic guarantee on the growth rate of the state of an unknown CSLS**.

[1]The "sub-confidence levels" are chosen such that they are equal. Thus $(100\% + \beta)/2 = 97,5\%$.

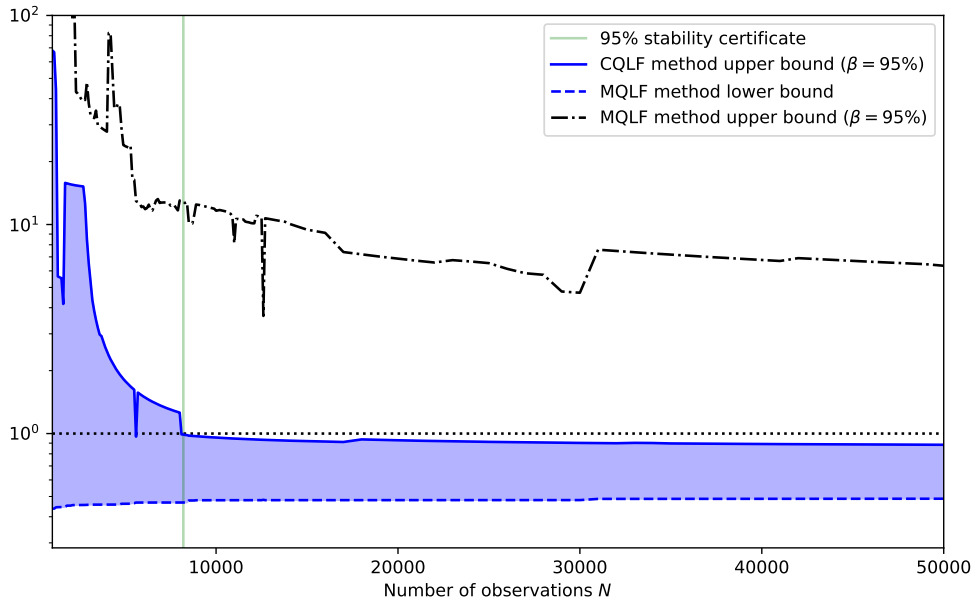


Figure 6.5: MQLF and CQLF upper bounds, as well as the MQLF lower bound for the multi-zone building system with constrained failing controller.

Conclusions

In this thesis, inspired by existing results [KBJT19, BJW21, RWJ21], **we took a step toward complexity and extended the scope of data-driven analysis for hybrid systems**. In particular, we provided two methods to derive probabilistic guarantees for the stability of data-driven constrained switching linear systems, a more general framework than arbitrary switching linear systems. We think that this constitutes an important advance of the state of the art for two reasons. First, in the context of the Cyber-Physical revolution, **hybrid systems prove to be central in more and more applications**. Second, as the complexity of such systems dramatically increases, **we believe in the importance of developing data-driven techniques for hybrid systems**. Moreover, we believe that this work can serve as a basis for a more global further research effort, for constrained switching linear systems or even more complex models.

Summary of the contributions

We started in Part I by introducing all the important concepts and tools to study stability of switching linear systems. More precisely, we presented model-based techniques for approximating the JSR and CJSR, in order to derive stability sufficient conditions. We also presented existing data-driven techniques for ASLSs.

Our main theoretical contributions state in Part II, where we build two data-driven stability techniques for CSLSs on top of existing results. In Chapter 3, we proposed a lifting result allowing us to reduce the computation of the CJSR of a given CSLS to the computation of a simpler JSR. We then build on top of [KBJT19, BJW21] the main theorem of Chapter 3, which provides **a first method based on common quadratic Lyapunov functions to derive probabilistic guarantees for the stability of CSLSs**. We claimed that in case of uniformity on the distribution of switching sequences, we can investigate further the obtained bound. In particular, we showed that a smaller entropy of the automaton allows for a better guarantee about the stability.

In Chapter 4, by considering that more information can be harvested, we leveraged the sensitivity analysis approach to propose a **method using another tool, namely multiple quadratic Lyapunov functions**. We used the CJSR to approximate the black-box stability of CSLSs, and generalized results of [RWJ21]. In particular, we

provided a first deterministic lower bound on the CJSR, as well as another probabilistic upper bound on it.

