



HAL
open science

Robust data-driven Lyapunov analysis with fixed data

Yingzhao Lian, Matteo Tacchi, Colin N. Jones

► **To cite this version:**

Yingzhao Lian, Matteo Tacchi, Colin N. Jones. Robust data-driven Lyapunov analysis with fixed data. 2023. hal-04101941

HAL Id: hal-04101941

<https://hal.univ-grenoble-alpes.fr/hal-04101941>

Preprint submitted on 21 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Robust data-driven Lyapunov analysis with fixed data

Yingzhao Lian¹, Matteo Tacchi^{1,2}, Colin Jones¹

May 21, 2023

Abstract

In this era of digitalization, data has widely been used in control engineering. While stability analysis is a mainstay for control science, most stability analysis tools still require explicit knowledge of the model or a high-fidelity simulator. In this work, a new data-driven Lyapunov analysis framework is proposed. Without using the model or its simulator, the proposed approach can learn a piece-wise affine Lyapunov function with a finite and fixed off-line dataset. The learnt Lyapunov function is robust to any dynamics that are consistent with the off-line dataset. Along the development of proposed scheme, the Lyapunov stability criterion is generalized. This generalization enables an iterative algorithm to augment the region of attraction.

Keywords

Data-driven method, Lyapunov stability analysis

Acknowledgments

This work was supported by the Swiss National Science Foundation under the “NCCR Automation” grant number 51NF40_180545.

The work of M. Tacchi was also supported by the French company RTE, under the RTE-EPFL partnership n°2022-0225.

1 Introduction

Stability analysis is a major research topic in control science. Among various stability criteria, Lyapunov analysis [1] plays a key role in this field. In this framework, stability analysis is reformulated into the search for a Lyapunov function. This has been widely studied in the model-based or sampling-based setup, where the user is assumed to have direct access to the model or its high-fidelity simulator. Note that the system model or its simulator may not be available all the time, it is usually possible and much easier to measure a finite number of system responses offline. Therefore, stability analysis based on fixed measured data poses a new challenge and becomes desirable. Motivated by this need, this work studies the Lyapunov analysis based on a given finite set of measurements of system response. The proposed approach can learn a piece-wise affine (PWA) Lyapunov function on a compact set without access to the system model/simulator. The contributions of this work are summarized as follows:

¹Laboratoire d’Automatique, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.
e-mail: {yingzhao.lian, colin.jones}@epfl.ch

²Univ. Grenoble Alpes, CNRS, Grenoble INP (Institute of Engineering Univ. Grenoble Alpes), GIPSA-lab, 38000 Grenoble, France. e-mail: matteo.tacchi@gipsa-lab.fr

- We formulate and prove a Lyapunov inference theorem, which generalizes existing Lyapunov stability analysis methods. This generalization is used to expand a prior inner estimate of the region of attraction to a larger set.
- We specify our Lyapunov stability criterion for PWA Lyapunov function on a compact set, to make it verifiable locally on this compact set rather than at all points.
- We make the Lyapunov criterion robust to all the models that are consistent with the measured dataset. This criterion is defined on general bounded evaluation function spaces, and has a convex form for Lipschitz functions.
- We develop an algorithm to learn a Lyapunov function robust to all models that are consistent with the measured dataset. The proposed algorithm only needs to solve a convex second-order cone program, regardless of the underlying unknown dynamics (non-)linearity.
- We discuss numerical results properties of the proposed algorithm, mostly focusing on the improvement of its computational efficiency.

Previous Work

Nonlinear Lyapunov stability analysis has been widely studied, where model-based approaches and sampling-based approaches form two main categories. In either approach, a Lyapunov candidate is optimized or synthesized by verifying the Lyapunov stability conditions. In the model-based approaches, the knowledge of the underlying model is explicitly used in the search of the Lyapunov function. On the contrary, the Lyapunov function is trained by penalizing the violation of the Lyapunov stability conditions on a dataset. Even though a standard Monte-Carlo sampling scheme can also give a probabilistic guarantee [2], it is always preferable to give a strict qualification in stability analysis. In this case, both model-based approaches and sample-based approaches require the explicit knowledge of the model. In particular, verification of the Lyapunov stability condition usually resort to nonlinear optimization or satisfiability modulo theory (SMT) solvers, such as dReal [3]. Note that when smooth dynamics are considered, one can write the Lyapunov stability condition with respect to any Lyapunov candidate into an explicit algebraic form (see e.g [4, 5, 6]). The SMT solver is accordingly used to check whether these algebraic inequalities are satisfied up to some user-defined tolerance [7].

To the best of the authors' knowledge, the first numerical method that finds a Lyapunov function solves the Zubov equation [8]. The Zubov equation models the Lyapunov function as the solution to a linear partial differential equation (PDE). The approximation of this PDE is solved by series expansion [8], collocation method [9], etc. One main advantage of the model-based approach is that the a-priori knowledge about the model can be used to reformulate the Lyapunov learning problem into a simpler problem. When polynomial dynamics are considered, a sum of square (SOS) programming relaxation can be used to search for polynomial Lyapunov functions [10]. Due to the nice algebraic properties of polynomials, the SOS framework has been further used to find the region of attraction [11, 12] and its sparsity structure has been used to improve its scalability [13, 14]. Parallel to the studies in polynomial dynamics, PWA dynamics is another genre attracting broad research interest [15, 16]. Such tremendous interest is also a result of the ubiquitous appearances of PWA functions in various controllers, such as ReLU-neural-networks-based controller and linear MPC [17]. For the PWA setup, optimization based approaches play the central role, which mostly applies linear matrix inequality [18, 12, 19] and mixed integer programming [20].

Unlike the model-based approaches, sampling-based methods highly rely on an efficient strategy of generating informative samples. The counter-example guided inductive synthesis

(CEGIS) [21, 22] is one major concept applied behind many sample-based approaches (see e.g. [23, 19, 24]). These approaches have direct access to the model or its simulator. During the learning process, they iteratively augment the sample dataset by adding counter examples to the Lyapunov candidate proposed in the current iteration. These algorithms train the Lyapunov function by penalizing the violation of the Lyapunov stability condition on the samples, and they converge when no further counter example can be generated [23, 5].

The search for a Lyapunov function is usually confined to a specific function class, such as generalized quadratic form [25] or positive definite kernel regressor [26]. In this work, we will focus on PWA Lyapunov functions defined on a compact set. Besides the advantages mentioned in the model-based approach paragraph, PWA Lyapunov candidates have shown nice interplay with Lipschitz dynamics. In particular, when the samples of system dynamics are assigned to the vertices of a grid, a robust Lyapunov stability condition on each simplex can be verified by only considering a tightened Lyapunov condition defined on its vertices. This family of methods is called the continuous piece-wise affine (CPA) method [27, 28]. The CPA method has been extended to more general problem setup: differential inclusion [29], switched system [30], etc. In this work, we also consider Lipschitz dynamics, but we do not assume that the data are located on the vertices. Therefore, we do not term our method a CPA method to avoid unnecessary confusion. More detailed comparison with the CPA method are postponed to Section 4.3.

The rest of this paper is organized as follows: In Section 2, some necessary tools from convex analysis are reviewed alongside the statement of the problem setup. In Section 3, we will first generalize the Lyapunov theorem in Section 3.1, this generalization will later be used to develop a local Lyapunov condition with PWA Lyapunov candidate in Section 3.2. In the sequel, Section 4 applies this local condition to a set of uncertain function defined by data, whose robust satisfaction is summarized as in Theorem 4. This theorem is later used to define a convex inequality condition for the class of Lipschitz function in Section 4.2, where the learning problem will be summarized. A comparison between the proposed learning problem and other related works are given in Section 4.3. The learnability of the proposed scheme is studied in Section 5.1, after which the proposed learning problem is recast into an equivalent form to enable higher computational efficiency in Section 5.2. The general learning algorithm are summarized in 5.3 with a numerical validation in Section 6. A conclusion wraps up this paper in Section 7.

Notation: $\{x_i\}_{i \in \mathcal{I}}$ is a set indexed by \mathcal{I} , and when there is no confusion, we drop the index set with $\{x_i\}$. \mathbb{N}_a denotes the set of positive integer less than a . \mathbb{R}_+ denotes the set of non-negative real numbers. $\mathbf{0}$ is a zero vector. $\mathcal{B}(x, r)$ denotes an open euclidean ball centred at x with radius r . $\text{co}(\mathcal{X})$ denotes the convex hull of $\mathcal{X} \subset \mathbb{R}^n$. $\text{Lip}(\mathcal{X})$ is the set of Lipschitz functions defined on \mathcal{X} .

If $A : \mathcal{H}_x \rightarrow \mathcal{H}_y$ is a linear operator mapping between Hilbert spaces \mathcal{H}_x and \mathcal{H}_y , then A^* denotes its adjoint as $A^* : \mathcal{H}_y \rightarrow \mathcal{H}_x$. $X \setminus Y := \{x \in X \mid x \notin Y\}$ for all $Y \subset X$. $\|x\|$ denotes the norm of $x \in \mathbb{R}^n$, whose dual norm is denoted by $\|x\|^*$, if not specified, $\|\cdot\|$ denotes the euclidean norm (in which case $\|\cdot\|^* = \|\cdot\|$). $|C|$ denotes the cardinality of a set C . If \mathcal{X} is a topological space, then $\text{int}(\mathcal{X})$, $\overline{\mathcal{X}}$ and $\partial\mathcal{X}$ denote its interior, closure and boundary, respectively.

