Counterexample-guided computation of polyhedral Lyapunov functions for hybrid systems ⋆

Guillaume O. Berger [a], Sriram Sankaranarayanan [a],

[a] *Department of Computer Science, University of Colorado, Boulder, USA*

Abstract

This paper presents a counterexample-guided iterative algorithm to compute convex, piecewise linear (polyhedral) Lyapunov functions for uncertain continuous-time linear hybrid systems. Polyhedral Lyapunov functions provide an alternative to commonly used polynomial Lyapunov functions. Our approach first characterizes intrinsic properties of a polyhedral Lyapunov function including its "eccentricity" and "robustness" to perturbations. We then derive an algorithm that either computes a polyhedral Lyapunov function proving that the system is stable, or concludes that no polyhedral Lyapunov function exists whose eccentricity and robustness parameters satisfy some user-provided limits. Significantly, our approach places no a priori bounds on the number of linear pieces that make up the desired polyhedral Lyapunov function.

The algorithm alternates between a learning step and a verification step, always maintaining a finite set of witness states. The learning step solves a linear program to compute a candidate Lyapunov function compatible with a finite set of witness states. In the verification step, our approach verifies whether the candidate Lyapunov function is a valid Lyapunov function for the system. If verification fails, we obtain a new witness. We prove a theoretical bound on the maximum number of iterations needed by our algorithm. We demonstrate the applicability of the algorithm on numerical examples.

Key words: Stability analysis, Lyapunov functions, Counterexample-guided methods, Linear optimization, Hybrid systems

1 Introduction

Lyapunov methods are a very useful tool for dynamical systems analysis (Khalil, 2002; Liberzon, 2003). The existence of a Lyapunov function for a given system allows to study important properties of the system, such as stability or positive invariants. However, the problem of finding such a function for a given system is quite challenging, especially for hybrid systems that switch between multiple modes with varying dynamics. In this paper, we consider the problem of automatically synthesizing Lyapunov functions to prove the stability of a class of uncertain piecewise linear systems. Such systems can switch between various modes wherein the dynamics for each mode are linear. Proving the stability of a piecewise linear system can be achieved using a *common Lyapunov function* which is shown to be decreasing according to the dynamics of all the modes of the system (Sun and Ge, 2011). However, finding a common Lyapunov function can be challenging, as well. Many methods search for *polynomial* Lyapunov functions using approaches such as *Sum-of-Squares* (SOS) programming (Lasserre, 2001). However, such methods also restrict the degree of the desired Lyapunov function to be within some a priori bounds. Failure to find such a function gives us little insight as to the nature of the underlying system.

In this paper, we focus on common Lyapunov functions that are convex and piecewise linear (also known as polyhedral) functions. Polyhedral functions are an interesting and relatively less studied class of Lyapunov functions when compared to polynomial Lyapunov functions. Their expressiveness can be modulated by adjusting the number of linear pieces (Blanchini and Miani, 2015). Furthermore, for a large class systems, including switched linear systems, there exist inverse results that show that if the system is stable, then a polyhedral Lyapunov function exists (Sun and Ge, 2011). The computation of polyhedral Lyapunov functions nevertheless remains difficult, even for linear systems. The main difficulties arise from the fact that the requirement for the function to decrease along the trajectories of the system

⋆ G. Berger is a BAEF fellow. S. Sankaranarayanan was supported by the US National Science Fundation (NSF).

Email addresses: `guillaume.berger@colorado.edu` (Guillaume O. Berger), `sriram.sankaranarayanan@colorado.edu` (Sriram Sankaranarayanan).

gives rise to nonlinear, nonconvex constraints (Blanchini and Miani, 2015). Furthermore, there is no known a priori bound on the number of pieces that such a function must have in order to be a Lyapunov function for a given class of systems (Ahmadi and Jungers, 2016).

We introduce a counterexample-guided approach to compute polyhedral Lyapunov functions for continuous-time, uncertain, piecewise linear dynamical systems. First, we formally characterize polyhedral Lyapunov functions in terms of key parameters that include (a) the *eccentricity* of the function which is defined analogously to the well-known concept of eccentricity of an ellipsoid; and (b) the *robustness* of the function which is defined in terms of limits on the perturbations of the dynamics of the underlying system that can be proven stable by the given polyhedral Lyapunov function. Naturally, we desire Lyapunov functions whose eccentricities are as small as possible and robustness parameters are as large as possible. In addition to the system being analyzed, our algorithm takes as inputs limits on the eccentricity and robustness parameters. It can yield two possible outcomes: (a) success along with a polyhedral Lyapunov function that proves the stability of the given system; or (b) failure to find a suitable function. However, unlike the related work we guarantee that if our algorithm fails then *no polyhedral Lyapunov function exists* for the system at hand, whose eccentricities and robustness parameters lie within the user-provided limits. By adjusting these limits, we can trade-off between the "complexity" of the Lyapunov functions we are searching for against computation time and the precision needed for the numerical computations involved. Significantly, our algorithm does not place any *a priori* bounds on the number of pieces of the desired Lyapunov function.

The algorithm itself is iterative, maintaining a finite set of states that we will call the *witness set*. Each iteration performs a learning step, followed by a verification step. In the learning step, a candidate polyhedral Lyapunov function is computed based on the witness set. Subsequently, the verification step checks whether the candidate Lyapunov function is a valid Lyapunov function for the system, and if not, outputs a new state wherein the Lyapunov conditions are violated. This state is added to the witness set in order to ensure that a previous candidate is never re-visited by our algorithm. The learning–verification procedure is repeated, until no counterexample is found in the verification step in which case our current candidate is the desired Lyapunov function or the learning step fails to identify a candidate which allows us to conclude that no Lyapunov function exists within user-provided eccentricity and robustness parameters.

A desirable property of our approach is that the learning and verification steps are implemented by solving a series of linear programming problems whose sizes are bounded by those of the underlying system and the number of witnesses found thus far. At the same time, we prove that the number of iterations of our approach is bounded and derive upper bounds in two different ways by establishing key properties of polyhedral functions whose eccentricities and robustness parameters are bounded.

The use of finitely many witness states allows our algorithm to avoid bilinear (nonconvex) constraints that often arise in *direct approaches* that try to enforce the Lyapunov conditions for an unknown polyhedral function over all (infinitely many) states.

We evaluate our approach on a series of numerical examples ranging from challenging instances that have been considered by other approaches as well as a family of piecewise linear systems known to be stable and with the number of state variables increasing from $d = 2$ to $d = 9$. We show that our approach terminates faster than the conservative upper bounds established by our theoretical analysis.

1.1 Related Work

Piecewise linear dynamical systems appear naturally in a wide range of applications, or as approximations of nonlinear systems (Pettit and Wellstead, 1995; Sun and Ge, 2011). Deciding stability of such systems is known to be extremely challenging even when restricted to simple dynamics (Prabhakar and Viswanathan, 2013). Ahmadi and Jungers show that for switched linear systems (even in 2 dimensions and with 2 modes), there is no a priori bound on the complexity of polynomial, polyhedral or piecewise quadratic functions sufficient to prove stability of the system (Ahmadi and Jungers, 2016). Moreover, the problem of deciding if there exists a polyhedral Lyapunov function with fixed number of facets is NP-hard (Berger and Sankaranarayanan, 2022). This motivates the benefit of computational methods, like the one described in this work, that do not restrict the complexity of the Lyapunov function up front.

Besides the many approaches for studying the stability of linear hybrid systems that consider quadratic or polynomial Lyapunov functions (Hassibi and Boyd, 1998; Sun and Ge, 2011; Jungers, 2009), polyhedral Lyapunov functions have also been well studied for this purpose. Available techniques for the computation of polyhedral Lyapunov functions can be divided into optimization-based and set-theoretic methods. Set-theoretic methods typically proceed by computing the image of some polyhedral set in the state space by the system and updating this set until it eventually provides a polyhedral Lyapunov function for the system (Miani and Savorgnan, 2005; Blanchini and Miani, 2015; Guglielmi et al., 2017). These methods are however usually restricted to discrete-time systems and often lack of formal complexity bounds. Optimization-based methods (Polański, 2000; Lazar and Doban, 2011; Ambrosino et al., 2012; Kousoulidis and Forni, 2021) aim to solve the nonlinear, nonconvex optimization problem accounting for the existence of a polyhedral Lyapunov function. This is achieved for instance by considering convex relaxations of the problem (Ambrosino et al., 2012), fixing some variables to make the problem tractable (Polański, 2000; Lazar and Doban, 2011), or using an alternating descent procedure to solve the optimization problem by repeatedly fixing one set of variables to fixed values while minimizing over the remaining variables (Kousoulidis and Forni, 2021). These methods are conservative as they lack formal guarantees of finding a global optimum for the optimization problem, so that they can fail to find a polyhedral Lyapunov function even if the system admits one. Our approach borrows from both of the above ones: namely, it uses an iterative process to recursively update the candidate Lyapunov function, and the update is done by solving a linear programming problem. Moreover, the approach allows to study continuous-time systems and comes with formal guarantees of convergence and complexity bounds on the running time of the procedure.