Comparaisons and examples are provided in Part III. In Chapter 5, **we compared the two derived methods**. We first discussed the limitations of each method compared to each other. In particular, we highlighted the limitations of the CQLF method compared to the MQLF method, depending on the nature of the considered CSLS. For numerical verification, **we implemented a `Julia` toolbox**. We then proceeded to a quantitative analysis of both methods. We verified the limitations of the CQLF method, and we analyzed further the influence of some parameters on both methods.

Finally, in Chapter 6, **we provided a concrete example**. We considered a **multi-zone building system**, whose zone temperatures are controlled by a model-based designed LQR controller. We then considered that the AC is subject to fails, with certain constraints. We then used our methods for CSLSs to answer the following question: suppose someone does not have access to the model but but only to a set of observations, can he assert with a certain confidence level that the system will converge with a certain rate to the target temperature? We concluded about the stability of the considered example.

Research perspectives

We think that our work can be followed by a further research effort, for CSLSs or even more complex hybrid systems.

First of all, we think that we can use existing tools to **make the methods derived in this thesis less conservative**. For example, our techniques are limited to quadratic Lyapunov functions. In this context, Sums-of-Squares techniques have proven to be useful for white-box stability analysis of CSLS [LJP16, LPJ19, PEDJ16], as well as for the data-driven stability analysis of ASLS [RWJ21]. As our MQLF method is based on [RWJ21], we think that it can be fairly easily be extended to Sums-of-Squares techniques. In the context of improving existing bounds, it is important to note that, for now, they rely on prior knowledge, which may not be available in a real data-driven application. In this purpose, it might be useful to tackle the topic of learning automaton information from observations (see e.g. [Ang87]). Finally, for the existing bounds, it can be useful to provide a computational complexity analysis. As the computations may take a lot of time, it can also be interesting to think about a real-time implementation, thus updating our bounds once new observations are available without having to re-compute an entire solution.

Also, we think that we can **explore other lifting ideas** as the one presented in Chapter 3, to derive other methods that may be better than the ones derived in this thesis. For example, for now, the MQLF method does not support l-steps horizon observations for $l > 1$. The concept of l-product lifting introduced in [PEDJ16] might be useful in this purpose. Other lifting to ASLSs have also been introduced in the model-based setting, and remain to be explored (see e.g. [XA20]).

In this thesis, we only focused on data-driven stability analysis of complex hybrid systems. The ideas exposed above are thus not an exhaustive list of possible improvement, as data-driven control of hybrid systems is a much wider field. Recently, **data-driven stabilization and controller design** (e.g. infinite-horizon LQR controller [KBM96]) have been proposed for ASLSs [WBJ21]. More complex hybrid systems such as CSLSs require more advanced controller design [DRI02, LD06, ELD14], and, to the best of our knowledge, data-driven controller design for such systems remains an open research field. Other types of controller such as MPC controller can also be explored (see [CB07] for an introduction).

Finally, another step towards complexity can be taken. ***Markovian Jump Linear Systems*** (*MJLS*s) are a popular and quite general model for complex dynamical systems with stochastic behaviour: they constitute one of the main models for stochastic hybrid systems (see the book [CMF05] or the article [FLF95] for a formal introduction and a historical overview). Apart from a few recent works [HS19, BTSB18] the field of MJLSs has not received much attention in the data-driven framework.

Bibliography

[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.

[Alu15] Rajeev Alur. *Principles of Cyber-Physical Systems*. MIT Press, 2015.

[Ang87] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

[AS98] Tsuyoshi Ando and Mau-hsiang Shih. Simultaneous contractibility. *SIAM Journal on Matrix Analysis and Applications*, 19(2):487–498, 1998.

[BJW21] Guillaume O. Berger, Raphaël M. Jungers, and Zheming Wang. Chance-constrained quasi-convex optimization with application to data-driven switched systems control. *arXiv:2101.01415 [cs, eess, math]*, 01 2021.

[BKSE12] Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing, 2012.

[BNT05] Vincent 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, 01 2005.

[Bra98] M.S. Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.