Finally, due to ambiguity in the literature, we indicate our definition of polyhedra and polytopes:

- A polyhedron $P \subset \mathbb{R}^n$ is an intersection of finitely many half spaces: $\exists m \in \mathbb{N}, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ such that

$$P = \{x \in \mathbb{R}^n \mid Ax - b \in (\mathbb{R}_+)^m\}.$$

When the polyhedron is bounded, with slightly abuse of notation, we denote the number of vertices by $|P|$.

- A polytope is a finite union of bounded polyhedra, which is not necessarily convex. Accordingly, a convex polytope is the convex hull of its vertices.

2 Preliminary

In this section, we will first review some results from convex analysis and then introduce the considered stability analysis problem.

2.1 Convex Analysis

Let \mathbb{R}_C^n denote the set of compact convex subsets in \mathbb{R}^n , the support function of a compact set $C \in \mathbb{R}_C^n$ is defined by

$$V_C(g) := \max_{x \in C} g^\top x ,$$

for all $g \in \mathbb{R}^n$, and any convex set can be uniquely characterized by its support function. Meanwhile, the indicator function of a convex set $C \in \mathbb{R}_C^n$ is defined by

$$\iota_C(x) := \begin{cases} 0, & x \in C \\ \infty, & x \notin C, \end{cases}$$

and this function is convex. With this definition, we also have

$$\iota_{C_1 \cap C_2}(x) = \iota_{C_1}(x) + \iota_{C_2}(x) , \quad (1)$$

as $\iota_{C_1 \cap C_2}(x) = 0$ if and only if $\iota_{C_1}(x) = 0$ (i.e. $x \in C_1$) and $\iota_{C_2}(x) = 0$ (i.e. $x \in C_2$), and thus $x \in C_1 \cap C_2$. A conjugate of a convex function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined, for $g \in \mathbb{R}^n$, by

$$h^*(g) := \max_{x \in \mathbb{R}^n} g^\top x - h(x) .$$

By this definition, the conjugate of a non-empty set indicator function is its support function [31, 11.4] as

$$V_C(\cdot) = \iota_C^*(\cdot) .$$

Given two proper, convex functions $h_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ and $h_2 : \mathbb{R}^n \rightarrow \mathbb{R}$, the infimal convolution is defined by

$$h_1 \# h_2(x) := \inf_{y \in \mathbb{R}^n} h_1(y) + h_2(x - y) ,$$

Geometrically, the epigraph of $h_1 \# h_2(\cdot)$ is the Minkowski sum of the epigraph of $h_1(\cdot)$ and $h_2(\cdot)$:

$$\left\{ (x, s) \in \mathbb{R}^n \times \mathbb{R} : h_1 \# h_2(x) < s \right\} = \left\{ (y + z, p + q) \in \mathbb{R}^n \times \mathbb{R} : h_1(y) < p \wedge h_2(z) < q \right\} .$$

In particular, $h_1(\cdot)$ and $h_2(\cdot)$ being proper and convex, so is $h_1 \# h_2$. For the sake of simplicity, we denote

$$\#_i h_i(x) := \inf \sum_i h_i(y_i) \quad \text{s.t.} \quad \sum_i y_i = x .$$

By Lagrangian multiplier, the following calculus of infimal convolution can be derived [32]

Proposition 1. *Let h_1, h_2 be two proper, convex functions, then*

$$(h_1 + h_2)^*(x) = h_1^* \# h_2^*(x) ,$$

2.2 Unknown dynamic system, fixed data

In our problem setup, we consider an **unknown** continuous time dynamic system on a dimension n_x compact set $\mathcal{X} \subset \mathbb{R}^{n_x}$,

$$\frac{dx}{dt} = f(x),$$

which we know has a locally asymptotically stable (LAS) equilibrium point (EP). And we have a size N_D **fixed** dataset sampled from this system as $D = \{(x_i, f_i = f(x_i))\}_{i=1}^{N_D}$ with $x_i \in \mathcal{X}$. The goal is to analyse the stability of this unknown dynamical system based on the fixed dataset D . Without loss of generality, we assume that

Assumption 1. $\mathbf{0} \in \mathcal{X}$ is the LAS EP and we have access to a compact subset \mathcal{X}_s of its region of attraction.

Note that \mathcal{X}_s models a conservative prior about the region of attraction (RoA) of the unknown dynamic system, e.g. deduced from engineering practice. Meanwhile, if no such information is available, one can also consider $\mathcal{X}_s = \{\mathbf{0}\}$. We denote the vector space of Lipschitz continuous functions on n_x dimensional domain \mathcal{X} by $\text{Lip}(\mathcal{X})^{n_x}$. To facilitate the analysis, we further assume that

Assumption 2. $f \in \mathcal{F}$ where $\mathcal{F} \subset \text{Lip}(\mathcal{X})^{n_x}$ is a vector space of Lipschitz continuous functions, such that

$$\forall h \in \mathcal{F}, x \mapsto \|h(x)\| \text{ is bounded within } \mathcal{X},$$

with upper bound denoted $\|h\|_\infty$ (clearly a norm on \mathcal{F}).

Remark 1. Assumption 2 means that the underlying dynamic system is bounded within \mathcal{X} (i.e. there will not exist infinite velocity). Meanwhile, \mathcal{F} does not need to be a Hilbert space, a typical case being the whole $\text{Lip}(\mathcal{X})^{n_x}$. If a Hilbert space \mathcal{F} is used, it is usually assumed to be infinite dimensional to ensure a large modelling capability.

Remark 2. Assumption 2 implies that the evaluation operator defined by

$$E_x(h) := h(x), \forall x \in \mathcal{X}, h \in \mathcal{F}$$

is bounded (i.e. continuous in h) with respect to the norm $\|\cdot\|_\infty$, meaning that

$$\forall x \in \mathcal{X}, \exists C_x \geq 0; \forall h \in \mathcal{F}, \|E_x(h)\| \leq C_x \|h\|_\infty.$$

Indeed, taking $C_x = 1$ for all $x \in \mathcal{X}$ is always possible. However, as soon as \mathcal{F} contains the space of polynomial vector fields, it cannot be complete with respect to this norm $\|\cdot\|_\infty$, as the Stone-Weierstrass theorem ensures that its uniform completion is the whole space of continuous functions (including non-Lipschitz ones). As a result, one might prefer the standard Lipschitz norm

$$\|h\|_{\text{Lip}} := \|h(\mathbf{0})\| + \underbrace{\sup_{x \neq y \in \mathcal{X}} \frac{\|h(x) - h(y)\|}{\|x - y\|}}_{\text{Lipschitz constant}}$$

for which $\text{Lip}(\mathcal{X})^{n_x}$ is complete and the E_x are also bounded:

$$\begin{aligned} \|E_x(h)\| &= \|h(x)\| \\ &= \|h(\mathbf{0}) + h(x) - h(\mathbf{0})\| \\ &\leq \|h(\mathbf{0})\| + \|h(x) - h(\mathbf{0})\| \\ &\leq \|h(\mathbf{0})\| + \|x\| \sup_{y \neq z \in \mathcal{X}} \frac{\|h(y) - h(z)\|}{\|y - z\|} \\ &\leq \underbrace{(1 + \|x\|)}_{C_x} \|h\|_{\text{Lip}} \end{aligned}$$

3 Piecewise affine functions for Lyapunov inference

In this section, we first try to augment the prior knowledge of stability in \mathcal{X}_s to a bigger set \mathcal{X} in Section 3.1. This result is refined to a Lyapunov candidate from the class of piece-wise affine (PWA) functions in Section 3.2.

3.1 Lyapunov inference

Before proceeding to the stability condition, we introduce two additional concepts on functions $L : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$. The sub-level set of L with level $\alpha \in \mathbb{R}$ is given by

$$L_{\leq \alpha} := \{x \in \mathbb{R}^{n_x} \mid L(x) \leq \alpha\}.$$

The Clarke generalized gradient of L at a point $x \in \mathbb{R}^{n_x}$ is the set given by

$$\partial_{\text{Cl}}L(x) := \text{co} \left\{ y \in \mathbb{R}^{n_x} : \begin{array}{l} \forall \epsilon > 0, \exists x_\epsilon \in \mathbb{R}^{n_x}; \\ \|x - x_\epsilon\| < \epsilon, \\ L \text{ is differentiable in } x_\epsilon, \\ \|y - \nabla L(x_\epsilon)\| < \epsilon \end{array} \right\}.$$

In [33, Theorem 2.5.1] it is proved that if L is Lipschitz continuous on a neighbourhood of x , then $\partial_{\text{Cl}}L(x) \neq \emptyset$. The Clarke gradient is a generalized gradient in the sense that if L is continuously differentiable on a neighbourhood of x , then trivially $\partial_{\text{Cl}}L(x) = \{\nabla L(x)\}$.

Now we summarize the Lyapunov stability condition on a compact set \mathcal{X} in the following Lyapunov inference theorem.