Our approach generates a finite set of states and uses them to infer a Lyapunov function candidate. This forms the basis of a "learning" approach to finding Lyapunov functions that has also been well studied. Topcu et al. use a randomly sampled set of states to derive polynomial Lyapunov functions with Sum-of-Squares programming to verify the result (Topcu et al., 2008). A key difference in this paper is that the generation of our witness set is based on counterexamples that refute previous candidates. Futhermore, we introduce a "gap" in our formulation of the learning versus the verification problems in order to yield formal termination guarantees. Kapinski et al. learn Lyapunov functions for nonlinear systems by combining learning of polynomial functions from finitely many samples (Kapinski et al., 2014). Recently, there have been many approaches that seek to infer neural networks as Lyapunov functions simultaneously with neural network controllers for dynamical systems (Chang et al., 2019; Dai et al., 2021). Likewise, Abate et al. present an approach that synthesizes neural-network-based Lyapunov functions for nonlinear systems (Abate et al., 2021). In all these approaches, a form of back-propagation is used to learn neural networks and a suitable solver is used to verify the result: dReal solver for nonlinear constraints in the case of Chang et al. (2019), Satisfiability-Modulo Theory (SMT) solver in the case of Abate et al. (2021) and a mixed integer solver in the case of Dai et al. (2021). Our approach does not synthesize controllers (although we would seek to do so in our future work). At the same time, neural networks with activation functions such as ReLU can be regarded as piecewise linear systems. Many of the approaches cited above use heuristic global optimization to generate counterexamples, whereas some approaches (notably, Chang et al., 2019; Dai et al., 2021; Abate et al., 2021) use expensive nonconvex solvers to find counterexamples. Nevertheless, there are no guarantees that this approach would find a Lyapunov function, even if one exists. In contrast, our approach offers a guarantee subject to constraints on the robustness and eccentricity of the desired function. Ravanbakhsh et al. derive polynomial Lyapunov functions by using a strategy similar to this paper: they generate polynomial Lyapunov functions for nonlinear systems by iterating between learning from a finite set of witnesses to using Sum-of-Squares optimization for verification (Ravanbakhsh and Sankaranarayanan, 2019). Unlike Ravanbakhsh et al., our work focuses on linear hybrid systems and polyhedral Lyapunov functions. This helps us avoid the use of Sum-of-Squares relaxations in favor of linear programming problems that lend themselves to precise and efficient solvers. As a result, our approach provides guarantees upon termination that are stronger than what could be obtained by a direct application of the methods discussed above to piecewise linear dynamics.

This work also extends from our recent work that provides a counterexample-guided method to compute polyhedral Lyapunov functions with fixed number of facets for linear hybrid dynamics (Berger and Sankaranarayanan, 2022). The most important difference of the present work is that our method does not place any a priori bound on the complexity of the learned Lyapunov function. Furthermore, our method presented here provides a guarantee that if it fails then no polyhedral Lyapunov function satisfying user-specified limits on robustness exists for the underlying system.

Outline. The paper is organized as follows. In Section 2, we introduce the problem of interest and remind important concepts related to Lyapunov analysis and polyhedral functions. In Section 3, we describe the algorithmic process to compute polyhedral Lyapunov functions. In Section 4, we provide a proof of termination and soundness of the algorithm. Finally, in Section 5, we demonstrate the applicability of the process on numerical examples.

Notation. $\|\cdot\|$ denotes a vector norm in $\mathbb{R}^d$ (typically the L_1 or L_∞ norm), and $\mathbb{S} : \{x \in \mathbb{R}^d : \|x\| = 1\}$ is the associated unit sphere. $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$, defined by $\|c\|_* = \max\{c^\top x : x \in \mathbb{S}\}$ (e.g., if $\|\cdot\|$ is the L_1 norm, then $\|\cdot\|_*$ is the L_∞ norm). By extension, $\|\cdot\|$ also denotes the matrix norm induced by $\|\cdot\|$ in $\mathbb{R}^{d\times d}$, defined by $\|A\| = \max\{\|Ax\| : x \in \mathbb{S}\}$.

2 Problem setting

We study continuous-time, uncertain, piecewise linear dynamical systems.

Definition 1. A *continuous-time, uncertain, piecewise linear dynamical system* is described by a finite set of modes Q, wherein each mode $q \in Q$ is associated with (i) a closed polyhedral conic region $H_q \subseteq \mathbb{R}^d$, and (ii) a transition matrix $A_q \in \mathbb{R}^{d\times d}$.

Let $\mathcal{F} : \mathbb{R}^d \rightrightarrows \mathbb{R}^d$ be the set-valued function defined by $\mathcal{F}(x) = \{A_q x : q \in Q,\ x \in H_q\}$. $\mathcal{F}$ completely describes the dynamics of the system. Unless otherwise stated, we will refer to the dynamical system as "System $\mathcal{F}$"; the sets Q, $(H_q)_{q\in Q}$ and matrices $(A_q)_{q\in Q}$ being implicit in the definition of $\mathcal{F}$.

A function $\xi : \mathbb{R}_{\geq 0} \to \mathbb{R}^d$ is called a *trajectory* of System $\mathcal{F}$ if it satisfies $\xi'(t) \in \mathcal{F}(\xi(t))$ for almost all $t \in \mathbb{R}_{\geq 0}$. We refer the reader to Goebel et al. (2012) for results concerning the existence and uniqueness of trajectories for piecewise, uncertain, linear dynamical systems.

2.1 Polyhedral Lyapunov stability analysis

We aim to study the stability of the origin for System $\mathcal{F}$, using Lyapunov analysis. First, let us recall the definition of the *Dini derivative*.

Definition 2 (Dini derivative)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ and $x, v \in \mathbb{R}^d$. The *upper-right Dini derivative* of f at x in the direction of v, denoted by $D^+f(x; v)$, is defined by $D^+f(x; v) = \limsup_{s\to 0^+} \frac{f(x+sv)-f(x)}{s}$.

Definition 3 (Lyapunov function)**.**

- A continuous function $V : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ is a *potential Lyapunov function* if (i) for all $x \in \mathbb{R}^d$, $V(x) = 0$ iff $x = 0$, and (ii) V is radially unbounded[1].
- A potential Lyapunov function $V : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ is a *Lyapunov function* for System $\mathcal{F}$ if for all $x \in \mathbb{R}^d \setminus \{0\}$ and $v \in \mathcal{F}(x)$, $D^+V(x; v) < 0$.

It is well known that if System $\mathcal{F}$ admits a Lyapunov function, then 0 is an asymptotically stable equilibrium point for System $\mathcal{F}$ (see, e.g., Blanchini and Miani, 2015, Theorem 2.19, or Khalil, 2002, Theorem 4.2). In this paper, we look for *polyhedral* Lyapunov functions, which are defined as the pointwise maximum of a finite set of linear functions.

Definition 4 (Polyhedral function)**.** A function $V : \mathbb{R}^d \to \mathbb{R}$ is a *polyhedral function* if there is a finite set $\mathcal{V} \subseteq \mathbb{R}^d$ s.t. for all $x \in \mathbb{R}^d$, $V(x) = \max_{c\in\mathcal{V}} c^\top x$.

As a convention, given a polyhedral function V, we let $\mathcal{V}$ be the set of coefficients of the various linear "pieces" defining V according to Definition 4. Note that $\mathcal{V}$ needs not be uniquely defined for a given function V. However, this will not pose an issue for the approach used in this paper. We also let $\mathcal{V}_{\max} : \max_{c\in\mathcal{V}} \|c\|_*$, and for all $x \in \mathbb{R}^d$, we let $\mathcal{V}(x) : \{c \in \mathcal{V} : V(x) = c^\top x\}$.

Proposition 5. *A polyhedral function V is a Lyapunov function for System $\mathcal{F}$ iff the following conditions hold:*

(C1) *For all $x \in \mathbb{R}^d \setminus \{0\}$, it holds that $V(x) > 0$.*
(C2) *For all $x \in \mathbb{R}^d \setminus \{0\}$, $v \in \mathcal{F}(x)$ and $c \in \mathcal{V}(x)$, it holds that $c^\top v < 0$.*

PROOF. The fact that (C1) is equivalent to V being a potential Lyapunov function is straightforward. Now, for a potential Lyapunov function V, the fact that (C2) is equivalent to V being a Lyapunov function for System $\mathcal{F}$ follows from the expression of the Dini derivative of V: for any $x \in \mathbb{R}^d$ and $v \in \mathbb{R}^d$, $D^+V(x; v) = \max\{c^\top v : c \in \mathcal{V}(x)\}$ (Blanchini and Miani, 2015, Eq. 2.30). □

2.2 Eccentricity and Robustness

We will now recast the conditions in Proposition 5 in a form that is tractable for optimization solvers and thus enables the overall approach that we will develop in this paper. To do so, we will describe parameters for a Lyapunov function that will be interpreted in terms of its *eccentricity* and *robustness* to perturbations.

Definition 6 (Robust Lyapunov conditions)**.** Let $\epsilon \geq 1$ be an *eccentricity* parameter, and $\theta > 0$ and $\delta > 0$ be *robustness* parameters. A function V is an *$(\epsilon, \theta, \delta)$-robust Lyapunov function* iff the following conditions hold:

(D1) For all $x \in \mathbb{R}^d$, $V(x) \geq \frac{1}{\epsilon}\mathcal{V}_{\max}\|x\|$.
(D2) For all $x \in \mathbb{R}^d$, $v \in \mathcal{F}(x)$ and $c \in \mathcal{V}$, $c^\top v \leq \frac{1}{\theta}(V(x) - c^\top x) - \delta\mathcal{V}_{\max}\|x\|$.

Theorem 7. *Let V be a polyhedral function with $\mathcal{V}_{\max} > 0$. V is a Lyapunov function for System $\mathcal{F}$ iff there exist $\epsilon \geq 1$, $\theta > 0$ and $\delta > 0$ s.t. V is an $(\epsilon, \theta, \delta)$-robust Lyapynov function.*

PROOF. See Appendix A. □

The parameter ϵ in Definition 6 measures the *eccentricity* of V, defined as $\frac{\max_{x\in\mathbb{R}^d:V(x)=1}\|x\|}{\min_{x\in\mathbb{R}^d:V(x)=1}\|x\|}$. We say that V has eccentricity ϵ if the latter ratio is smaller than or equal to ϵ. Naturally, the eccentricity ϵ for any function V satisfies $\epsilon \geq 1$.