[BSH21] Filip Belić, Dražen Slišković, and Željko Hocenski. Detailed thermodynamic modeling of multi-zone buildings with resistive-capacitive method. *Energies*, 14(21), 2021.

[BT00a] Vincent Blondel and John Tsitsiklis. The boundedness of all products of a pair of matrices is undecidable. *Systems & Control Letters*, 41:135–140, 10 2000.

[BT00b] Vincent Blondel and John Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36:1249–1274, 09 2000.

[BTSB18] R. L. Beirigo, M. G. Todorov, and A. M. S. Barreto. Online td(a) for discrete-time markov jump linear systems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2229–2234, 2018.

[BWJ22a] Adrien Banse, Zheming Wang, and Raphaël M. Jungers. Black-box stability analysis of hybrid systems with sample-based multiple lyapunov functions, 2022.

[BWJ22b] Adrien Banse, Zheming Wang, and Raphaël M. Jungers. Learning stability guarantees for data-driven constrained switching linear systems, 2022.

[Cal10] G. Calafiore. Random convex programs. *SIAM Journal on Optimization*, 20:3427–3464, 09 2010.

[CB07] E. F. Camacho and C. Bordons. *Model Predictive control*. Springer London, 2007.

[CFM+19] Borui Cui, Cheng Fan, Jeffrey Munk, Ning Mao, Fu Xiao, Jin Dong, and Teja Kuruganti. A hybrid building thermal modeling approach for predicting temperatures in typical, detached, two-story houses. *Applied Energy*, 236:101–116, 2019.

[CG16] Marco Campi and Simone Garatti. Wait-and-judge scenario optimization. *Mathematical Programming*, 167, 07 2016.

[CG18] Marco C. Campi and Simone Garatti. *Introduction to the Scenario Approach*. MOS-SIAM Series on Optimization, 11 2018.

[CL08] Christos G. Cassandras and Stéphane Lafortune. *Introduction to Discrete Event Systems*. Springer US, 2008.

[CLS03] M.C. Campi, A. Lecchini, and S.M. Savaresi. An application of the virtual reference feedback tuning method to a benchmark problem. *European Journal of Control*, 9(1):66–76, 2003.

[CMF05] Oswaldo Luiz Valle Costa, Ricardo Paulino Marques, and Marcelo Dutra Fragoso. *Discrete-Time Markov Jump Linear Systems*. Springer London, 2005.

[Cro11] Martin Crowder. Nist handbook of mathematical functions edited by frank w. j. olver, daniel w. lozier, ronald f. boisvert, charles w. clark. *International Statistical Review*, 79(1):131–132, 2011.

[Dai11] Xiongping Dai. A gel'fand-type spectral radius formula and stability of linear constrained switching systems. *arXiv:1107.0124 [cs, math]*, 07 2011.

[DHvdWH11] M. C. F. Donkers, W. P. M. 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.

[DRI02] J. Daafouz, P. Riedinger, and C. Iung. Stability analysis and control synthesis for switched systems: a switched lyapunov function approach. *IEEE Transactions on Automatic Control*, 47(11):1883–1887, 2002.

[ELD14] Ray Essick, Ji-Woong Lee, and Geir E. Dullerud. Control of linear switched systems with receding horizon modal information. *IEEE Transactions on Automatic Control*, 59(9):2340–2352, 2014.

[FLC16] Pedro Fazenda, Pedro Lima, and Paulo Carreira. Context-based thermodynamic modeling of buildings spaces. *Energy and Buildings*, 124:164–177, 2016.

[FLF95] Y Fang, KA Loparo, and X Feng. Stability of discrete time jump linear systems. *Journal of Mathematical Systems, Estimation and Control*, 5(3):275–321, 1995.

[GST12] Rafal Goebel, Ricardo G. Sanfelice, and Andrew R. Teel. Hybrid dynamical systems. 03 2012.

[HGGL98] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin. Iterative feedback tuning: theory and applications. *IEEE Control Systems Magazine*, 18(4):26–41, 1998.

[HJT12] W.P.M.H. (Maurice) Heemels, Karl Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. pages 3270–3285, 12 2012.