Theorem 2. *Let Assumption 1 and 2 hold, and suppose that there exists a Lipschitz continuous function $L : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ such that:*

- 2.1) *There exists $\alpha \in \mathbb{R}$ such that $\mathcal{X} = L_{\leq \alpha}$.*
- 2.2) *$f(x)^\top y < 0$ for all $x \in \mathcal{X} \setminus \mathcal{X}_s$, $y \in \partial_{\text{Cl}}L(x)$.*

Then for all $x_0 \in \text{int}(\mathcal{X})$,

$$x(t|x_0) \xrightarrow[t \rightarrow \infty]{} \mathbf{0}$$

where $t \mapsto x(t|x_0)$ denotes the (unique by Lipschitz continuity of f – see Assumption 2) continuously differentiable solution to the Cauchy problem

$$\begin{cases} \frac{d}{dt}x(t|x_0) = f(x(t|x_0)) \\ x(0|x_0) = x_0. \end{cases}$$

Proof. The proof is similar to the proof of the Lasalle theorem [34]. Let $x_0 \in \text{int}(\mathcal{X})$ and define the entering time

$$\tau(x_0) := \inf\{t \geq 0 : x(t|x_0) \in \mathcal{X}_s\}.$$

If $\tau(x_0) < \infty$, then $x(\tau(x_0)|x_0) \in \mathcal{X}_s$ and using the semigroup property and Assumption 1, one obtains $x(t + \tau(x_0)|x_0) = x(t|x(\tau(x_0))) \xrightarrow[t \rightarrow \infty]{} \mathbf{0}$ so that $x(t|x_0) \xrightarrow[t \rightarrow \infty]{} \mathbf{0}$.

We now proceed to the non-trivial case where $\tau(x_0) = \infty$, i.e. $\forall t \geq 0, x(t|x_0) \notin \mathcal{X}_s$, and consider the exit time $T := \inf\{t \geq 0 : x(t|x_0) \in \partial\mathcal{X}\}$. Notice that as $x_0 \in \text{int}(\mathcal{X})$ and $t \mapsto x(t|x_0)$ is continuous, $T > 0$. Then, using Lipschitz continuity of L , condition 2.2) and [35, Lemma 2.15], for almost all $t \in [0, T)$, $\frac{d}{dt}L(x(t|x_0)) < 0$, so that $L(x(t|x_0))$ is decreasing on $[0, T)$ by the mean value inequality. This yields that $\forall t \in (0, T)$, one has $L(x(t|x_0)) < L(x_0) \leq \alpha$, and by condition 2.1) and continuity of $t \mapsto L(x(t|x_0))$, $T = \infty$ and thus $L(x(t|x_0))$ is decreasing on $[0, \infty)$ and for all $t \geq 0, x(t|x_0) \in \mathcal{X} \setminus \mathcal{X}_s$ regarding $\tau(x_0) = \infty$.

We now consider the compact set $\Omega := \mathcal{X} \setminus \mathcal{X}_s$ from which the trajectory $x(t|x_0)$ never escapes. Since L is continuous and Ω is compact, by the Weierstrass extreme value theorem, one has that

$$-\infty < \ell := \inf_{\Omega} L,$$

so that the function $t \mapsto L(x(t|x_0))$ is decreasing and lower bounded, hence it has a limit $c := \lim_{t \rightarrow \infty} L(x(t|x_0)) \in \mathbb{R}$ that it does not attain in finite time.

We also consider the limit set

$$\Gamma(x_0) := \{p \in \Omega : \forall \epsilon, T > 0, \exists t \geq T; \|x(t|x_0) - p\| < \epsilon\},$$

that has the following property (see [34]): as $t \mapsto x(t|x_0)$ is bounded, $\Gamma(x_0) \subset \Omega$ is nonempty, compact and invariant (forward and backwards). Thus, $\exists p_0 \in \Gamma(x_0)$ and $\forall t \geq 0, x(t|p_0) \in \Gamma(x_0)$. Moreover, continuity of the function L and the definition of $\Gamma(x_0)$ ensure that for all $p \in \Gamma(x_0)$, $L(p) = c$, so that $\forall t \geq 0, L(x(t|p_0)) = c$. By Lipschitz continuity of L and $x(\cdot|p_0)$, we deduce that for almost all $t \geq 0, \frac{d}{dt}L(x(t|p_0)) = 0$. In addition to that, [35, Lemma 2.15] states that for almost all $t \geq 0, \exists y \in \partial_{C^1}L(x(t|p_0))$ such that $0 = \frac{d}{dt}L(x(t|p_0)) = y^\top f(x(t|p_0))$. From these two points and condition 2.2) we deduce that for almost all $t \geq 0, x(t|p_0) \in \mathcal{X}_s$. Additionally, by definition of $\mathcal{X}_s, x(t|p_0) \xrightarrow[t \rightarrow \infty]{} \mathbf{0}$, and by invariance and compactness of $\Gamma(x_0)$, we have $\mathbf{0} \in \Gamma(x_0)$.

Finally, we recall the definition of Assumption 1's local asymptotic stability condition:

$$\exists \epsilon > 0 \text{ s.t. } \|x_1\| < \epsilon \implies x(t|x_1) \xrightarrow[t \rightarrow \infty]{} \mathbf{0}.$$

Moreover, by definition of $\Gamma(x_0) \ni \mathbf{0}$, for all $T > 0$ we are given a $t_0 \geq T$ such that $\|x(t_0|x_0)\| < \epsilon$. Those two observations, taking $x_1 = x(t_0|x_0)$, ensure that

$$x(t|x_0) \xrightarrow[t \rightarrow \infty]{} \mathbf{0}.$$

□

Intuitively speaking, the proof of Theorem 2 considers two cases. The first part with finite entering time $\tau(x_0)$ concerns the case where $\mathbf{0} \in \text{int}(\mathcal{X}_s)$. Meanwhile, the second part of the proof regarding $\tau(x_0) = \infty$ deals with the case where $\mathbf{0} \in \partial\mathcal{X}_s$. While so far numerical considerations lead us to limit to the former case, it is worth noticing that in theory Lyapunov inference can be performed even when the equilibrium point lies on the boundary of the prior region of attraction estimate (including the case $\mathcal{X}_s = \{\mathbf{0}\}$).

Remark 3. The local asymptotic stability condition around $\mathbf{0}$ given in Assumption 1 is necessary. To see this, we construct a two-dimensional counter example:

$$\frac{dx}{dt} = \underbrace{\|x\| \|x - (1, 0)\|^2}_{\text{normal speed}} (-x_2, x_1) + \underbrace{v_r(x)}_{\text{radial speed}} x \quad (2)$$

$$v_r(x) = \begin{cases} \frac{1}{\|x\|} - 1, & \|x\| \geq \frac{3}{4} \\ 1 - \frac{1}{2\|x\|}, & \frac{1}{4} \leq \|x\| < \frac{3}{4} \\ -1, & \|x\| < \frac{1}{4} \end{cases} \quad (3)$$

in the following, we would try to expand the ROA around $x_s = (1, 0)$. Note that to shift this point to $\mathbf{0}$, one only has to do the change of variable $\tilde{x} = x - (1, 0)$.

Now we start to analyse the behaviour of these dynamics in three cases (in addition to the trivial case of equilibrium point).

- If $\|x_0\| = 1$, then $v_r(x_0) = 0$ so that $\|x(t)\|$ is constant (equal to 1) and the normal speed is positive unless / until $x(t) = (1, 0)$.
- If $\|x_0\| = \frac{1}{2}$, again the radial speed is zero and the solution will stay on the circle centred at $\mathbf{0}$ with radius $\frac{1}{2}$, permanently rotating as the normal speed never vanishes. This is an unstable limit cycle.
- If $\|x_0\| > \frac{1}{2}$, then the radial speed will have same sign as $1 - \|x\|$ and $\|x(t)\|$ will converge to 1. In particular, if $\|x_0\| \geq \frac{3}{4}$, then $\|x(t|x_0)\| = 1 + (\|x_0\| - 1)e^{-t}$.
- If $\|x_0\| < \frac{1}{2}$, then the radial speed will stay negative and $x(t)$ will converge to $\mathbf{0}$. In particular, if $\|x_0\| < \frac{1}{4}$, then $\|x(t|x_0)\| = \|x_0\|e^{-t}$.

The discussion above allows us to set $\mathcal{X}_s = \partial\mathcal{B}(0, 1)$: it is true that $\forall x_0 \in \mathcal{X}_s, x(t|x_0) \xrightarrow[t \rightarrow \infty]{} (1, 0)$.

Assuming that $\mathcal{X} = \overline{\mathcal{B}(0, \frac{5}{4})} \setminus \mathcal{B}(0, \frac{3}{4})$, with a Lyapunov candidate $L(x) = |\|x\| - 1|$, we have $L_{\leq \frac{1}{4}} = \mathcal{X}$, which fulfills sublevel set condition 1.1) in Theorem 2. Additionally, on $\mathcal{X} \setminus \mathcal{X}_s$, $\nabla L(x) = \text{sgn}(\|x\| - 1)x$ and thus $f(x)^\top \nabla L(x) = -\|x\||1 - \|x\|| < 0$, satisfying condition 1.2) in Theorem 2. However, $\forall x_0 \in \mathcal{X} \setminus \mathcal{X}_s, x(t|x_0)$ never converges to $(1, 0)$. Instead, it has a limit set $\Gamma(x_0) = \partial\mathcal{B}(0, 1) \subset \partial\mathcal{X}_s$ (see proof of Theorem 2), and the trajectory will circulate forever around the $\partial\mathcal{B}(0, 1)$ without convergence to any point (Fig. 1)

Remark 4. Condition 2.1) is important. To see this, we can assume that $\mathcal{X} \subsetneq L_{\leq \alpha}$, by continuity of $L(x)$, there exists $x_{ext} \notin \mathcal{X}$ such that $L(x_{ext}) \leq \alpha$. For any $x_0 \in \mathcal{X} \setminus \mathcal{X}_s$, we can only ensure that the evaluation of $L(x)$ is decreasing, but it is possible to leave \mathcal{X} heading towards x_{ext} . Note that our condition 2) in Theorem 2 only holds on \mathcal{X} , no stability guarantee can be given in this case.