Lemma 8. *Let $\epsilon \geq 1$. Then, ϵ satisfies (D1) in Definition 6 iff V has eccentricity ϵ.*

PROOF. Direct from (D1) in Definition 6 and the observation that $\min_{x\in\mathbb{R}^d:V(x)=1}\|x\| = 1/\mathcal{V}_{\max}$. □

The parameter θ in Definition 6 can be seen as a *time discretization* parameter, as (D2) can be rewritten as

[1] A function $V : \mathbb{R}^d \to \mathbb{R}$ is *radially unbounded* if $\lim_{r\to\infty} \inf\{V(x) : x \in X,\ \|x\| \geq r\} = \infty$.

$c^\top(x+\theta v) \leq V(x) - \theta\delta\mathcal{V}_{\max}\|x\|$. When combined with the eccentricity ϵ and "slack" δ, the largest time step that can be used in (D2) also gives a measure of the *robustness* of V as a Lyapunov function w.r.t. perturbations of System $\mathcal{F}$, as we will see below.

First, we establish the following result that holds in the absence of any perturbations. It notes that an (ϵ,θ,δ)-robust Lyapunov function V remains a robust Lyapunov function for $\epsilon' \geq \epsilon$, $\theta' \leq \theta$ and $\delta' \leq \delta$.

Lemma 9. *Let V be an (ϵ,θ,δ)-robust Lyapunov function. It holds that V is an $(\epsilon',\theta',\delta')$-robust for any $\epsilon' \in [\epsilon,\infty)$, $\theta' \in (0,\theta]$ and $\delta' \in (0,\delta]$.*

PROOF. Suppose V satisfies (D1) and (D2) in Definition 6 for parameters (ϵ,θ,δ). Since $\epsilon' \geq \epsilon$, we obtain from (D1) that for all $x \in \mathbb{R}^d$, $V(x) \geq \frac{1}{\epsilon}\mathcal{V}_{\max}\|x\| \geq \frac{1}{\epsilon'}\mathcal{V}_{\max}\|x\|$. Likewise, from (D2), we obtain that for all $x \in \mathbb{R}^d$, $v \in \mathcal{F}(x)$ and $c \in \mathcal{V}$, $c^\top v \leq \frac{1}{\theta}(V(x) - c^\top x) - \delta\mathcal{V}_{\max}\|x\| \leq \frac{1}{\theta'}(V(x) - c^\top x) - \delta\mathcal{V}_{\max}\|x\| \leq \frac{1}{\theta'}(V(x) - c^\top x) - \delta'\mathcal{V}_{\max}\|x\|$, where in the second inequality, we have used that $V(x) - c^\top x \geq 0$ and $\theta' \leq \theta$, and in the third inequality, we have used that $\delta' \leq \delta$. Therefore, V satisfies (D1)–(D2) in Definition 6 for $(\epsilon',\theta',\delta')$. $\square$

We now introduce the notion of *perturbation* of System $\mathcal{F}$, which will lead to the notion of *robustness* of a Lyapunov function w.r.t. perturbations of the system. For the sake of simplicity, we assume for this definition and Theorems 11 and 12 below that all regions $(H_q)_{q\in Q}$ satisfy $H_q = \mathbb{R}^d$. Let us define $a_{\max} : \max_{q\in Q}\|A_q\|$.

Definition 10 (Perturbed system)**.** A *γ-perturbation* of System $\mathcal{F}$ (with $H_q = \mathbb{R}^d$) is a continuous-time, uncertain, piecewise linear dynamical system (Definition 1) with set of modes $Q' = Q$, set of regions $(H'_q)_{q\in Q}$ satisfying $H'_q = \mathbb{R}^d$ for all $q \in Q$, and set of matrices $(A'_q)_{q\in Q}$ satisfying $\|A'_q - A_q\| \leq \gamma a_{\max}$ for all $q \in Q$.

The following theorems establish links between the robustness parameters (ϵ,θ,δ) of the polyhedral Lyapunov function and its robustness w.r.t. γ-perturbations of the system.

Theorem 11 (Sufficient condition for robustness)**.** *Let V be an (ϵ,θ,δ)-robust polyhedral Lyapunov function for system $\mathcal{F}$. Then, V is a Lyapunov function for any γ-perturbation of System $\mathcal{F}$ with $\gamma \in (0, \frac{\delta}{a_{\max}})$.*

PROOF. Let $\mathcal{F}'$ be a γ-perturbation of System $\mathcal{F}$. Fix $x \in \mathbb{S}$, $v' \in \mathcal{F}'(x)$ and $c \in \mathcal{V}(x)$. According to Definition 10, let $v \in \mathcal{F}(x)$ be s.t. $\|v - v'\| \leq \gamma a_{\max}$. It holds that $|c^\top v - c^\top v'| \leq \gamma a_{\max}\mathcal{V}_{\max}$. Hence, $c^\top v' \leq c^\top v + \gamma a_{\max}\mathcal{V}_{\max} < c^\top v + \delta\mathcal{V}_{\max}$, where the last inequality comes from the assumption on γ. By (D2), it follows that $c^\top v' < 0$, so that (C2) in Proposition 5 is satisfied for x and v'. Since x and v' were arbitrary, this concludes the proof. $\square$

Theorem 12 (Necessary condition for robustness)**.** *Let $\epsilon \geq 1$ and $\gamma > 0$. Let V be a polyhedral potential Lyapunov function with eccentricity ϵ. Assume that V is a Lyapunov function for any γ-perturbation of System $\mathcal{F}$. Then, (D2) in Definition 6 holds with any (θ,δ) satisfying $0 < \theta \leq \frac{\gamma}{2a_{\max}(\epsilon+\gamma)}$ and $0 < \delta \leq \frac{\gamma a_{\max}}{2\epsilon}$.*

PROOF. See Appendix B. $\square$

The relation between the parameters (ϵ,θ,δ) and the robustness of the Lyapunov function w.r.t. perturbations of the system being established, we focus in the following of the paper on finding (ϵ,θ,δ)-robust polyhedral Lyapunov functions for System $\mathcal{F}$.

Finally, we observe that the parameters ϵ, θ and δ are invariant w.r.t. positive scaling of V. That is, if V is an (ϵ,θ,δ)-robust polyhedral Lyapunov function, then so is the function $\frac{1}{\lambda}V$ for any $\lambda > 0$. Therefore, in the following, we restrict our attention to polyhedral functions with $\mathcal{V}_{\max} \leq 1$, i.e., with $\mathcal{V} \subseteq \mathbb{B}^* : \{c \in \mathbb{R}^d : \|c\|_* \leq 1\}$.

Example 13 (Running illustrative example)**.** Consider the continuous-time piecewise linear dynamical system described by $Q : \{1,2\}$, $H_1 : \mathbb{R}\times\mathbb{R}_{\geq 0}$, $H_2 : \mathbb{R}\times\mathbb{R}_{\leq 0}$,

$$A_1 : \begin{bmatrix} -0.2 & 1.0 \\ -1.0 & -0.2 \end{bmatrix} \quad \text{and} \quad A_2 : \begin{bmatrix} 0.01 & 1.0 \\ -1.0 & 0.01 \end{bmatrix}.$$

The vector field of the system is represented in Figure 1 (gray arrows). This system does not admit a polynomial Lyapunov function; see Lemma 14 below. Nevertheless, as we will see throughout the paper, we can compute a polyhedral Lyapunov function for the system, thereby proving that it is asymptotically stable. We can also evaluate the stability margin of the system, by computing parameters (ϵ,θ,δ) for which the system does not admit a robust polyhedral Lyapunov function.

Lemma 14. *System $\mathcal{F}$ described in Example 13 does not admit a polynomial Lyapunov function.*

PROOF. For a proof by contradiction, assume that V is a polynomial Lyapunov function for the system. Let $x_0 : [1,0]^\top$, and let $p : \mathbb{R} \to \mathbb{R}$ be the univariate polynomial defined by $p(r) = V(rx_0)$. Since V is radially unbounded, p is of nonzero even degree. Fix $r \in \mathbb{R}_{>0}$ and let $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^d$ be the trajectory with

$x(0) = rx_0$. It holds that $x(\pi) = e^{A_2\pi}rx_0 = -e^{0.01\pi}rx_0$. Hence, $V(x(0)) = p(r)$ and $V(x(\pi)) = p(-e^{0.01\pi}r)$. Since p is of nonzero even degree and $e^{0.01\pi} > 1$, there is $r > 0$ s.t. $p(-e^{0.01\pi}r) > p(r)$. Thus, with such a r, $V(x(\pi)) > V(x(0))$, contradicting the property that V decreases along the trajectories of the system (Khalil, 2002, Theorem 4.1), concluding the proof. $\square$

2.3 Overview of the Algorithm

We are given as inputs (i) the description of the system $\mathcal{F}$, and (ii) parameters $(\epsilon, \theta, \delta)$. This paper will present an algorithm to (a) compute a polyhedral Lyapunov function for System $\mathcal{F}$ if one exists; or (b) conclude that no polyhedral Lyapunov function V exists with eccentricity ϵ and robustness parameters (θ, δ).

As expected, we conclude that the system is stable if the procedure computes a polyhedral Lyapunov function. Furthermore, the parameters associated with this function helps determine how much the system can be perturbed while guaranteeing stability (as given by Theorem 12). However, if the procedure fails to find a Lyapunov function, we conclude that no Lyapunov function with eccentricity smaller than ϵ and robustness parameters larger than θ, δ exists. Thus, we conclude upon failure of our algorithm that the system itself is unstable or its stability requires a polyhedral Lyapunov function with larger eccentricity or smaller robustness than the input limits provided.

The iterative process is based on maintaining a finite set of states, called *witnesses*: $X : \{x_1, \ldots, x_N\} \subseteq \mathbb{R}^d$. The witness set is initialized to $X : \emptyset$. Each step iterates between two algorithms in succession:

- A *Learner*, which computes a polyhedral function satisfying the conditions of Theorem 7 over the finite set of witnesses, or concludes that no polyhedral Lyapunov function satisfying these conditions for all $x \in \mathbb{R}^d$ exists, that is, there is no polyhedral Lyapunov function for System $\mathcal{F}$ with eccentricity ϵ and robustness parameters (θ, δ).
- A *Verifier*, which, given a candidate polyhedral function (found by the learner), verifies whether the conditions of Proposition 5 are satisfied by all states. If the verification succeeds, we have our desired Lyapunov function. Otherwise, the verifier algorithm returns a point (called a *counterexample*) where the candidate fails to satisfy the Lyapunov conditions.