[HS19] Bin Hu and Usman Ahmed Syed. Characterizing the exact behaviors of temporal difference learning algorithms using markov jump linear system theory, 2019.

[JHK16] Raphael M. Jungers, W. P. M. H. Heemels, and Atreyee Kundu. Observability and controllability analysis of linear systems subject to data losses, 2016.

[JP11] Raphaël M. Jungers and Vladimir Y. Protasov. Fast methods for computing the p-radius of matrices. *SIAM Journal on Scientific Computing*, 33(3):1246–1266, 2011.

[Jun09] Raphaël Jungers. *The Joint Spectral Radius: Theory and Applications*. Springer, 05 2009.

[KBJT19] Joris Kenanian, Ayca Balkan, Raphaël M. Jungers, and Paulo Tabuada. Data driven stability analysis of black-box switched linear systems. *Automatica*, 109, 2019.

[KBM96] Mayuresh V. Kothare, Venkataramanan Balakrishnan, and Manfred Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

[KK12] Kyoung-Dae Kim and Panganamala Ramana Kumar. Cyber–physical systems: A perspective at the centennial. *Proceedings of the IEEE*, 100:1287–1308, 2012.

[KK17] Alireza Karimi and Christoph Kammer. A data-driven approach to robust control of multivariable systems by convex optimization. *Automatica*, 85:227–233, Nov 2017.

[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, 03 2009.

[LD06] Ji-Woong Lee and Geir Dullerud. Uniform stabilization of discrete-time switched and markovian jump linear systems. *Automatica*, 42:205–218, 02 2006.

[Lee15] Edward A. Lee. The past, present and future of cyber-physical systems: A focus on models. *Sensors*, 15(3):4837–4869, 2015.

[Li11] S. Li. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics & Statistics*, 4:66–70, 2011.

[Lib03] Daniel Liberzon. *Switching in Systems and Control*. 2003.

[LJP16] Benoît Legat, Raphaël M. Jungers, and Pablo A. Parrilo. Generating unstable trajectories for switched systems via dual sum-of-squares techniques. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, HSCC '16, page 51–60, New York, NY, USA, 2016. Association for Computing Machinery.

[LM95] Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.

[LM99] D. Liberzon and A.S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19(5):59–70, 1999.

[LPJ19] Benoit Legat, Pablo A. Parrilo, and Raphael M. Jungers. An entropy-based bound for the computational complexity of a switched system. *IEEE Transactions on Automatic Control*, 64(11):4623–4628, nov 2019.

[LR06] B. Lincoln and A. Rantzer. Relaxing dynamic programming. *IEEE Transactions on Automatic Control*, 51(8):1249–1260, 2006.

[LS16] Edward Ashford Lee and Sanjit Arunkumar Seshia. *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. The MIT Press, 2nd edition, 2016.

[MB73] Kantilal L. Majumder and G. P. Bhattacharjee. Inverse of the incomplete beta function ratio. *Journal of The Royal Statistical Society Series C-applied Statistics*, 22:411–414, 1973.

[PEDJ16] Matthew Philippe, Ray Essick, Geir E. Dullerud, and Raphaël M. Jungers. Stability of discrete-time switching systems with constrained switching sequences. *Automatica*, 72:242–250, 2016.

[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*, 428:2385–2402, 2008.

[PJ14] Matthew Philippe and Raphaël M. Jungers. Converse lyapunov theorems for discrete-time linear switching systems with regular switching sequences, 2014.

[Ran06] Anders Rantzer. Relaxed dynamic programming in switching systems. *Control Theory and Applications, IEE Proceedings* -, 153:567– 574, 10 2006.

[RG60] Gian–Carlo Rota and W. Gilbert Strang. A note on the joint spectral radius. *Indagationes Mathematicae (Proceedings)*, 63:379–381, 1960.

[RWJ21] Anne Rubbens, Zheming Wang, and Raphaël M. Jungers. Data-driven stability analysis of switched linear systems with sum of squares guarantees, 2021.

[SG] Zhendong Sun and Shuzhi Sam Ge. *Stability Theory of Switched Dynamical Systems*. Communications and Control Engineering.

[Tab09] Paulo Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. 06 2009.