Remark 5. Theorem 2 is a generalization of the classical Lyapunov-Massera local asymptotic stability theorem, included in it as the case where L is continuously differentiable and $\mathcal{X}_s = \{\mathbf{0}\}$. However we do not assume that $L \geq 0$ with equality only satisfied in $\mathbf{0}$, as $\mathbf{0}$ is already assumed LAS.

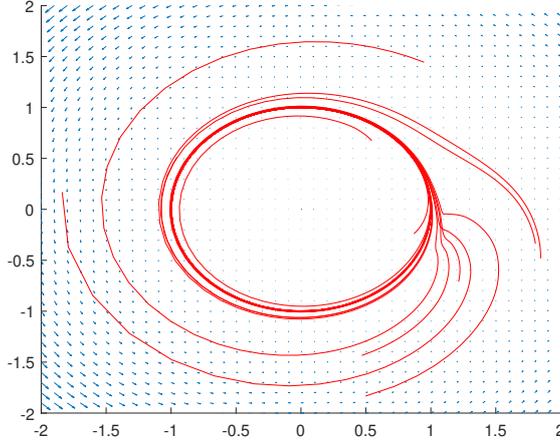


Figure 1: Vector field and trajectories with different initial points of dynamic system (2) that converges to $(1, 0)$.

3.2 Piecewise Affine Lyapunov Function

PWA functions have strong modelling capability because they are dense in the space of continuous functions with a compact domain [36, Chapter 7.4]. This part will refine Theorem 2 to the Lipschitz continuous PWA Lyapunov function. For the sake of simplicity, we further assume

Assumption 3. \mathcal{X} is a polytope.

When \mathcal{X} is not a polytope, it can be inner-approximated by a polytope up to arbitrary accuracy, thus this assumption will not limit the application of the proposed analysis. Additionally, it is worth noting that the definition of polytope used in this paper is not necessarily convex (see Section 2.2).

We now introduce our Lyapunov candidate under the form of a Lipschitz continuous PWA function. Let $\{C_k\}_{k=1}^{N_C}$ be an N_C -piece tessellation of \mathcal{X} (i.e. $\cup_k C_k = \mathcal{X}$ and $\text{int}(C_k) \cap \text{int}(C_\ell) = \emptyset$ if $k \neq \ell$) where the C_k are convex polytopes (without loss of generality we take $C_1 \ni \mathbf{0}$). For $k \in \mathbb{N}_{N_C}$, we denote the vertices of C_k by $\{v_{j,k}\}_{j=1}^{|C_k|}$. Using this structure, a PWA Lyapunov candidate $L_{\mathcal{X}}$ is defined on \mathcal{X} by

$$\forall k \in \mathbb{N}_{N_C}, x \in C_k, L_{\mathcal{X}}(x) = g_k^\top x + b_k. \quad (4)$$

With appropriate conditions on $g_k \in \mathbb{R}^{n_x}, b_k \in \mathbb{R}$, continuity of $L_{\mathcal{X}}$ on \mathcal{X} is enforced: for any common vertex $v \in C_k \cap C_\ell$ (i.e. $\exists i \in \mathbb{N}_{|C_k|}, j \in \mathbb{N}_{|C_\ell|}$ such that $v = v_{i,k} = v_{j,\ell}$), the condition

$$(g_k - g_\ell)^\top v = b_\ell - b_k \quad (5)$$

should hold. Then, $L_{\mathcal{X}}$ is Lipschitz continuous on \mathcal{X} with

$$\|L_{\mathcal{X}}\|_{\text{Lip}} = |b_1| + \max_{1 \leq k \leq N_C} \|g_k\| < \infty.$$

Remark 6. Existence of a set $\{g_k, b_k\}_{k=1}^{N_C}$ such that (4) defines a continuous function is guaranteed for any tessellation of \mathcal{X} , as a constant function on \mathcal{X} always satisfies the continuity condition (i.e. $g_k = \mathbf{0}, \forall k \in \mathbb{N}_{N_C}$ and $b_i = b_j, \forall i, j \in \mathbb{N}_{N_C}$). When the tessellation is defined by a power-diagram, existence of convex PWA functions is further guaranteed [37].

Then, our PWA Lyapunov candidate $L_{\mathcal{X}}$ defined on \mathcal{X} can be extended to the whole \mathbb{R}^{n_x} , such that Theorem 2 can be applied.:

$$L(x) := \begin{cases} \alpha + \eta \operatorname{dist}(x, \mathcal{X}), & x \in \mathbb{R}^{n_x} \setminus \mathcal{X} \\ L_{\mathcal{X}}(x), & x \in \mathcal{X} \end{cases} \quad (6)$$

where $\eta := \max_{1 \leq k \leq N_C} \|g_k\|$ is $L_{\mathcal{X}}$'s Lipschitz constant, and $\alpha \in \mathbb{R}$ is used to define the sublevel set (see Theorem 2) and can be tuned during the Lyapunov analysis.

Regarding our PWA LF candidate, the stability conditions in Theorem 2 can be restated as

Theorem 3. *Let Assumptions 1 to 3 hold, and consider the function $L_{\mathcal{X}}$ and L defined by (4), (5) and (6). If L moreover satisfies the following conditions*

3.1) α is such that the evaluation of L on the vertices satisfies

$$\forall j, k \text{ s.t. } v_{j,k} \in \mathcal{X} \setminus \partial\mathcal{X}, \quad L(v_{j,k}) < \alpha \quad (7a)$$

$$\forall j, k \text{ s.t. } v_{j,k} \in \partial\mathcal{X}, \quad L(v_{j,k}) = \alpha \quad (7b)$$

3.2) $\forall k \in \mathbb{N}_{N_C}, \forall x \in C_k \cap (\mathcal{X} \setminus \mathcal{X}_s),$

$$f(x)^\top g_k < 0 \quad (8)$$

Then L is Lipschitz continuous and for all $x_0 \in \operatorname{int}(\mathcal{X}),$

$$x(t|x_0) \xrightarrow[t \rightarrow \infty]{} \mathbf{0}.$$

Proof. We first prove Lipschitz continuity of L . L is Lipschitz continuous on \mathcal{X} by its definition (5). By (6), as the distance function is 1-Lipschitz, L is Lipschitz continuous on $\mathbb{R}^n \setminus \mathcal{X}$ with same Lipschitz constant as in \mathcal{X} . Then, to prove global Lipschitz continuity of L we only need to prove continuity at the boundary $\partial\mathcal{X}$, which is straightforward using condition (7b). Indeed, by construction of the C_k and Assumption 3, any $x \in \partial\mathcal{X}$ is a convex combination of some vertices $v_{j,k} \in \partial\mathcal{X}: \exists k \in \mathbb{N}_{N_C}, \{\lambda_j\}_{j=1}^{|C_k|} \subset [0, 1]$ s.t.

$$v_{j,k} \in \operatorname{int}(\mathcal{X}) \Rightarrow \lambda_j = 0, \quad \sum_{j=1}^{|C_k|} \lambda_j = 1, \quad \sum_{j=1}^{|C_k|} \lambda_j v_{j,k} = x$$

so that $L(x) = g_k^\top x + b_k = \sum_j \lambda_j \underbrace{(g_k^\top v_{j,k} + b_k)}_{\alpha} = \alpha$, and $L = \alpha$ on $\partial\mathcal{X}$, which is consistent with the limit $\operatorname{dist}(x, \mathcal{X}) \rightarrow 0$ in (6).

We are now going to use Theorem 2 to complete our proof. We first check condition 2.1): $\mathcal{X} = L_{\leq \alpha}$. As $\mathcal{X} \supset L_{\leq \alpha}$ is trivially deduced from (6), we focus on the converse inclusion. Let $x \in \mathcal{X}$ and let us prove that $x \in L_{\leq \alpha}$ i.e. $L(x) \leq \alpha$.

By construction, $\exists k \in \mathbb{N}_{N_C}$ s.t. $x \in C_k$, and thus $\exists \{\lambda_j\}_{j=1}^{|C_k|} \subset [0, 1]$ s.t. $\sum_j \lambda_j = 1$ and $\sum_j \lambda_j v_{j,k} = x$. Again,

$$L(x) = \sum_{j=1}^{|C_k|} \lambda_j \underbrace{(g_k^\top v_{j,k} + b_k)}_{\leq \alpha} \leq \alpha$$

using condition 3.1), which is the announced result.

We then move on to condition 2.2) in Theorem 2. Let $x \in \mathcal{X} \setminus \mathcal{X}_s$ and define $K(x) := \{k \in \mathbb{N}_{N_C} \mid x \in C_k\}$.

If $x \in \text{int}(\mathcal{X})$ and $K(x)$ has a single element k , then $x \in \text{int}(C_k)$ and L is smooth on a neighbourhood of x with $\partial_{\text{Cl}}L(x) = \{g_k\}$ and we conclude using condition 3.2).

Else, if $x \in \text{int}(\mathcal{X})$ but $|K(x)| > 1$, we deduce from the previous point and the definition of the Clarke gradient that

$$\partial_{\text{Cl}}L(x) = \text{co}\{g_k \mid k \in K(x)\}$$

so that $\forall y \in \partial L(x)$, $\exists\{\lambda_k\}_{k \in K(x)} \subset [0, 1]$ s.t.

$$\sum_{k \in K(x)} \lambda_k = 1 \text{ and } \sum_{k \in K(x)} \lambda_k g_k = y.$$

This yields that for such y

$$f(x)^\top y = \sum_{k \in K(x)} \lambda_k \underbrace{f(x)^\top g_k}_{<0} < 0$$

using condition 3.2).