At the end of each iteration, we have three possible outcomes: (a) the learner refutes the existence of a "robust-enough" Lyapunov function, i.e., with eccentricity ϵ and robustness parameters (θ, δ); (b) the candidate polyhedral function verifies the Lyapunov conditions; or (c) a new witness point is added and the set X grows in cardinality by one element. We demonstrate that this process will eventually terminate, and provide upper bounds on the total number of iterations to termination.

The algorithm is described in Section 3 and the proof of its termination and soundness is presented in Section 4.

3 Description of the algorithm

First, we present the learner, then the verifier, and finally the overall algorithmic process. We assume input parameters $\epsilon \geq 1$, $\theta > 0$ and $\delta > 0$.

3.1 Learner: Computation of a candidate polyhedral Lyapunov function

Let $X : \{x_1, \ldots, x_N\} \subseteq \mathbb{R}^d$ be a finite set of witnesses. We consider the following computational problem, which aims to find a polyhedral function satisfying the conditions of Definition 6 at every witness point. The function V will have as many as $|X|$ pieces, one piece c_x associated with each witness $x \in X$.

Problem 15. Find a polyhedral function V with set $\mathcal{V} : \{c_x : x \in X\} \subseteq \mathbb{B}^*$ s.t. for all $x \in X$, (i) $c_x^\top x \geq \frac{1}{\epsilon}\|x\|$, and (ii) for all $v \in \mathcal{F}(x)$ and $c \in \mathcal{V}$, it holds that $c^\top v \leq \frac{1}{\theta}(c_x^\top x - c^\top x) - \delta\|x\|$.

The constraints (i)–(ii) in Problem 15 are linear in the decision variables $\{c_x : x \in X\}$ and the constraint $\mathcal{V} \subseteq \mathbb{B}^*$ is convex. Thus, Problem 15 can be cast as a convex optimization problem. These problems can be solved efficiently and accurately using for instance interior-point algorithms (Boyd and Vandenberghe, 2004).

We have the following co-soundness result:

Lemma 16. *If System $\mathcal{F}$ admits an $(\epsilon, \theta, \delta)$-robust polyhedral Lyapunov function, then Problem 15 is feasible.*

PROOF. Straightforward: it suffices to take, for each $x \in X$, $c_x \in \mathcal{V}(x)$, where V is an $(\epsilon, \theta, \delta)$-robust polyhedral Lyapunov function for System $\mathcal{F}$. $\square$

Conversely, any solution to Problem 15 is potentially a Lyapunov function for System $\mathcal{F}$. *However, it would need be verified for all state $x \in \mathbb{R}^d$, rather than just at the finite set of witnesses.*

Note that the constraints (i)–(ii) in Problem 15 are invariant w.r.t. positive scaling of the witnesses. Therefore, in the following, we assume w.l.o.g. that all witnesses are normalized to the unit sphere.

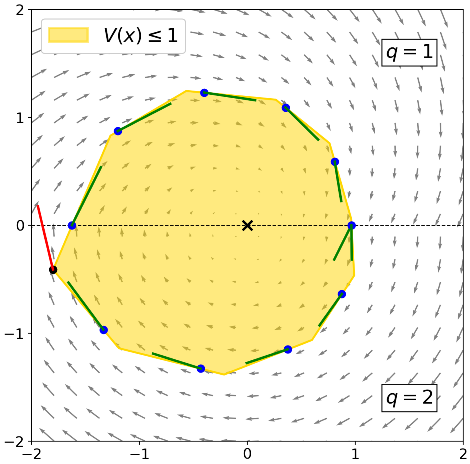


Fig. 1. (Running illustrative example). Candidate polyhedral function V provided by the learner (Problem 15), with parameters $\epsilon : 10$, $\theta : 1/4$ and $\delta : 0.05$, for System $\mathcal{F}$ (gray arrows) and witness set X (blue dots). A counterexample (black dot) provided by the verifier (Problem 19) for System $\mathcal{F}$ and the candidate polyhedral function V.

Assumption 17. $X \subseteq \mathbb{S}$.

Example 18 (Running illustrative example)**.** Consider System $\mathcal{F}$ described in Example 13. Let X be a set of 10 circularly placed points in $\mathbb{R}^2$. Let $\epsilon : 10$, $\theta : 1/4$ and $\delta : 0.05$. With these parameters, Problem 15 has a feasible solution V. The 1-sublevel set of V, the witness points $x \in X$ and the flow directions $v \in \mathcal{F}(x)$ are represented in Figure 1 (blue dots and green lines). We see that the flow directions always point toward to interior of the 1-sublevel set.

3.2 Verifier: Verification of the candidate polyhedral Lyapunov function and counterexample generation

Let V be a candidate polyhedral function output by the learner. We consider the problem of computing a *counterexample* x that fails to satisfy the conditions (C1)–(C2) in Proposition 5.

Problem 19. Find $x \in \mathbb{R}^d \setminus \{0\}$ s.t. either (i) $V(x) \leq 0$, or (ii) there is $v \in \mathcal{F}(x)$ and $c \in \mathcal{V}(x)$ s.t. $c^\top v \geq 0$.

Problem 19 can be approached by solving the following optimization problems: (i)

$$\begin{aligned} &\text{find} && x && \text{(1a)}\\ &\text{s.t.} && c^\top x \leq 0, \quad \forall c \in \mathcal{V}, && \text{(1b)}\\ & && \|x\| \geq 1; && \text{(1c)} \end{aligned}$$

and (ii) for each $c \in \mathcal{V}$ and $q \in Q$:

$$\begin{aligned} &\text{find} && x \in H_q && \text{(2a)}\\ &\text{s.t.} && c^\top x = 1 \geq b^\top x, \quad \forall b \in \mathcal{V}, && \text{(2b)}\\ & && c^\top A_q x \geq 0. && \text{(2c)} \end{aligned}$$

The constraint (1c) is nonconvex, but it can be enforced as the disjunction of $2d$ linear constraints (e.g., one for each component of x to be equal to -1 or 1), thereby allowing to solve (1) by solving $2d$ linear programs. Thus, solving problems (1) and (2) amounts to solve $2d+|\mathcal{V}||Q|$ linear programs with d variables and $|\mathcal{V}|+\max_{q\in Q} m_q+1$ constraints (where m_q is the number of linear constraints describing H_q), which can be achieved efficiently and reliably (Nesterov and Nemirovskii, 1994).

Theorem 20 (Soundness and completeness)**.** *Problem 19 is feasible iff* (1) *or* (2) *has a feasible solution.*

PROOF. Straightforward by the positive homogeneity of the problem. □

Example 21 (Running illustrative example)**.** Consider System $\mathcal{F}$ described in Example 13. Let V be the candidate polyhedral function computed by the learner in Example 18. We check whether there is a counterexample for V. It turns out to be the case since Problem 19 has a feasible solution $x \approx [-0.62, -0.85]^\top$. The counterexample point x and the flow direction $v \in \mathcal{F}(x)$ are represented in Figure 1 (black dot and red line). We see that the flow direction points towards the exterior of the 1-sublevel set of V.

3.3 Overall algorithmic process

We now describe the overall algorithmic process to compute a polyhedral Lyapunov function for System $\mathcal{F}$, or conclude that no polyhedral Lyapunov function with eccentricity ϵ and robustness parameters (θ, δ) exists.

The process starts with an empty set of witnesses (or alternatively a finite set of witnesses given as input of the algorithm). Then, the process enters a loop, in which at each iteration, the following learning step and verification step are performed sequentially: (a) From the current set of witnesses, the learner outputs a candidate polyhedral Lyapunov functions, or concludes that no $(\epsilon, \theta, \delta)$-robust polyhedral Lyapunov function exists. In the latter case, the algorithm stops and outputs FAIL. (b) The verifier checks whether the candidate polyhedral function provides a valid Lyapunov function for System $\mathcal{F}$. If it is the case, then the algorithm stops and outputs the candidate polyhedral function. Otherwise, it produces a counterexample, which is added to the witness set. The algorithm then proceeds with the next iteration of the loop. The process is described in Algorithm 1.

Example 22 (Running illustrative example)**.** Consider System $\mathcal{F}$ described in Example 13. Let $\epsilon : 10$, $\theta : 1/4$

Algorithm 1: Learning a Polyhedral Lyapunov Function.

Data: System $\mathcal{F}$, eccentricity $\epsilon \geq 1$, robustness parameters $\theta > 0$ and $\delta > 0$.
Result: Polyhedral Lyapunov function V, or FAIL.

```
/* Initialization                              */
X_0 ← ∅
/* Learning loop                               */
for k = 0, 1, ... do
    Compute candidate polyhedral function V_k by
      solving Problem 15 with X : X_k
    if Problem 15 has no solution then return FAIL
    Find a counterexample x_k for the candidate V_k
      by solving Problem 19 with V : V_k
    if Problem 19 has no solution then return V_k
    x̄_k ← x_k/‖x_k‖
    X_{k+1} ← X_k ∪ {x̄_k}
```

and $\delta : 0.05$. The construction of a polyhedral Lyapunov function for this system, using Algorithm 1, is illustrated in Figure 2. We start with a set X_0 of 4 points in $\mathbb{R}^2$. At each step k, a polyhedral function V_k, satisfying Problem 15 with $X : X_k$, is computed. Then, the verifier checks whether it can find a counterexample x_k satisfying Problem 19 for $V : V_k$. For several different steps k, we have represented, in Figure 2, the set X_k, the function V_k and the counterexample x_k.