[TB97] John N. Tsitsiklis and Vincent D. Blondel. The lyapunov exponent and joint spectral radius of pairs of matrices are hard—when not impossible—to compute and to approximate. *Mathematics of Control, Signals and Systems*, 10:31–40, 1997.

[WBJ21] Zheming Wang, Guillaume O. Berger, and Raphaël M. Jungers. Data-driven control of switched linear systems with probabilistic stability guarantees, 2021.

[XA20] Xiangru Xu and Behcet Acikmese. Approximation of the constrained joint spectral radius via algebraic lifting, 2020.

[YMH98] Hui Ye, A.N. Michel, and Ling Hou. Stability theory for hybrid dynamical systems. *IEEE Transactions on Automatic Control*, 43(4):461–474, 1998.

Appendix A

Design of infinite-horizon LQR feedback for ASLSs

Consider a discrete-time input-output switching system of the form

$$x(t+1) = A_{\sigma(t)}x(t) + Bu(t), \tag{A.1}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^d$, $A_{\sigma(t)} \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times d}$. $A_{\sigma(t)} \in \mathbf{\Sigma} = \{A_1, \dots, A_m\}$, and the switching is arbitrary, it means that each element $\sigma(t)$ in a switching sequence $(\sigma(0), \sigma(1), \dots)$ takes on arbitrary values in $[m]$.

Let $\mathbf{x} = (x(0), x(1), \dots)$, $\mathbf{u} = (u(0), u(1), \dots)$ and $\boldsymbol{\sigma} = (\sigma(0), \sigma(1), \dots)$ be respectively one infinite-horizon state, input and switching sequences. For some *stage costs* $Q \succ 0$ and $P \succ 0$, consider the following quadratic cost:

$$J_\infty(\mathbf{x}, \mathbf{u}, \boldsymbol{\sigma}) := \sum_{t=0}^{\infty} x(t)^T Q x(t) + u(t)^T R u(t). \tag{A.2}$$

The inifinite-horizon LQR problem is the problem of finding

$$J^*(x(0)) = \inf_{\mathbf{u}} \sup_{\boldsymbol{\sigma}\in[m]^{\mathbb{N}}} J_\infty(\mathbf{x}, \mathbf{u}, \boldsymbol{\sigma}). \tag{A.3}$$

We restrict ourselves to a static feedback $u(t) = Kx(t)$ for some matrix $K \in \mathbb{R}^{d\times n}$. It can be shown (see e.g. [LR06] or [Ran06]), that if there are matrices (K, P) satisfying the following set of LMIs:

$$\forall A \in \mathbf{\Sigma}, \quad (A+BK)^T P (A+BK) \preceq P - Q - K^T R K, \tag{A.4}$$

then $J^*(x) \leq \|x(0)\|_P^2$ for all $x(0) \in \mathbb{R}^n$. An additional manipulation using the *Schur complement* allows to say that, letting $S = P^{-1}$ and $Y = KS$, (see [KBM96, Theorem 1]),

solving LMIs (A.4) is equivalent to solve the following optimization problem:

$$\min_{S,Y} -\log\det(S) \tag{A.5a}$$

$$\forall A \in \mathbf{\Sigma} : \begin{pmatrix} S & SA^T + Y^T B^T & S & Y^T \\ AS + BY & S & 0 & 0 \\ S & 0 & Q^{-1} & 0 \\ Y & 0 & 0 & R^{-1} \end{pmatrix} \preceq 0. \tag{A.5b}$$

Defintions of S and Y then allow to recover K and P as $P = S^{-1}$ and $K = YS^{-1} = YP$.

Appendix B

`Julia` code overview

A global overview of the source code can be found in Figure B.1. The entire source code and reproducible examples are located at `https://github.com/adrienbanse/DataDrivenCSLS.jl`.