Else, if $K(x)$ has a single element k but $x \in \partial\mathcal{X}$ (i.e. x is in the interior of a facet of \mathcal{X}), we notice that conditions (7a), (7b) enforce that L is constant equal to α on $C_k \cap \partial\mathcal{X}$ and less than α in $\text{int}(C_k)$, which yields, using the definition (4), that g_k is orthogonal to $\partial\mathcal{X}$ in x and points outward of \mathcal{X} . Thus, the unit normal vector to $\partial\mathcal{X} \cap C_k$ at x pointing outward of \mathcal{X} is given by

$$\nu(x) = \frac{g_k}{\|g_k\|}.$$

Let $\epsilon > 0$ and $x_\epsilon \in \mathbb{R}^{n_x} \setminus \mathcal{X}$ s.t. $\|x - x_\epsilon\| < \epsilon$. The definition (6) ensures that L is differentiable in x_ϵ and that

$$\nabla L(x_\epsilon) \xrightarrow{\epsilon \rightarrow 0} \left(\max_{1 \leq \ell \leq N_C} \|g_\ell\| \right) \nu(x) = \left(\max_{1 \leq \ell \leq N_C} \|g_\ell\| \right) \frac{g_k}{\|g_k\|}.$$

denoting

$$\tilde{g}_k := \left(\max_{1 \leq \ell \leq N_C} \|g_\ell\| \right) \frac{g_k}{\|g_k\|} = \frac{\eta}{\|g_k\|} g_k,$$

we then deduce that $\partial_{\text{Cl}}L(x) = \text{co}\{g_k, \tilde{g}_k\} = [1, \eta/\|g_k\|] g_k$ and we conclude the proof using condition 3.2).

Eventually, in the last case where $|K(x)| > 2$ and $x \in \partial\mathcal{X}$ (i.e. x is on the boundary of a facet of \mathcal{X}), from the above we deduce that if $y \in \partial_{\text{Cl}}L(x)$, then $\exists\{\lambda_k\}_{k \in K(x)}, \{\mu_k\}_{k \in K(x)} \subset [0, 1]$ s.t. $\sum_{k \in K(x)} \lambda_k + \mu_k = 1$ and

$$y = \sum_{k \in K(x)} (\lambda_k g_k + \mu_k \tilde{g}_k) = \sum_{k \in K(x)} \left(\lambda_k + \frac{\eta}{\|g_k\|} \mu_k \right) g_k$$

and

$$f(x)^\top y = \sum_{k \in K(x)} \left(\lambda_k + \frac{\eta}{\|g_k\|} \mu_k \right) \underbrace{f(x)^\top g_k}_{<0} < 0$$

using condition 3.2). □

Finally, we would wrap up this part by sorting out the logic flow in this theoretical Section 3 again. The ultimate goal is to extend some prior knowledge of RoA (i.e. \mathcal{X}_s) to a bigger set $\text{int}(\mathcal{X})$ via PWA continuous function, which is not smooth. Theorem 2 gives this characterization

with respect to a continuous Lyapunov candidate via its Clarke gradient evaluation within the set $\mathcal{X} \setminus \mathcal{X}_s$. A specific characterization based on a continuous PWA Lyapunov candidate is then summarized in Theorem 3. This theorem is useful as it allows us to define the Lyapunov candidate only on the region of interest (i.e. \mathcal{X}), while the general Theorem 2 requires the definition of the Lyapunov candidate on the whole state space. Moreover, Theorem 3 reformulates the stability analysis into the analysis on function evaluation on the vertices and the negativity test on each affine piece. Additionally, this negativity test is local with respect to each affine piece, which implies that a local refinement of the Lyapunov candidate $L(x)$ is possible, further discussion of this aspect will be given at the end of the following section.

4 Learning Robust PWA Lyapunov Function

Recall the ultimate goal of this work; we would like to learn a PWA Lyapunov function for an **unknown** Lipschitz dynamic system $f(x)$ based on a given **fixed** dataset $D = \{x_i, f_i\}_{i=1}^{N_D}$. Regarding the underlying dynamic system, we further assume that an overestimate of the Lipschitz constant is given:

Assumption 4. M is a known overestimate of the Lipschitz constant of f , i.e. for all $x, y \in \mathbb{R}^{n_x}$

$$\|f(x) - f(y)\| \leq M\|x - y\|.$$

Remark 7. In the case where \mathcal{F} is a Hilbert space with inner product $\langle \cdot; \cdot \rangle_{\mathcal{F}}$, it is also possible to work with another assumption that might be more convenient in some settings, but introduces additional technicalities in the learning process, namely: M is a known overestimate of the Hilbert norm of f , i.e.

$$\|f\|_{\mathcal{F}} = \sqrt{\langle f; f \rangle_{\mathcal{F}}} \leq M.$$

In order to formulate a tractable learning problem, we further assume:

Assumption 5. The tessellation of $\overline{\mathcal{X} \setminus \mathcal{X}_s}$ is fixed by a given set $\{C_k\}_{k=1}^{N_C}$.

The discussion about why we make this assumption is postponed to Remark 9. In this Section, we will show how to learn the PWA Lyapunov function on the tessellation $\{C_k\}_{k=1}^{N_C}$ based on Theorem 3. In the following subsections, we will gradually develop a Lyapunov function learning scheme under the form of an optimization problem.

Subsection 4.1 studies the negative condition for a general hypothesis space, and Subsection 4.2 studies the condition in the most basic hypothesis space, i.e. the Lipschitz function space.

4.1 Robust Lyapunov Condition

Although condition (8) is local to each affine piece, it still poses a numerically intractable infinite dimensional constraint, especially when the dynamic system is unknown and uncertain as $f \in \mathcal{F}$. In this subsection, we will show how this condition can be relaxed to a more tractable form via the calculus of the infimal convolution.

Based on dynamic evaluation $f_i := f(x_i)$ on location x_i in the dataset D , the hypothesis space of the underlying dynamic system is tightened to

$$\mathcal{F}_D := \{h \in \mathcal{F} \mid E_{x_i} h = f_i, \forall i \in \mathbb{N}_{N_D}\}, \quad (9)$$

where N_D is the number of the collected data points. Accordingly, we further define

$$\mathcal{F}_{D,i} := \{h \in \mathcal{F} \mid E_{x_i} = f_i\}, \quad (10)$$

such that

$$\mathcal{F}_D = \bigcap_i \mathcal{F}_{D,i}. \quad (11)$$

The following theorem is the cornerstone representing the infinite dimensional Lyapunov constraint with a finite number of constraints.

Theorem 4. *The condition $0 > g_k^\top f(x)$ holds for any $x \in C_k$ if there exist a set $\{\tilde{g}_{i,k}\}_{i=1}^{N_D} \subset \mathbb{R}^{n_x}$ such that*

$$\sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k, \text{ and } \sum_{i=1}^{N_D} \max_{x \in C_k} \iota_{\mathcal{F}_{D,i}}^*(E_x^* \tilde{g}_{i,k}) < 0. \quad (12)$$

Proof. The key idea in this proof is to view the negativity condition (8) as an evaluation of a conjugate function in direction g_k :

$$\begin{aligned} 0 > \max_{x \in C_k} \max_{f \in \mathcal{F}_D} g_k^\top f(x) &\stackrel{(a)}{\iff} 0 > \max_{x \in C} \max_{f \in \mathcal{F}} g_k^\top (E_x f) - \iota_{\mathcal{F}_D}(f) \\ &\stackrel{(b)}{\iff} 0 > \max_{x \in C_k} \iota_{\mathcal{F}_D}^*(E_x^* g_k) \\ &\stackrel{(c)}{\iff} 0 > \max_{x \in C_k} \iota_{\bigcap_i \mathcal{F}_{D,i}}^*(E_x^* g_k) \\ &\stackrel{(d)}{\iff} 0 > \max_{x \in C_k} \left(\sum_{i=1}^{N_D} \iota_{\mathcal{F}_{D,i}}^*(E_x^* g_k) \right)^* \\ &\stackrel{(e)}{\iff} 0 > \max_{x \in C_k} \#_i \iota_{\mathcal{F}_{D,i}}^*(E_x^* g_k) \\ &\stackrel{(f)}{\iff} 0 > \max_{x \in C_k} \inf_{\sum \tilde{g}_{i,k} = g_k} \sum_i \iota_{\mathcal{F}_{D,i}}^*(E_x \tilde{g}_{i,k}) \\ &\stackrel{(g)}{\iff} 0 > \inf_{\sum \tilde{g}_{i,k} = g_k} \max_{x \in C_k} \sum_i \iota_{\mathcal{F}_{D,i}}^*(E_x \tilde{g}_{i,k}) \\ &\stackrel{(h)}{\iff} \exists \sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k \text{ s.t. } \max_{x \in C_k} \sum_i \iota_{\mathcal{F}_{D,i}}^*(E_x^* \tilde{g}_{i,k}) < 0. \end{aligned}$$

(a) writes the feasible set \mathcal{F}_D into the objective function by indicator function, and (b) follows the definition of the convex conjugate (see Section 2.1). (c) applies the decomposition given in (11), and (d) applies the calculus of indicator functions given in (1). (e) applies calculus of conjugate function in Proposition 1, whose explicit form is written in (f). (g) applies the min-max inequality [38, Chapter 3.14]. Finally, as often performed in robust optimization (see e.g. [39]), the infimum operator is replaced by an existence assertion in (h). \square

The key concept behind Theorem 4 is the decomposition of the hypothesis space in (11). In particular, $\iota_{\mathcal{F}_{D,i}}^*(E_x^* \tilde{g}_{i,k}) = \max_{f \in \mathcal{F}_{D,i}} g_{i,k}^\top f(x)$ is related to the uncertainty quantified from one data point, which usually has an easy-to-evaluate explicit closed solution. In comparison,

the explicit solution is usually not available or difficult to evaluate when the whole dataset D is considered. For example, when \mathcal{F} is the space of Lipschitz functions, then the uncertainty boundary quantified by one data point defines a shifted cone. However, the uncertainty upper and lower bounds are PWA and non-trivial to evaluate [40] when the whole dataset D is used. Recall Assumption 2, it is also reasonable to consider a reproducing kernel Hilbert space (RKHS)¹, which underpins various uncertainty quantification methods such as Gaussian process regression [42] and deterministic error bound methods [43]. All these methods require computing the inverse of the Gram matrix or solving a second order cone program, which has an easy-to-evaluate explicit solution only when one data point is considered.