After 27 steps, the process has computed a polyhedral Lyapunov function for System $\mathcal{F}$. The computed function is represented in the last plot of Figure 2. We have also represented a sample trajectory of the system. We observe that the trajectory does not leave the sublevel set, as predicted by the theory of Lyapunov.

Finally, if we set $\delta : 0.1$ (instead of 0.05) with the remaining parameters retaining their previous values ($\epsilon : 10$, $\theta : 1/4$), then after 20 steps, we conclude that Problem 15 has no solution. Thus, the algorithm outputs FAIL. This shows that System $\mathcal{F}$ does not admit a polyhedral Lyapunov function with eccentricity $\epsilon \leq 10$ and robustness parameters $\theta \geq 1/4$ and $\delta \geq 0.1$.

4 Proof of termination of the algorithm

The proof of termination exploits the gap between the constraints enforced by the learner and the constraints verified by the verifier. Indeed, the learner requires that the polyhedral function has eccentricity ϵ and robustness parameters (θ, δ) (over the set of witnesses), whereas the verifier only checks whether the function is a valid Lyapunov function. From this, we get that if V is a candidate provided by the learner with witness set X, and X' is "sufficiently close" to the witness set X, then V satisfies the conditions of the verifier on the set X'. Similarly, if V' is a small perturbation of V, then V' also satisfies the conditions of the verifier on the set X. This leads to two different arguments for termination with corresponding bounds on the number of iterations needed by our algorithm in the worst case. We discuss these properties in detail in Subsections 4.1 and 4.2, and derive associated complexity bounds for the algorithm.

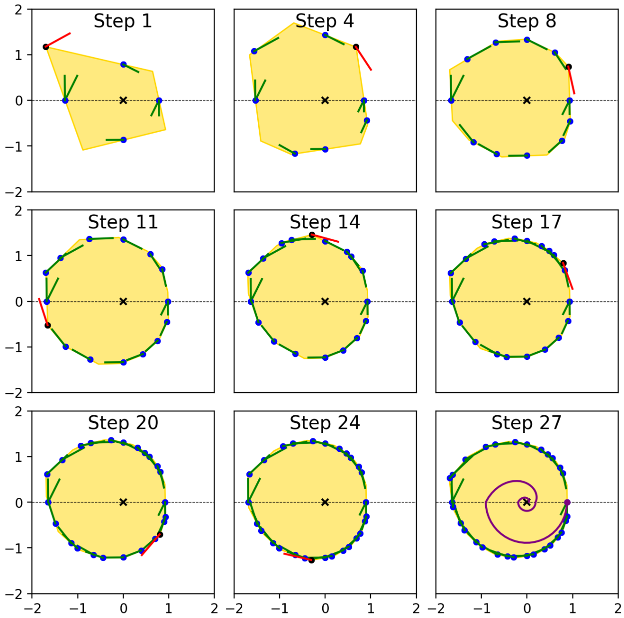


Fig. 2. (Running illustrative example). Different steps of the construction of a polyhedral Lyapunov function for System $\mathcal{F}$ in Example 22. *Yellow:* 1-sublevel set of V_k. *Blue dots:* witness points $x \in X_k$ scaled to be on the 1-level set of V_k. *Green lines:* flow directions of the system at the scaled witness points. *Black dot:* counterexample points $x \in X_k$ scaled to be on the 1-level set of V_k. *Red lines:* flow directions of the system at the scaled counterexample points. *Purple curve:* a sample trajectory of System $\mathcal{F}$.

4.1 Perturbation of the witness set

We discuss the property that the candidate function provided by the learner satisfies the conditions of the verifier at points that are close to the witness set. First, we introduce the notion of "inflation" of the witness set.

Definition 23. Let $X \subseteq X' \subseteq \mathbb{S}$ and $r \geq 0$. We say that X' is an *r-inflation* of X (w.r.t. the regions $(H_q)_{q \in Q}$) if for all $q \in Q$ and $x' \in X' \cap H_q$, there is $x \in X \cap H_q$ s.t. $\|x - x'\| < r$.

We now prove the property. Fix $\epsilon \geq 1$, $\theta > 0$ and $\delta > 0$. Define $\bar{r} : \min\{\frac{1}{\epsilon}, \frac{\theta\delta}{2+\theta a_{\max}}\}$.

Theorem 24. *Let $X \subseteq \mathbb{S}$. Let V be a solution to Problem 15. Let $X' \subseteq \mathbb{S}$ be an $\bar{r}$-inflation of X. Then, for all $x' \in X'$, (i) $V(x') > 0$, and (ii) for all $v' \in \mathcal{F}(x')$ and $c' \in \mathcal{V}(x')$, $c'^\top v' < 0$.*

PROOF. Let $q \in Q$ and $x' \in X' \cap H_q$. Let $x \in X \cap H_q$ be s.t. $\|x - x'\| < \bar{r}$. We first show that $V(x') > 0$. It holds that $c_x^\top x \geq \frac{1}{\epsilon}$ and $|c_x^\top x - c_x^\top x'| < \bar{r}$. Hence, by definition of $\bar{r}$, $c_x^\top x' > 0$, so that $V(x') > 0$.

Now, let $c \in \mathcal{V}(x')$. Let $v : A_q x$ and $v' : A_q x'$. Note that $\|v - v'\| = \|A_q(x - x')\| < a_{\max}\bar{r}$. We show that $c^\top v' < 0$. It holds that $c^\top x' \geq c_x^\top x' > c_x^\top x - \bar{r}$, $|c^\top x - c^\top x'| < \bar{r}$, and $|c^\top v - c^\top v'| < a_{\max}\bar{r}$. Also, $c^\top v \leq \frac{1}{\theta}(c_x^\top x - c^\top x) - \delta$. Hence, $c^\top v' < c^\top v - \frac{1}{\theta}(c_x^\top x - c^\top x) + \bar{r}(\frac{2}{\theta} + a_{\max}) \leq 0$, where the last inequality follows from the definition of $\bar{r}$. Since q, x, and c were arbitrary, this concludes the proof. □

Corollary 25. *Let $X_k \subseteq X_{k+1} \subseteq \mathbb{S}$ be two consecutive witness sets generated during the execution of Algorithm 1. Then, X_{k+1} is not an $\bar{r}$-inflation of X_k.*

PROOF. If X_{k+1} is an $\bar{r}$-inflation of X_k, then by Theorem 24, the counterexample $\bar{x}_k$ does not belong to X_{k+1}, contradicting the definition $X_{k+1} : X_k \cup \{\bar{x}_k\}$. □

From Corollary 25, we derive the following upper bound on the number of iterations of the algorithm. For given $s > 0$ and compact set $C \subseteq \mathbb{R}^d$, let $\mathrm{Pack}(s; C)$ denote the *s-packing number* of C, i.e, the largest cardinality of a subset $\hat{C} \subseteq C$ s.t. for all $x, y \in \hat{C}$, if $x \neq y$ then $\|x - y\| \geq s$.

Theorem 26 (Termination). *Algorithm 1 terminates in at most $|Q|\,\mathrm{Pack}(\bar{r}; \mathbb{S})$ steps.*

PROOF. Assume that Algorithm 1 produces at least $K+1$ counterexamples: $\bar{x}_0, \ldots, \bar{x}_K$. For each k, let $q_k \in Q$ be s.t. $\bar{x}_k \in H_{q_k}$ and $\min_{x \in X_k \cap H_{q_k}} \|x - \bar{x}_k\| \geq \bar{r}$ (by Corollary 25, such a q_k always exists). For each $q \in Q$, let $X_K {\downarrow} q : \{\bar{x}_k : q_k = q\}$. By the pigeonhole principle, there is $q \in Q$ s.t. $|X_K {\downarrow} q| \geq (K+1)/Q$. Fix such a q. It holds that for all $x, y \in X_K {\downarrow} q$, if $x \neq y$ then $\|x - y\| \geq \bar{r}$. Thus, $|X_K \downarrow q|$ is upper bounded by $\mathrm{Pack}(\bar{r}; \mathbb{S})$. This proves that $K + 1 \leq |Q|\,\mathrm{Pack}(\bar{r}; \mathbb{S})$. □

4.2 Perturbation of the candidate function

We discuss the property that a small enough perturbation of the candidate function provided by the learner, still satisfies the conditions of the verifier on the same witness set. We first formalize the property of “small enough” perturbation.

Definition 27. Let V_1 and V_2 be two polyhedral functions and $s \geq 0$. We say that V_1 and V_2 are *s-close* if for all $c_1 \in \mathcal{V}_1$, $\inf_{c_2 \in \mathcal{V}_2} \|c_1 - c_2\|_* < s$, and for all $c_2 \in \mathcal{V}_2$, $\inf_{c_1 \in \mathcal{V}_1} \|c_1 - c_2\|_* < s$,

We now prove the property. Fix $\epsilon \geq 1$, $\theta > 0$ and $\delta > 0$. Remember the definition $\bar{r} : \min\{\frac{1}{\epsilon}, \frac{\theta\delta}{2 + \theta a_{\max}}\}$.

Theorem 28. *Let $X \subseteq \mathbb{S}$. Let V be a solution to Problem 15. Let V' be a polyhedral function s.t. V and V' are $\bar{r}$-close. Then, for all $x \in X$, (i) $V'(x) > 0$, and (ii) for all $v \in \mathcal{F}(x)$ and $c' \in \mathcal{V}'(x)$, $c'^\top v < 0$.*

PROOF. Let $x \in X$. We first show that $V'(x) > 0$. Let $c \in \mathcal{V}(x)$, and let $c' \in \mathcal{V}'$ be s.t. $\|c - c'\|_* < \bar{r}$. It holds that $c^\top x \geq \frac{1}{\epsilon}$ and $|c^\top x - c'^\top x| < \bar{r}$. Hence, by definition of $\bar{r}$, $c'^\top x > 0$, so that $V'(x) > 0$.