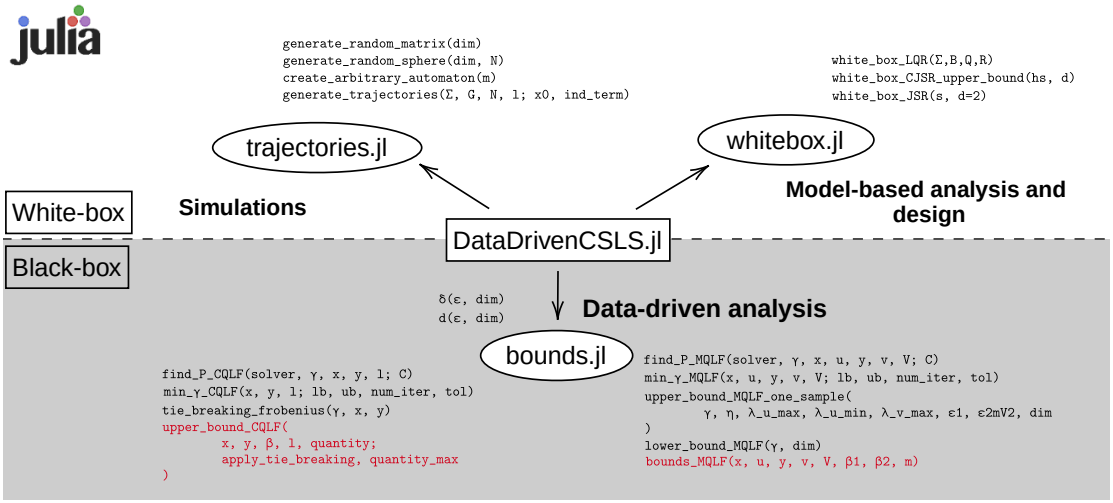


Figure B.1: Overview of the source code. There are three files around the main module `DataDrivenCSLS.jl`. Two of them, `trajectories.jl` and `whitebox.jl` allows to simulate and analyze in the white-box setting (to assess the black-box methods). The last one, `bounds.jl` contains the implementation of the CQLF and MQLF methods themselves.

We will now give a brief description for each implemented function. The `trajectories.jl` file allows to simulate CSLSs. It contains the following functions:

- `generate_random_matrix`: generates random matrices with unit spectral radius, i.e. generates A such that $\rho(A) = 1$;

- `generate_random_sphere`: generates N points uniformly distributed on the unit sphere $\mathbb{S}$;
- `create_arbitrary_automaton`: returns a flower of specified order m, i.e. an automaton with one node and m self-loops;
- `generates_trajectories`: generates N trajectories for a specified CSLS $S(\mathbf{G}, \mathbf{\Sigma})$ with specified initial state x_0; also the possibility to simulate a dynamical system $x_{t+1} = A_{\sigma_t} x_t + v$ where $v \in \mathbb{R}^n$ is a constant additional term, and where $A_{\sigma_t} \in \mathbf{\Sigma}$ and σ_t is constrained by $\mathbf{G}$.

The `whitebox.jl` file allows to design and analyze CSLSs in the white-box setting. It contains the following functions:

- `white_box_LQR`: design a white-box LQR controller for CSLSs as explained in Appendix A;
- `white_box_CJSR_upper_bound`: gives an upper bound on the CJSR of a white-box CSLS;
- `white_box_JSR`: gives an approximation on the JSR of a white-box ASLS.

The `bounds.jl` constitutes our main practical contribution. It provides tools to approximate the CJSR from a finite set of observations. First, some mathematical quantities are implemented:

- `δ`: spherical cap such as defined in Equation 2.7;
- `d`: quantity $d(\varepsilon) = \sqrt{2 - 2\delta(\varepsilon)}$ such as defined in Theorem 4.3.3.

In `bounds.jl`, the CQLF method such as described in Chapter 3 is implemented with the following functions:

- `find_P_CQLF`: optimization model that allows to find the matrix P such as defined in constraint (3.4b), such that LMIs (3.4c) hold for a specified value of γ;
- `min_γ_CQLF`: bissection procedure allowing to find the tightest γ such that there is a matrix P found by `find_P_CQLF`, it solves optimization problem (3.4) without the tie-breaking rule;
- `tie_breaking_frobenius`: implements the tie-breaking rule defined by multi-objective (3.4a), find the matrix P with minimal Frobenius norm such that LMIs (3.4c) hold for fixed optimal γ;
- `upper_bound_CQLF`: returns the CQLF upper bound such as defined in Theorem 3.2.2 from a finite set of observations.