Remark 8. In this work, we consider a decomposition of the hypothesis space \mathcal{F}_D into the intersection of single-data models, *i.e.* $\mathcal{F}_D = \cap \mathcal{F}_{D,i}$. It is also possible to generalize the result to other decomposition, dubbed $\{\mathcal{F}_j\}$. If $\mathcal{F} = \cap_j \mathcal{F}_j$, where the conclusion in Theorem 4 needs slight modification accordingly:

$$\sum_j \tilde{g}_{j,k} = g_k, \quad 0 > \sum_j \max_{x \in C_k} \iota_{\mathcal{F}_j}^*(E_x^* \tilde{g}_{j,k}).$$

Remark 9. Even though function evaluation on a fixed location defines a linear operator from the hypothesis space \mathcal{F} to \mathbb{R}^{n_x} , the mapping from the evaluation point to this operator is in general nonlinear. Assumption 5 is posed to develop a tractable formulation by avoiding this nonlinear mapping, and the next subsection will make use of this property. It is also noteworthy that, with a fixed tessellation, the parameters of the Lyapunov candidate on each affine piece (*i.e.* g_k, b_k on C_k) can be uniquely determined by the function evaluation on the vertices.

Another main benefit of a fixed tessellation is that it allows a direct control over the model complexity of the Lyapunov candidate. In particular, consider two Lyapunov candidates $L_1(x)$ and $L_2(x)$ with their corresponding partitions $\{C_{1,k}\}$ and $\{C_{2,k}\}$. Then, we can state that $L_1(x)$ is a refinement of $L_2(x)$ (*i.e.* $L_1(x)$ has a higher degree of modelling capability than $L_2(x)$) if $\forall C_{2,k}, \exists \{C_{1,j}\}_{j \in \mathcal{I}_k}$ such that $\cup_j C_{1,j} = C_{2,k}$. As condition (8) is local to each affine piece, if one affine piece C_k violates the assumptions of Theorem 4, then we can refine the model locally by further partition this piece.

4.2 A Convex Tractable Case: Lipschitz Dynamics

Theorem 4 gives us a representation of condition (8), but such a representation remains abstract and hard to check numerically; thus, we will now recast this representation under a tractable form, in a specific case. Although we have discussed that it is possible to consider a more complex function space property on top of the Lipschitz property in Section 4.1, this section will show that a convex learning problem exists even when we consider the most basic hypothesis space:

$$\mathcal{F} = \text{Lip}(\mathcal{X})^{n_x} \tag{13}$$

In such case, the following corollary holds:

Corollary 5. *The condition $0 > g_k^\top f(x)$ holds for any $x \in C_k$ if there exists a set $\{\tilde{g}_{i,k}\}_{i=1}^{N_D} \subset \mathbb{R}^{n_x}$ such that $\sum_i \tilde{g}_{i,k} = g_k$ and for all $j \in \mathbb{N}_{|C_k|}$*

$$\sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \|v_{j,k} - x_i\| < 0. \tag{14}$$

¹A reproducing kernel Hilbert space (RKHS) over a set X is a Hilbert space of functions from X to \mathbb{R} such that for each $x \in X$, the evaluation functional $E_x g := g(x)$ is bounded [41]

Proof.

$$\begin{aligned}
\forall j, 0 > \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \|v_{j,k} - x_i\| &\stackrel{(a)}{\iff} 0 > \max_{x \in C_k} \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \|x - x_i\| & (15) \\
&\stackrel{(b)}{\iff} 0 > \max_{x \in C_k} \sum_{i=1}^{N_D} \max_{y_i \in \mathcal{B}(f_i, M \|x - x_i\|)} \tilde{g}_{i,k}^\top y_i \\
&\stackrel{(c)}{\implies} 0 > \max_{x \in C_k} \sum_{i=1}^{N_D} \max_{f \in \mathcal{F}_{D,i}} \tilde{g}_{i,k}^\top f(x) \\
&\stackrel{(d)}{\iff} 0 > \max_{x \in C_k} \sum_{i=1}^{N_D} \iota_{\mathcal{F}_{D,i}}^*(E_x^* \tilde{g}_{i,k}).
\end{aligned}$$

To show (a), we notice that the right-hand side of (15) is a convex maximization problem over a bounded convex polytope, its optimal solution is attained on its vertices, i.e. $\{v_{j,k}\}$. (b) uses the Cauchy-Schwarz inequality on the second term of the sum $\tilde{g}_{i,k}^\top y_i = \tilde{g}_{i,k}^\top f_i + \tilde{g}_{i,k}^\top (y_i - f_i)$. (c) follows the assumption of Lipschitz constant overestimate (Assumption 4). Finally, (d) applies the definition of the conjugate function. \square

Remark 10. If instead of Assumption 4, we suppose that $\|f\|_{\mathcal{F}} \leq M$, then one has a bound function $B_M : \mathbb{R}^{n_x} \rightarrow \mathbb{R}_+$ (depending on M) such that for any $x \in \mathbb{R}^{n_x}$, $h^* \in \operatorname{argmin}_{h \in \mathcal{F}_D} \|h\|_{\mathcal{F}}$,

$$\|f(x) - h^*(x)\| \leq B_M(x),$$

so that in the previous proof one has to replace $M\|x - x_i\|$ with $B_M(x)$. However, the problem here is that B_M is not convex, so that relation (b) does not hold anymore, and we would need other arguments (out of the scope of this article) to obtain a finite dimensional constraint.

Using Theorem 3 and Corollary 5, we get the following conditions for $\operatorname{int}(\mathcal{X})$ to be a positively invariant subset of the region of attraction of our unknown system:

$$\begin{aligned}
\exists \{\tilde{g}_{i,k} \mid i \in \mathbb{N}_{N_D}, k \in \mathbb{N}_{N_C}\} \subset \mathbb{R}^{n_x}, & \quad \exists \{g_k \mid k \in \mathbb{N}_{N_C}\} \subset \mathbb{R}^{n_x}, \\
\exists \{b_k \mid k \in \mathbb{N}_{N_C}\} \subset \mathbb{R}, & \quad \exists \alpha \in \mathbb{R} \\
\forall k, \ell \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|} \text{ s.t. } v_{j,k} \in C_\ell, & \quad (g_k - g_\ell)^\top v_{j,k} = b_\ell - b_k & (5)
\end{aligned}$$

$$\forall k \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, \text{ s.t. } v_{j,k} \in \mathcal{X} \setminus \partial \mathcal{X} \quad g_k^\top v_{j,k} + b_k < \alpha \quad (7a)$$

$$\forall k \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, \text{ s.t. } v_{j,k} \in \partial \mathcal{X} \quad g_k^\top v_{j,k} + b_k = \alpha \quad (7b)$$

$$\forall k \in \mathbb{N}_{N_C}, \quad \sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k \quad (12)$$

$$\forall k \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, \quad \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \|v_{j,k} - x_i\| < 0. \quad (14)$$

Hence, our Lyapunov inference now boils down to a finite number of equality and strict inequality tests. Regarding condition (14) for a fixed $k \in \mathbb{N}_{N_C}$, we introduce slack variables to transform this certification problem into an optimization problem, with optimality giving the best robust certificates possible. This results in the following optimization problem:

$$s_{\alpha,\epsilon}^* := \min_{\{g_k, b_k\}, \{\tilde{g}_{i,k}\}, \{s_{j,k}\}} \sum_{k=1}^{N_C} \sum_{j=1}^{|C_k|} s_{j,k}$$

$$\forall k, \ell \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, s_{j,k} \geq -\epsilon \quad (16a)$$

$$v_{j,k} \in C_\ell \implies (g_k - g_\ell)^\top v_{j,k} = b_\ell - b_k \quad (16b)$$

$$v_{j,k} \in \mathcal{X} \setminus \partial\mathcal{X} \implies g_k^\top v_{j,k} + b_k \leq \alpha - \epsilon \quad (16c)$$

$$v_{j,k} \in \partial\mathcal{X} \implies g_k^\top v_{j,k} + b_k = \alpha \quad (16d)$$

$$\sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k \quad (16e)$$

$$\sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \|v_{j,k} - x_i\| \leq s_{j,k} \quad (16f)$$

where $\epsilon > 0$ is a user-defined negativity tolerance and $\alpha \in \mathbb{R}$ is the user-defined maximal value of the Lyapunov function. Constraint (16b) is the continuity condition, and constraint (16c) is the interior condition (7a) and constraint (16d) is the boundary condition (7b), both stated in Theorem 3. When the slack variables satisfy $s_{j,k} < 0$, constraints (16e) and (16f) correspond to the negative condition (8) stated in Theorem 3. Optimization problem (16) comes with the following result:

Theorem 6. $\forall h \in \mathcal{F}$ whose function evaluations are consistent with the unknown underlying dynamic system f ($h(x_i) = f_i, \forall x_i, f_i \in D$), the solution to problem (16) defines a Lyapunov function for dynamic system h on \mathcal{X} when its optimal value verifies $s_{\alpha,\epsilon}^* = -\epsilon \sum_{k=1}^{N_C} |C_k|$.