Now, let $v \in \mathcal{F}(x)$ and $c' \in \mathcal{V}'(x)$. We show that $c'^\top v < 0$. Let $c \in \mathcal{V}$ be s.t. $\|c - c'\|_* < \bar{r}$, and $c'_x \in \mathcal{V}'$ be s.t. $\|c_x - c'_x\|_* < \bar{r}$. It holds that $c'^\top x \geq c'^\top_x x > c_x^\top x - \bar{r}$, $|c^\top x - c'^\top x| < \bar{r}$ and $|c^\top v - c'^\top v| < \bar{r} a_{\max}$. Also, $c^\top v \leq \frac{1}{\theta}(c_x^\top x - c^\top x) - \delta$. Hence, $c'^\top v < c^\top v - \frac{1}{\theta}(c_x^\top x - c^\top x) + \bar{r}(\frac{2}{\theta} + a_{\max}) \leq 0$, where the last inequality follows from the definition of $\bar{r}$. Since x, v and c' were arbitrary, this concludes the proof. □

Corollary 29. *Let $V_0, V_1, \ldots$ be the candidate polyhedral functions generated by the learner during the execution of Algorithm 1. Then, for any $k_1 < k_2$, V_{k_1} and V_{k_2} are not $\bar{r}$-close.*

PROOF. If V_{k_1} and V_{k_2} are $\bar{r}$-close, then by Theorem 28 the counterexample $\bar{x}_{k_1}$ is not in X_{k_2}, contradicting the definition $X_{k_2} : X_{k_1} \cup \{\bar{x}_{k_1}, \ldots, \bar{x}_{k_2 - 1}\}$. □

From Corollary 29, we derive the following upper bound on the complexity of the algorithm. Let us define the *covering number* $\mathrm{Cov}_*(s; C)$ for a compact set $C \subseteq \mathbb{R}^d$ as the smallest cardinality of a set $\hat{C} \subseteq \mathbb{R}^d$ s.t. for all $c \in \hat{C}$ there exists $c' \in \hat{C}$ s.t. $\|c - c'\| \leq s$.

Theorem 30 (Termination). *Algorithm 1 terminates in at most N steps, where N is the maximal cardinality of a set $\{V_k\}_{k=1}^N$ of polyhedral functions satisfying Problem 15 (for some witness sets $\{X_k\}_{k=1}^N$), that are not $\bar{r}$-close to each other. It holds that $N \leq 2^{\mathrm{Cov}_*(\frac{\bar{r}}{2}; \mathbb{B}^*)}$.*

PROOF. The bound N on the number of steps follows directly from Corollary 29. We derive the bound $N \leq 2^{\mathrm{Cov}_*(\frac{\bar{r}}{2}; \mathbb{B}^*)}$ as follows. Let $\{V_k\}_{k=1}^N$ be as in the statement of the theorem, and let $\hat{C} \subseteq \mathbb{R}^d$ be an $\frac{\bar{r}}{2}$-covering of $\mathbb{B}^*$. For each k, let $\hat{C}_k \subseteq \hat{C}$ be a minimal $\frac{\bar{r}}{2}$-covering of $\mathcal{V}_k$. We show that for any $k_1 \neq k_2$, $\hat{C}_{k_1} \neq \hat{C}_{k_2}$. For a proof by contradiction, let $k_1 \neq k_2$ be s.t. $\hat{C}_{k_1} = \hat{C}_{k_2}$. Let $c_1 \in \mathcal{V}_1$. There is $c \in \hat{C}_{k_1} = \hat{C}_{k_2}$ s.t. $\|c - c_1\|_* < \frac{\bar{r}}{2}$. Moreover, since $\hat{C}_{k_2}$ is minimal, there is $c_2 \in \mathcal{V}_2$ s.t. $\|c -$

$c_2\|_* < \frac{\bar{r}}{2}$. For such a c_2, it holds that $\|c_1 - c_2\|_* < \bar{r}$. Similarly, one can show that for all $c_2 \in \mathcal{V}_2$, there is $c_1 \in \mathcal{V}_1$ s.t. $\|c_1 - c_2\|_* < \bar{r}$. Thus, V_1 and V_2 are $\bar{r}$-close, a contradiction. Hence, we have shown that $k_1 \neq k_2$ implies $\hat{C}_{k_1} \neq C_{k_2}$. Since there are at most $2^{|\hat{C}|}$ subsets of $\hat{C}$, it follows that $K \leq 2^{|\hat{C}|}$. Since $\hat{C}$ was arbitrary, this concludes the proof. □

Thus, we have provided two proofs of terminaton for Algorithm 1 that yield two bounds on the running time of the algorithm.

Corollary 31 (Termination)**.** *Algorithm 1 terminates in at most* $\min\{|Q|\,\mathrm{Pack}(\bar{r};\mathbb{S}), N\}$ *steps, where N is defined in Theorem 30.*

Moreover, upon termination, the process outputs FAIL *only if System $\mathcal{F}$ does not admit an (ϵ,θ,δ)-robust polyhedral Lyapunov function; otherwise, it outputs a polyhedral Lyapunov function for System $\mathcal{F}$.*

PROOF. The bound on the number of steps follows from Theorems 26 and 30. Now, the fact that the algorithm outputs FAIL only if System $\mathcal{F}$ does not admit an (ϵ,θ,δ)-robust polyhedral Lyapunov function follows from Lemma 16. Otherwise, if the algorithm provides a polyhedral function, it means that this function has passed the verifier test, so that it is a valid Lyapunov function for System $\mathcal{F}$ (Theorem 20). □

Remark 32. For reasonable numbers of modes $|Q|$, the upper bound $2^{\mathrm{Cov}_*(\frac{\bar{r}}{2};\mathbb{B}^*)}$ on N is always worse than the bound $|Q|\,\mathrm{Pack}(\bar{r};\mathbb{S})$. However, it should be noted that N is in general much smaller than $2^{\mathrm{Cov}_*(\frac{\bar{r}}{2};\mathbb{B}^*)}$ because the set of polyhedral functions satisfying Problem 15 is much smaller than the set of all polyhedral functions; especially if the system is not robustly stable. Therefore, we have kept both bounds in Corollary 31: the first bound is more useful when the system is robustly stable, because in that case $\bar{r}$ is expected to be large, so that $\mathrm{Pack}(\bar{r};\mathbb{S})$ is smaller; the second bound is more useful when the system is not robustly stable, because in that case the set of possible candidate functions is expected to be smaller, so that N is smaller.

5 Numerical examples

All computations were made on a laptop with processor Intel Core i7-7600u and 16 GB RAM running Windows. We use Gurobi™, under academic license, as linear optimization solver.

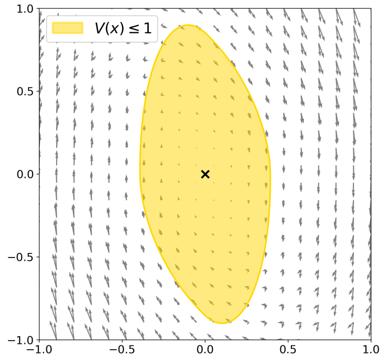


Fig. 3. Polyhedral Lyapunov function for the system described in Subsection 5.1 with $\alpha : 6$ (gray arrows). The function contains 300 linear pieces.

5.1 Benchmark: 2D uncertain linear system

This system was introduced in Zelentsovsky (1994) and was used by many authors as a benchmark comparing the performance of various Lyapunov synthesis approaches. The system is described by $\mathcal{F}(x) = \{A_p x : p \in \{0,\alpha\}\}$ where

$$A_p : \begin{bmatrix} 0 & 1 \\ -2 & -1 \end{bmatrix} + p \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}.$$

Zelentsovsky (1994) shows that the system with $\alpha : 3.82$ admits a quadratic Lyapunov function; Blanchini and Miani (1996) provide a polyhedral Lyapunov function for the system with $\alpha : 6$; Xie et al. (1997) provide a piecewise quadratic function for the system with $\alpha : 6.2$; Chesi et al. (2009) provide a polynomial Lyapunov function of degree 20 for the system with $\alpha : 6.8649$; and Ambrosino et al. (2012) provide a polyhedral Lyapunov function with 9694 vertices for the system with $\alpha : 6.87$.

Using Algorithm 1, we can compute a polyhedral Lyapunov function for the system with $\alpha : 6$. We use the parameters $\epsilon : 50$, $\theta : 1/64$ and $\delta : 0.001$. The computation takes about 5 minutes, and the resulting function contains 300 linear pieces (see Figure 3). We also show that the system with $\alpha : 6.87$ does not admit a polyhedral Lyapunov function with these parameters.

Remark 33. Let us mention that the algorithms in the above papers focus on uncertain linear systems, while our algorithm also tackles *piecewise* uncertain linear systems. Another difference, e.g., with Ambrosino et al. (2012), is that our algorithm does not involve any hyperparameters that must be set by the user; it just requires the inputs ϵ, θ and δ, that are used to verify whether the system admits a polyhedral Lyapunov function with specified eccentricity and robustness.

5.2 *Controlled mass–spring system*

We consider a mass–spring system whose dynamics is described by $\ddot{x} = -\frac{k}{m}x + \frac{1}{m}F$, where $m = 0.1\,\mathrm{kg}$ and $k = 2\,\mathrm{N/m}$. We control this system using a PID controller defined by $F(t) = -K_i y(t) - K_p x(t) - K_d \dot{x}(t)$, where $y(t) = \int_0^t x(\tau)\,\mathrm{d}\tau$, $K_i = 44\,\mathrm{N/m\cdot s}$, $K_p = 24\,\mathrm{N/m}$ and $K_d = 3.2\,\mathrm{N\cdot s/m}$. The force that can be applied on the system can only be *nonnegative*. To counterbalance the accumulation of the error when the input is negative, we add an *anti-windup* mechanism to the system. The block-diagram of the resulting system is depicted in Figure 4a. The dynamics of the system is described by the following piecewise linear system:

$$\text{if } K_d\dot{x} + K_p x + K_i y \le 0: \quad \begin{cases} \dot{y} = x, \\ m\ddot{x} + K_d\dot{x} + (k+K_p)x + K_i y = 0, \end{cases} \tag{3a}$$

$$\text{if } K_d\dot{x} + K_p x + K_i y \ge 0: \quad \begin{cases} \dot{y} = -y + x, \\ m\ddot{x} + kx = 0. \end{cases} \tag{3b}$$

Remark 34. Since (3b) is not stable, System (3) does not admit a Lyapunov function symmetric around the origin (including any polynomial Lyapunov function).