Finally, the MQLF method is also implemented in `bounds.jl`. The following functions constitute the method:

- `find_P_MQLF`: optimization model that allows to find the set of matrices $\{P_u,\, u \in V\}$ such as defined in constraint (4.6b), such that LMIs (4.6c) hold for a specified value of γ;
- `min_γ_MQLF`: bissection procedure allowing to find the tightest γ such that there is a set of matrices $\{P_u,\, u \in V\}$ found by `find_P_MQLF`, it solves optimization problem (4.6);
- `upper_bound_MQLF_one_sample`: computes the MQLF upper bound (4.29) (see Corollary 4.4.1) with only one sample;
- `lower_bound_MQLF`: computes the MQLF lower bound such as defined in Proposition 4.2.1;
- `bounds_MQLF`: returns the MQLF lower and upper bounds such as respectively defined in Proposition 4.2.1 and Corollary 4.4.1 from a finite set of observations.

Appendix C

Examples involved in Chapter 5

First, note that for the sake of readibility, we do not present the set of matrices here. We thus invite the reader to check the source code in appendix to this document[1].

To study the dimensions, we defined four CSLSs $S(\mathbf{G}, \mathbf{\Sigma}_i)$ with $\mathbf{G}$ as depicted in Figure 3.2, and $\mathbf{\Sigma}_i \subset \mathbb{R}^{i \times i}$, with $i = 2, 3, 4, 5$.

To study the number of nodes, we defined four CSLSs $S(\mathbf{G}_i, \mathbf{\Sigma})$, where $\mathbf{G}_i(V_i, E_i)$, with $|V_i| = i$ for $i = 2, 3, 4, 5$ are as respectively depicted in Figure C.1, Figure C.2, Figure C.3 and Figure C.4.

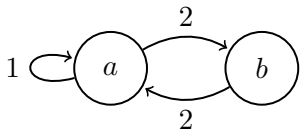


Figure C.1: Automaton $\mathbf{G}_2$ with 2 nodes for number of nodes analysis of Section 5.2.2.

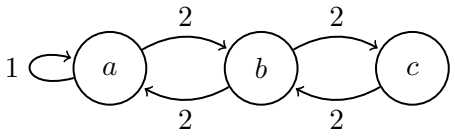


Figure C.2: Automaton $\mathbf{G}_3$ with 3 nodes for number of nodes analysis of Section 5.2.2.

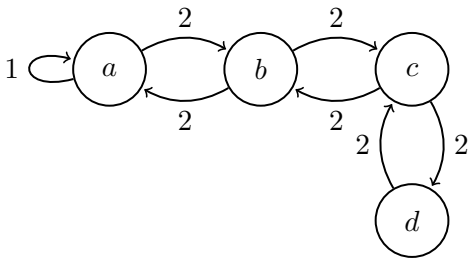


Figure C.3: Automaton $\mathbf{G}_4$ with 4 nodes for number of nodes analysis of Section 5.2.2.

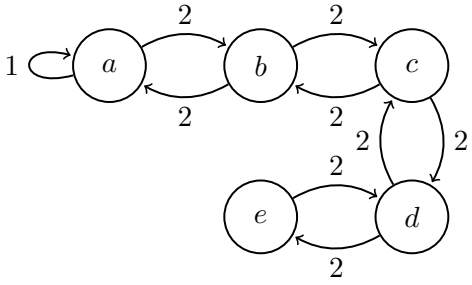


Figure C.4: Automaton $\mathbf{G}_5$ with 5 nodes for number of nodes analysis of Section 5.2.2.

[1] Also available at `https://github.com/adrienbanse/DataDrivenCSLS.jl`.