Proof. We need to show that an optimal solution $\{g_k^{\text{opt}}, b_k^{\text{opt}}, \tilde{g}_{i,k}^{\text{opt}}, s_{j,k}^{\text{opt}}\}$ satisfying

$$\sum_{j,k} s_{j,k}^{\text{opt}} = -\epsilon \sum_k |C_k|$$

will also satisfy the conditions given in Theorem 3.

First of all, by constraint (16b) the learnt PWA function is continuous (see Equation (4)). On top of this, by constraints (16d) and (16c), a solution to problem (16) recovers the condition 3.1) in Theorem 3. In the rest of this proof, we need to show that constraints (16e) and (16f) are equivalent to the negative condition 3.2) in Theorem 3 (i.e. $0 > g_k^\top f(x), \forall x \in C_k$).

Note that, by $s_{j,k} \geq -\epsilon$ (constraint (16a)), the optimal value $\sum_{j,k} s_{j,k}^{\text{opt}} = -\epsilon \sum_{k=1}^{N_C} |C_k|$ implies that $s_{j,k} = -\epsilon, \forall j \in \mathbb{N}_{|C_k|}, k \in \mathbb{N}_{N_C}$. Therefore, we have

$$0 > -\epsilon \geq \sum_i i = 1^{N_D} \tilde{g}_{i,k}^\top f_i + M \|\tilde{g}_{i,k}\| \|v_{j,k} - x_i\|$$

$$\stackrel{(a)}{\implies} 0 > \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + M \|\tilde{g}_{i,k}\| \|v_{j,k} - x_i\|$$

$$\stackrel{(b)}{\implies} 0 > g_k^\top h(x) \quad \forall x \in C_k, h \in \mathcal{F}_D,$$

where (a) holds by taking $\tilde{g}_{i,k} = \mathbf{0}, \forall i$ such that $(x_i, f_i) \notin D$ and (b) is Corollary 5. Based on this, we can derive:

In summary, constraints (16f) and (16e) guarantees the satisfaction of condition 3.2) in Theorem 3 for any $h \in \mathcal{F}_D$. Hence, we conclude the proof. \square

The implication of this theorem is strong as it states that under the assumption of Lipschitz dynamics, we can learn/validate a LF by convex programming (16) even when the unknown underlying dynamic system f is nonlinear.

Remark 11. It is possible to adapt the proposed scheme (16) to the case where the measurements are contaminated by bounded measurement noise. More specifically, if the noisy measurement $\{\tilde{x}_i, \tilde{f}_i\}$ is disturbed by measurement noise bounded by η (i.e. the noise-free data lies at $\{x_i, f_i\}$, such that $\|x_i - \tilde{x}_i\| \leq \eta$ and $\|f_i - \tilde{f}_i\| \leq \eta$). The constraint (16f) is accordingly modified to

$$\sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| (M(\|v_{j,k} - x_i\| + \eta) + \eta) \leq s_{j,k}.$$

For the sake of a clear presentation, we only consider the noise-free measurement in the rest of this paper.

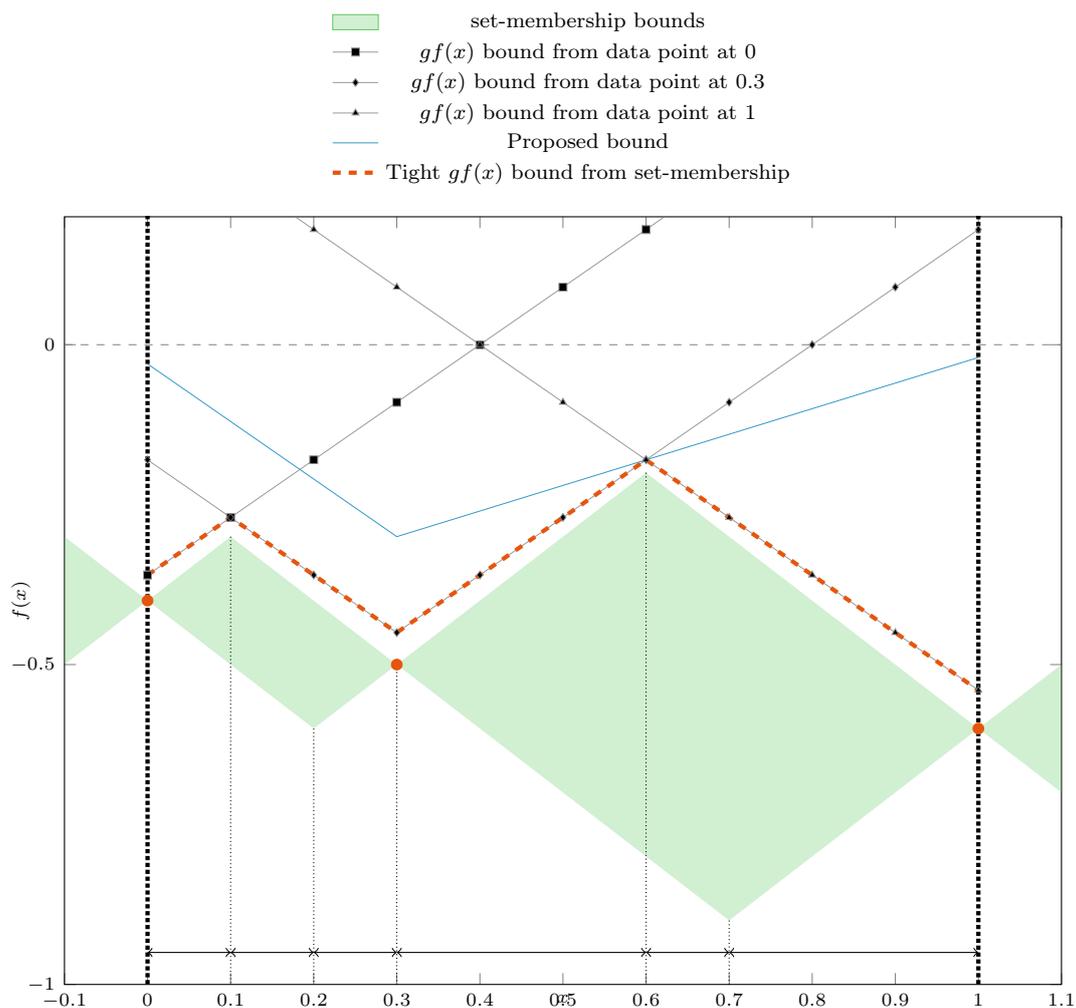


Figure 2: Comparison between set membership method and the proposed method. The proposed scheme is evaluated by $\tilde{g}_1 = 0, \tilde{g}_2 = 0.65, \tilde{g}_3 = 0.35$.

4.3 Comparison with related works

We would like to wrap up this subsection by comparing the proposed learning scheme with other existing methods. In comparison with other PWA Lyapunov function based methods (see e.g. [29, 27]), the proposed scheme shows two major differences. First, the location of the samples and the tessellation of the PWA Lyapunov candidate are decoupled in the proposed scheme. While in existing methods, the data are sampled on the vertices of the tessellation, therefore, the data locations are usually structural due to the choice of the tessellation. Secondly, the robust Lyapunov stability conditions considered in the existing methods only consider the model uncertainty quantified by one data point. On the contrary, the proposed scheme synthetically makes use of the uncertainty quantified by each data point while maintaining a convex tractable structure.

Another framework related to the proposed approach is the set-membership method [40]. In short, the set-membership method models the set of Lipschitz functions whose evaluation on the points $\{x_i\}$ are consistent with data $\{x_i, f(x_i)\}$. The evaluation upper/lower bounds given by this method are PWA. To the best of our knowledge, even though the set-membership method could be extended to vector-valued functions, its typical applications are still limited to real-valued functions [44, 45] (i.e. $f(x) \in \mathbb{R}$). This is one major difference between the proposed scheme and the set-membership method. Now, we will show their difference regarding real-valued functions. To better demonstrate the difference, we consider a specific example in \mathbb{R} (Figure 2), whose Lipschitz overestimate is set to $M = 1$ and the data points are:

$$\{(0, f(0) = -0.4), (0.3, f(0.3) = -0.5), (1, f(1) = -0.6)\}$$

Now, consider a Lyapunov function candidate $L(x) = gx$ with $g = 0.9$ within interval $[0, 1]$. If the evaluation bounds of set-membership method are used, the Lyapunov decreasing condition needs to be examined in all the sub-intervals generated by the PWA bounds (plotted as two-headed arrow in Figure 2). Determination of these sub-intervals is computationally heavy. Instead, if we hope to simplify the analysis by only taking one data point into consideration, none of the simple model generated by one data point can justify the Lyapunov decreasing condition (see black lines of different markers in Figure 2). These two aspects together imply the use of the whole dataset is necessary. The proposed scheme synthesizes the knowledge of simple models via a convex optimization. One optimal solution to the proposed scheme is plotted as a blue line in Figure 2, which only utilizes the last two points (i.e. $\tilde{g}_1 = 0$). It is worth noting that, in this example, if we only consider the left data and the right data point, the Lyapunov decreasing condition will fail even when the set-membership method is used. Hence, we can observe that the proposed method is able to search for the data points that are relevant to the Lyapunov decreasing condition. Additionally, this process is done by polynomial time convex optimization algorithms [46]. On the contrary, even though the set-membership method gives the tightest bound, checking the Lyapunov decreasing condition with these bounds is NP-hard, as it requires vertex elimination of the Voronoi cells.