Using Algorithm 1, we compute a polyhedral Lyapunov function for this system; thereby showing that it is asymptotically stable. We use the parameters $\epsilon : 50$, $\theta : 1/32$ and $\delta : 0.001$. The computation takes 15 seconds and outputs a polyhedral Lyapunov function with 127 linear pieces (see Figure 4b).

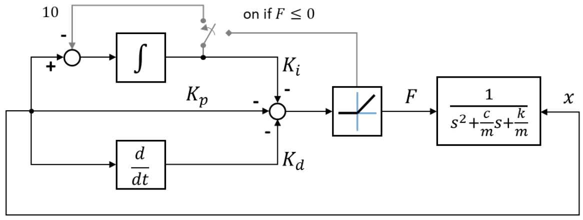
5.3 *Performance evaluation*

We want to evaluate the performance of the process, in terms of computation time and complexity of the resulting Lyapunov function, as a function of the dimension of the system and the stability margin of its matrices. Therefore, for $d \in \mathbb{N}_{>0}$, we let $U \in \mathbb{R}^{d\times d}$ be an orthogonal matrix, and for a parameter $\gamma \in \mathbb{R}$, we define the $d \times d$ matrix:

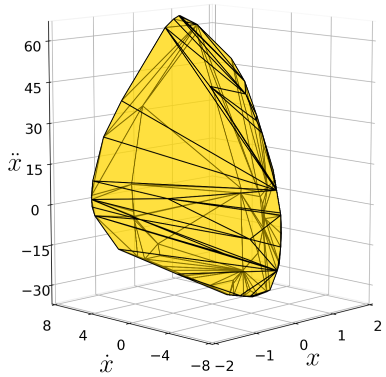
$$\Pi_\gamma : U(\mathbf{1}\mathbf{1}^\top - (d+\gamma)I)U^\top, \quad \mathbf{1} = [1,\ldots,1]^\top \in \mathbb{R}^d.$$

Consider System $\mathcal{F}$ with $Q : \{1,2\}$, $H_1 : \mathbb{R}_{\ge 0} \times \mathbb{R}^{d-1}$ and $H_2 : \mathbb{R}_{\le 0} \times \mathbb{R}^{d-1}$, $A_1 : \Pi_1$ and $A_2 : \Pi_\gamma$. For $\gamma < 0$, the system is unstable (thus, does not admit any Lyapunov function); see Lemma 37 in Appendix C. For $\gamma > 0$, the system is asymptotically stable and admits a $2d$-piece polyhedral Lyapunov function; see Lemma 38 in Appendix C.

For different values of $d \in \mathbb{N}_{>0}$ and $\gamma \in \mathbb{R}_{>0}$, we consider the parameters $\epsilon : 10$, $\theta : 1/8$ and $\delta \in \{\frac{\gamma}{50}, \gamma\}$. We use



(a) Block-diagram of the system system actuated by a ReLU-saturated PID controller. When the actuation force is saturated, an anti-windup mechanism (in gray) counterbalances the accumulation of the integrated error.



(b) 1-sublevel set of the Lyapunov function

Fig. 4. Example of Subsection 5.2

Algorithm 1 to compute a polyhedral Lyapunov function for System $\mathcal{F}$, or conclude that no polyhedral Lyapunov function with the given parameters exists. For each case, we measure the computation time and the number of linear pieces of the obtained Lyapunov function, or the number of iterations before the algorithm outputs FAIL. The results are gathered in Table 1. We observe that the computation time and the number of pieces/iterations increase when d increases and/or γ decreases. This is in accordance with the complexity analysis given in Corollary 31, which state that the maximal number of iterations depends on the $\bar{r}$-packing number in dimension d, wherein $\bar{r}$ depends, among others, on γ.

6 Conclusions

We presented an algorithmic framework to compute polyhedral Lyapunov functions for continuous-time piecewise linear systems. Compared to previous approaches in the literature, a key asset of our approach is that we do not put a priori bound on the number of facets of the Lyapunov function and we provide formal guarantees on the computational complexity of the algorithm. While these theoretical guarantees can be large when the dimension of the system increases, we demonstrate their practical applicability on numerical examples of dimensions from 2 to 9.

For further work, we will consider the problem of con-

	γ	$\delta = \frac{\gamma}{50}$ T [sec]	pieces	$\delta = \gamma$ T [sec]	iterations
$d = 4$	1	0.37	12	0.01	2
	0.1	0.84	21	0.23	13
	0.01	4.48	37	1.17	25
$d = 5$	1	0.47	14	0.01	2
	0.1	2.82	30	0.33	14
	0.01	11.54	53	2.47	31
$d = 6$	1	1.31	20	0.01	2
	0.1	11.67	46	1.07	22
	0.01	12.32	52	4.58	40
$d = 7$	1	2.22	23	0.01	2
	0.1	26.42	61	2.16	25
	0.01	34.09	68	11.94	41
$d = 8$	1	20.20	50	0.01	2
	0.1	47.27	87	5.65	35
	0.01	160.18	133	7.95	46
$d = 9$	1	13.85	45	0.01	2
	0.1	92.77	98	10.22	42
	0.01	206.85	149	24.61	54

Table 1
Evaluation of the performance of Algorithm 1 on System $\mathcal{F}$ described in Subsection 5.3. "T" refers to the total computation time of the algorithm. "Pieces" refers to the number of linear pieces of the computed polyhedral Lyapunov function, and "iterations" refers to the number of iterations needed by the algorithm to conclude that no polyhedral Lyapunov function exists.

troller synthesis for linear hybrid systems, by extending the approach for the learning of polyhedral *control* Lyapunov functions (CLFs) from counterexamples. This aim to address a lot of interesting control problems where the dynamics of a complex system (e.g., robot dynamics, power electronics) can be abstracted by linear hybrid systems (such as Neural Networks).

Acknowledgements

To be added in the final version.

References

Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, and Andrea Peruffo. Formal synthesis of Lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, 2021. doi: 10.1109/LCSYS.2020.3005328.

Amir Ali Ahmadi and Raphaël M Jungers. Lower bounds on complexity of Lyapunov functions for switched linear systems. *Nonlinear Analysis: Hybrid Systems*, 21:118–129, 2016. doi: 10.1016/j.nahs.2016.01.003.

Roberto Ambrosino, Marco Ariola, and Francesco Amato. A convex condition for robust stability analysis via polyhedral Lyapunov functions. *SIAM Journal on Control and Optimization*, 50(1):490–506, 2012. doi: 10.1137/110796479.

Guillaume O Berger and Sriram Sankaranarayanan. Learning fixed-complexity polyhedral Lyapunov functions from counterexamples, 2022. arXiv preprint arXiv: 2204.06693 (v1: 14$^{\text{th}}$ April 2022).

Franco Blanchini and Stefano Miani. On the transient estimate for linear systems with time-varying uncertain parameters. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 43 (7):592–596, 1996. doi: 10.1109/81.508180.

Franco Blanchini and Stefano Miani. *Set-theoretic methods in control*. Birkhäuser, Cham, 2$^{\text{nd}}$ edition, 2015. doi: 10.1007/978-3-319-17933-9.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, UK, 2004. doi: 10.1017/CBO9780511804441.

Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural Lyapunov control. In *NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 3245–3254. ACM, 2019. doi: 10.5555/3454287.3454579.

Graziano Chesi, Andrea Garulli, Alberto Tesi, and Antonio Vicino. *Homogeneous polynomial forms for robustness analysis of uncertain systems*, volume 390 of *Lecture Notes in Control and Information Sciences*. Springer, London, 2009. doi: 10.1007/978-1-84882-781-3.

Hongkai Dai, Benoit Landry, Lujie Yang, Marco Pavone, and Russ Tedrake. Lyapunov-stable neural-network control. In *Proceedings of Robotics: Science and Systems*, Virtual, 2021. doi: 10.15607/RSS.2021.XVII.063.

Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. *Hybrid dynamical systems: modeling stability, and robustness*. Princeton University Press, Princeton, NJ, 2012.

Nicola Guglielmi, Linda Laglia, and Vladimir Protasov. Polytope Lyapunov functions for stable and for stabilizable LSS. *Foundations of Computational Mathematics*, 17(2):567–623, 2017. doi: 10.1007/s10208-015-9301-9.

Arash Hassibi and Stephen Boyd. Quadratic stabilization and control of piecewise-linear systems. In *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, pages 3659–3664. IEEE, 1998. doi: 10.1109/ACC.1998.703296.

Raphaël M Jungers. *The joint spectral radius: theory and applications*. Springer, Berlin, 2009. doi: 10.1007/978-3-540-95980-9.

James Kapinski, Jyotirmoy V Deshmukh, Sriram Sankaranarayanan, and Niko Aréchiga. Simulation-guided Lyapunov analysis for hybrid dynamical systems. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 133–142. ACM, 2014. doi: 10.1145/2562059.2562139.

Hassan K Khalil. *Nonlinear systems*. Prentice-Hall, Upper Saddle River, NJ, 3rd edition, 2002.

Dimitris Kousoulidis and Fulvio Forni. Polyhedral Lyapunov functions with fixed complexity. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 3293–3298. IEEE, 2021. doi: 10.1109/CDC45484.2021.9683697.

Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001. doi: 10.1137/S1052623400366802.

Mircea Lazar and Alina I Doban. On infinity norms as Lyapunov functions for continuous-time dynamical systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7567–7572. IEEE, 2011. doi: 10.1109/CDC.2011.6161163.

Daniel Liberzon. *Switching in systems and control*. Birkhäuser, Boston, MA, 2003. doi: 10.1007/978-1-4612-0017-8.

Stefano Miani and Carlo Savorgnan. MAXIS-G: a software package for computing polyhedral invariant sets for constrained LPV systems. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 7609–7614. IEEE, 2005. doi: 10.1109/CDC.2005.1583390.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, Philadelphia, PA, 1994. doi: 10.1137/1.9781611970791.

Njal BOL Pettit and Peter E Wellstead. Analyzing piecewise linear dynamical systems. *IEEE Control Systems Magazine*, 15(5):43–50, 1995. doi: 10.1109/37.466263.

Andrzej Polański. On absolute stability analysis by polyhedral Lyapunov functions. *Automatica*, 36(4):573–578, 2000. doi: 10.1016/S0005-1098(99)00180-6.

Pavithra Prabhakar and Mahesh Viswanathan. On the decidability of stability of hybrid systems. In *HSCC'13: Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 53–62. ACM, 2013. doi: 10.1145/2461328.2461339.

Hadi Ravanbakhsh and Sriram Sankaranarayanan. Learning control Lyapunov functions from counterexamples and demonstrations. *Autonomous Robots*, 43 (2):275–307, 2019. doi: 10.1007/s10514-018-9791-9.

Zhendong Sun and Shuzhi Sam Ge. *Stability theory of switched dynamical systems*. Springer, London, 2011. doi: 10.1007/978-0-85729-256-8.

Ufuk Topcu, Andrew Packard, and Peter Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44(10):2669–2675, 2008. doi: 10.1016/j.automatica.2008.03.010.

Lin Xie, Serge Shishkin, and Minyue Fu. Piecewise Lyapunov functions for robust stability of linear time-varying systems. *Systems & Control Letters*, 31(3): 165–171, 1997. doi: 10.1016/S0167-6911(97)00027-3.

AL Zelentsovsky. Nonquadratic Lyapunov functions for robust stability analysis of linear uncertain systems. *IEEE Transactions on Automatic Control*, 39(1):135–138, 1994. doi: 10.1109/9.273350.

A Proof of Theorem 7

Proof of "if" direction. It is straightforward to see that (D1)–(D2) implies (C1)–(C2) in Proposition 5. Therefore, by Proposition 5, (D1)–(D2) implies that V is a Lyapunov function System $\mathcal{F}$.

Proof of "only if" direction. Assume that V is a Lyapunov function for System $\mathcal{F}$. Then, by (C1) in Proposition 5 and V being continuous, there is $\epsilon \geq 1$ s.t. for all $x \in \mathbb{S}$, $V(x) \geq \frac{\mathcal{V}_{\max}}{\epsilon}$. Now, since V is positively homogeneous of degree 1, it follows that for all $x \in \mathbb{R}^d$, $V(x) \geq \frac{\mathcal{V}_{\max}}{\epsilon}\|x\|$, so that (D1) holds.

It remains to find θ and δ s.t. (D2) holds. Therefore, fix $x \in \mathbb{S}$ and $c \in \mathcal{V}$. First, assume that $c \in \mathcal{V}(x)$. Then, by (C2), there is $\delta_{x,c} > 0$ s.t. for all $v \in \mathcal{V}$, $c^\top v < -\delta_{x,c}\mathcal{V}_{\max}$. Now, assumme that $c \notin \mathcal{V}(x)$, i.e., $c^\top x < V(x)$. Thus, there is $\delta_{x,c} > 0$ and $\theta_{x,c} > 0$ s.t. for all $v \in \mathcal{V}$, $c^\top(x + \theta_{x,c}v) < V(x) - \delta_{x,c}\mathcal{V}_{\max}$. Since $\mathcal{F}$ and V are continuous, there is a neighborhood $\mathcal{N}_x \subseteq \mathbb{S}$ of x s.t. for all $x' \in \mathcal{N}_x$ and all $v \in \mathcal{F}(x')$, $c^\top(x' + \theta_{x,c}v) < V(x') - \delta_{x,c}\mathcal{V}_{\max}$.

Since x was arbitrary, the above holds for every $x \in \mathbb{S}$. By compactness of $\mathbb{S}$, there exists a finite set of points $x_1, \ldots, x_n \in \mathbb{S}$ s.t. $\bigcup_{i=1}^n \mathcal{N}_{x_i}$ covers $\mathbb{S}$. We can then define $\theta : \min_{i:1,\ldots,n} \theta_{x_i,c} > 0$ and $\delta : \min_{i:1,\ldots,n} \theta_{x_i,c} > 0$. (D2) is then satisfied for c and for all $x \in \mathbb{S}$. By the positive homogeneity of V, it then follows that (D2) is satisfied for c and for all $x \in \mathbb{R}^d$. Now, since there is a finite set of values for c, we can find θ and δ so that (D2) holds, concluding the proof. □

B Proof of Theorem 12

First, note that by (D1) in Definition 6, it holds that for all $x \in \mathbb{R}^d \setminus \{0\}$ and $c \in \mathcal{V}(x)$, $\|c\|_* \geq \frac{\mathcal{V}_{\max}}{\epsilon}$. W.l.o.g. assume that $\mathcal{V}_{\max} = 1$. Denote $\eta : \frac{\gamma a_{\max}}{\epsilon}$. We will need the following result.

Lemma 35. *For all $x \in \mathbb{R}^d$, $v \in \mathcal{F}(x)$ and $c \in \mathcal{V}(x)$, $c^\top v \leq -\eta\|x\|$.*

PROOF. Let $x \in \mathbb{R}^d$, $v \in \mathcal{F}(x)$ and $c \in \mathcal{V}(x)$ Since V is a Lyapunov function for any γ-perturbation of System $\mathcal{F}$, it holds that for all $u \in \mathbb{R}^d$ s.t. $\|u\| \leq \gamma a_{\max}\|x\|$, $c^\top(v + u) \leq 0$. This implies that $c^\top v \leq -\gamma a_{\max}\|c\|_*\|x\| \leq -\frac{\gamma a_{\max}}{\epsilon}\|x\| = -\eta\|x\|$. □

We proceed with the proof of the theorem. Therefore, fix $x \in \mathbb{R}^d$, $q \in Q$ and $c \in \mathcal{V}$. Let $v : A_q x$. We will show that $c^\top v \leq \frac{1}{\theta}(V(x) - c^\top x) - \frac{\eta}{2}\|x\|$, where $\theta : \frac{\eta}{2a_{\max}^2 + 2\eta a_{\max}}$.

First, if $c \in \mathcal{V}(x)$, then we are done by Lemma 35. Thus, assume that $V(x) > c^\top x$.

Let $s : \frac{2}{\eta\|x\|}(V(x) - c^\top x) > 0$ and $y : x + sv$. Let $c_y \in \mathcal{V}(y)$. We will need the following result.

Lemma 36. $c^\top v \leq c_y^\top v + \frac{\eta}{2}\|x\|$.

PROOF. $sc_y^\top v \geq c_y^\top x - V(x) + sc_y^\top v = c_y^\top y - V(x) \geq c^\top y - V(x) = c^\top x + sc^\top v - V(x) = s(c^\top v - \frac{\eta}{2}\|x\|)$. □

Note that $\|v\| \leq a_{\max}\|x\|$, so that $\|y - x\| \leq sa_{\max}\|x\|$. Let $w : A_q y$. By Lemma 35, it holds that $c_y w \leq -\eta\|y\|$. Thus, $c_y^\top w \leq -\eta(1 - sa_{\max})\|x\|$. Also, $\|w - v\| = \|A_q(y - x)\| \leq sa_{\max}^2\|x\|$. Hence, by Lemma 36, $c^\top v \leq s(a_{\max}^2 + \eta a_{\max})\|x\| - \frac{\eta}{2}\|x\|$. From the definition of s, this gives $c^\top v \leq \frac{1}{\theta}(V(x) - c^\top x) - \frac{\eta}{2}\|x\|$. Since x, v and c were arbitrary, this concludes the proof. □

C Results in Subsection 5.3

Lemma 37. *For $\gamma < 0$, the system described in Subsection 5.3 is unstable.*

PROOF. Let $x : \pm U\mathbf{1}$, where the sign "$\pm$" is chosen is such a way that $x \in H_2$. It holds that x is an eigenvector of Π_γ with eigenvalue $-\gamma > 0$, so that the trajectory starting from x is divergent. □

Lemma 38. *For $\gamma > 0$, the system described in Subsection 5.3 admits a $2d$-piece polyhedral Lyapunov function, e.g., the polyhedral function $x \mapsto \|U^\top x\|_\infty$.*

PROOF. W.l.o.g. (using a change of coordinates if necessary) assume that $U = I$. Let $V : \|\cdot\|_\infty$. Since $\|\cdot\|_\infty$ is a norm, it is clear that V satisfies (C1) in Proposition 5. To show that V satisfies (C2) in Proposition 5, let $x \in \mathbb{R}^d$. Let $e \in \{0,1\}^d$ and $\sigma \in \{-1,1\}$ be s.t. $e^\top\mathbf{1} = 1$ and $V(x) = \sigma e^\top x$. It holds that $\sigma e^\top \Pi_\gamma x = \sigma\mathbf{1}^\top x - \sigma(d + \bar{\gamma})e^\top x \leq -\sigma\bar{\gamma}e^\top x = -\bar{\gamma}V(x)$, where $\bar{\gamma} \in \{1, \gamma\}$ depending on whether $x \in H_1$ or H_2. Hence, V satisfies (C2) in Proposition 5, concluding the proof. □