Appendix D

Equivalence between described failures and considered matrices in Section 6.1.4

Suppose without loss of generality that the controller receives measure $x_1(k) = 0$ for some k. Then

$$\begin{aligned} u(k) &= Kx(k) \\ &= \begin{pmatrix} K_{11}x_1(k) + K_{12}x_2(k) + K_{13}x_3(k) \\ K_{21}x_1(k) + K_{22}x_2(k) + K_{23}x_3(k) \\ K_{31}x_1(k) + K_{32}x_2(k) + K_{33}x_3(k) \end{pmatrix} \\ &= \begin{pmatrix} K_{12}x_2(k) + K_{13}x_3(k) \\ K_{22}x_2(k) + K_{23}x_3(k) \\ K_{32}x_2(k) + K_{33}x_3(k) \end{pmatrix}. \end{aligned} \tag{D.1}$$

Thus, using the same notations as in Equation (6.10), if one wants to model the fact that only the second controller works, if is equivalent to consider the switching linear system

$$x(k+1) = (A_\sigma + BK^{(2)})x(k), \tag{D.2}$$

where

$$K^{(2)} = \begin{pmatrix} 0 & K_{12} & K_{13} \\ 0 & K_{22} & K_{23} \\ 0 & K_{23} & K_{33} \end{pmatrix}. \tag{D.3}$$

A very similar analysis can be done for modelling $K^{(0)}$ and $K^{(1)}$. And one can consider that the matrix K switches in Equation (6.10) to model that the each controller may fail from time to time.

Appendix E

Proofs of Proposition 6.2.1 and Proposition 6.2.2

We first prove Proposition 6.2.1:

Proof. Suppose there are $\gamma > 0$, $P \in \mathcal{S}^n$ such that, for a fixed $l > 0$, and for all $\mathbf{A} \in \mathbf{\Pi}_l$, LMIs $\mathbf{A}^T P \mathbf{A} \preceq \gamma^{2l} P$ hold. Then for all $x \in \mathbb{R}^n$, $x\mathbf{A}^T P \mathbf{A} x \leq \gamma^{2l} x^T P x$.

Now, for all $x(0) \in \mathbb{R}^n \setminus \{0\}$, $x(lt) = \mathbf{A}_{t-1} \dots \mathbf{A}_0 x(0)$, with $\mathbf{A}_i \in \mathbf{\Pi}_l$ for $i = 0, \dots, t-1$. It yields $x(lt)^T P x(lt) \leq \gamma^{2lt} x(0)^T P x(0)$.

Now, since $P \in \mathcal{S}^n$, $\lambda_{\min}(P) I_n \preceq P \preceq \lambda_{\max}(P) I_n$, with $\lambda_{\min}(P)$ and $\lambda_{\max}(P)$ respectively the minimal and maximal eigenvalues of P. It follows that, for all $x \in \mathbb{R}^n$, $\lambda_{\min}(P)\|x\|_2^2 \leq x^T P x \leq \lambda_{\max}\|x\|_2^2$. And thus:

$$\lambda_{\min}(P)\|x(lt)\|_2^2 \leq x(lt)^T P x(lt) \leq \gamma^{2lt} x(0)^T P x(0) \leq \lambda_{\max}(P)\gamma^{2lt}\|x(0)\|_2^2, \qquad \text{(E.1)}$$

which yields $\|x(lt)\|_2 \leq C\gamma^{lt}\|x(0)\|_2$ with $C = \sqrt{\lambda_{\max}(P)/\lambda_{\min}(P)}$. □

The proof of Proposition 6.2.2 follows the same line as the proof of Proposition 6.2.1:

Proof. First, for any $x(0) \in \mathbb{R}^n \setminus \{0\}$, $x(t) = A_{\sigma(t-1)} \dots A_{\sigma(0)} x(0)$ for $(\sigma(t-1), \dots, \sigma(0))$ a word of length t accepted by $\mathbf{G}(V, E)$. Since LMIs $A_\sigma^T P_v A_\sigma \leq \gamma^2 P_u$ hold for all $(u, v, \sigma) \in E$, for any $t \in \mathbb{N}$, there are $u, w \in V$ such that $x(t)^T P_w x(t) \leq \gamma^{2t} x(0)^T P_u x(0)$, with u and w respectively the nodes at the beginning and the end of a length t path in $\mathbf{G}$(V, E). With a very similar reasoning as above, it allows us to conclude that $\|x(t)\|_2 \leq C\gamma^t\|x(0)\|_2$ with $C = \sqrt{\lambda_{\max}(P_u)/\lambda_{\min}(P_w)}$. □

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**