Remark 12. Note that in the proof of Theorem 5, we use the Cauchy-Schwarz inequality in (15), that holds only with Euclidean 2-norm. Actually, other norms can be considered, and the resulting problem (16) will have different properties accordingly. In particular, if 1-norm or ∞ -norm is used, the resulting problem is a linear program. For the sake of simplicity, this paper uses the Euclidean norm only, and as a result, the learning problem (16) is a second order cone programming (SOCP) (Details in Section. 5).

5 Algorithm Development

After the introduction of the Lyapunov learning problem (16), we will discuss its learnability in Section 5.1. The original learning problem (16) will be recast to an equivalent but numerically more efficient form in Section 5.2. In the end, the main algorithms are summarized in Section 5.3.

5.1 Learnability

Above all, the set of Lyapunov functions is closed under positive scaling. More specifically, if a Lyapunov function $L(x)$ is learnt from problem (16), then its positive scaling $\lambda L(x)$ with $\lambda > 0$ is also a Lyapunov function. Thus, the user-defined value α in (16c) will not introduce conservativeness in the learning problem. Furthermore, due to the introduction of the slack variables $\{s_{j,k}\}$, the learning problem (16) is always feasible. It is natural to ask the following core question in the limiting case:

If \mathcal{X} is the RoA of the underlying dynamic system f and we are allowed to evaluate the dynamic system f in any finite set of points $x \in \mathcal{X}$, can we always learn a Lyapunov function for any RoA subsets $\mathcal{X} \setminus \mathcal{X}_s$?

Unfortunately, the answer to this question is **no**, and a counterexample is summarized in the following Corollary:

Corollary 7. *Let Assumption 1 hold and let the hypothesis space \mathcal{F} satisfy Equation (13), if $\mathbf{0} \notin \text{int}(\mathcal{X}_s)$, then the optimal value to problem (16) verifies:*

$$\sum_{j,k} s_{j,k}^{\text{opt}} > -\epsilon \sum_{k=1}^{N_C} |C_k|$$

Proof. Note that the user-defined tolerance is negative (i.e. $-\epsilon < 0$) and arbitrary, we need to show that there exist one $s_{j,k}^{\text{opt}} \geq 0$ in the optimal solution.

By Assumption 1, we have $f(\mathbf{0}) = \mathbf{0}$. Using the Lipschitz condition and Assumption 4 yields

$$\|f_i\| = \|f_i - \mathbf{0}\| \leq M \|x_i - \mathbf{0}\|$$

As $\mathbf{0}$ is the LAS EP, we have $\mathbf{0} \in \mathcal{X}_s$, and by assumption $\mathbf{0} \notin \text{int}(\mathcal{X}_s)$, so $\mathbf{0} \in \partial\mathcal{X}_s$. Consider the affine piece C_k that contains $\mathbf{0}$ (i.e. $\mathbf{0} \in C_k$ and it is a vertex of C_k). By inspecting the constraint (14) and using Cauchy-Schwarz inequality, one has:

$$\begin{aligned} & \tilde{g}_{i,k}^\top f_i + M \|\tilde{g}_{i,k}\| \|\mathbf{0} - x_i\| \\ & \geq M \|\tilde{g}_i\| \|\mathbf{0} - x_i\| - \|\tilde{g}_i\| \|f_i\| \geq 0. \end{aligned}$$

Thus, the constraint (16f) will become:

$$s_{j,k} \geq \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + M \|\tilde{g}_{i,k}\| \|v_{j,k} - x_i\| \geq 0 > -\epsilon,$$

which is sufficient to conclude the proof. \square

Even though our theory of expanding a-priori knowledge from \mathcal{X}_s to a bigger set \mathcal{X} holds for any \mathcal{X}_s (Section 3), Corollary 7 shows that, if the domain of a Lyapunov function contains $\mathbf{0}$, then it is impossible to learn this function by only assuming Lipschitz continuity (Equation (13)).

Similar observation of infeasibility was also made in the constructive proof of converse Lyapunov theorem [47]. Meanwhile, many numerical methods also have similar limitation around the invariant set [48, Chapter 2.11]; we refer the interested reader to [49, 50] for more details. In summary, if we do not further assume other function space structure on the hypothesis space \mathcal{F} , following assumption is required to ensure the learnability of problem (16):

Assumption 6. $\mathbf{0} \in \text{int}(\mathcal{X}_s)$.

We would stress again that the assumption above is only necessary for the learning scheme based on problem (16), and a learning scheme without this assumption is left for future research.

After answering the aforementioned problem by Corollary 7, the follow-up core question is:

Given an RoA prior estimate \mathcal{X}_s , what condition on dataset D should hold to enable learning the PWA Lyapunov function on $\mathcal{X} \setminus \mathcal{X}_s$?

Obviously, it is impossible to answer this question with a sufficient condition, regarding the arbitrariness of the unknown dynamic system f . However, if we only assume the function space to be Lipschitz (Equation (13)), we can still give an initial check of the learnability of problem (16). In order to discuss this necessary condition, we first define

$$r_i := \frac{\|f_i\|}{M}.$$

We state the necessary condition as follows:

Lemma 8. *Let Assumption 1 hold and let hypothesis space $\mathcal{F} = \text{Lip}(\mathcal{X})^{n_x}$. If solutions to problem (16) define a Lyapunov function as in Theorem 3, then $\mathcal{X} \setminus \mathcal{X}_s \subset \cup_{i=1}^{N_D} \mathcal{B}(x_i, r_i)$.*

Proof. From the proof of Theorem 6, the solution to problem (16) is a Lyapunov function if for $\forall k \in \mathbb{N}_{N_C}$, $j \in \mathbb{N}_{|C_k|}$, $s_{j,k} = -\epsilon$. Let $x \in \mathcal{X} \setminus \mathcal{X}_s$, and without loss of generality, we suppose $x \in C_k$ for some $k \in \mathbb{N}_{N_C}$. By Corollary 5,

$$0 > -\epsilon = s_{j,k} \geq \sum_i i = 1^{N_D} \tilde{g}_{i,k}^\top f_i + M \|\tilde{g}_{i,k}\| \|x - x_i\|,$$

which implies that there exist at least one $n \in \mathbb{N}_{N_d}$ such that

$$\begin{aligned} & 0 > \tilde{g}_{n,k}^\top f_n + M \|\tilde{g}_{n,k}\| \|x - x_n\| \\ \implies & |\tilde{g}_{n,k}^\top f_n| > M \|\tilde{g}_{n,k}\| \|x - x_n\| \\ \stackrel{(a)}{\implies} & \|f_n\| \|\tilde{g}_{n,k}\| > M \|\tilde{g}_{n,k}\| \|x - x_n\| \\ \iff & x \in \mathcal{B}\left(x_n, \frac{\|f_n\|}{M}\right) \\ \iff & x \in \mathcal{B}(x_n, r_n) \\ \implies & x \in \cup_{i=1}^{N_D} \mathcal{B}(x_i, r_i) \end{aligned}$$

where (a) follows from the Cauchy-Schwarz inequality. Due to the inclusion condition is satisfied for any point $x \in \mathcal{X} \setminus \mathcal{X}_s$, we conclude the proof with $\mathcal{X} \setminus \mathcal{X}_s \subset \cup_i^{N_D} \mathcal{B}(x_i, r_i)$. \square

This lemma shows the connection between learning a PWA Lyapunov function and the set covering problem, which was proved to be equivalent to a non-convex semi-infinite problem [51], and thus one should not try to check the condition in Lemma 8 numerically. Recall a key idea behind the problem (16): the global analysis on $\mathcal{X} \setminus \mathcal{X}_s$ is reduced to the analysis on the vertices. This inspires us to relax the continuous set covering problem to the covering problem of the vertices and we state this condition in the following Corollary

Corollary 9. *Let Assumption 1 hold and let hypothesis space $\mathcal{F} = \text{Lip}(\mathcal{X})^{n_x}$, if solutions to problem (16) define a Lyapunov function as in Theorem 3, then*

$$\{v_{i,k}\}_{\substack{1 \leq k \leq N_C \\ 1 \leq i \leq |C_k|}} \subset \cup_{i=1}^{N_D} \mathcal{B}(x_i, r_i), \quad (17)$$

This necessary condition (17) can be checked in polynomial time. If this test fails, it means that there exist $v_{i,k} \notin \cup_{i=1}^{N_D} \mathcal{B}(x_i, r_i)$, and therefore, the data is not informative enough to learn a PWA Lyapunov function by only assuming the Lipschitz condition (Equation (13)). Accordingly, the learning process will be terminated. Intuitively, the points which $v_{i,k} \notin \cup_{i=1}^{N_D} \mathcal{B}(x_i, r_i)$ should suggest the location where additional samples are required. We leave the investigation about this aspect for future work.

5.2 Computationally efficient recasting

In this part, we will discuss how we recast the original problem (16) to an equivalent problem that can be handled numerically more efficiently.

Data Refinement

One main computational bottleneck for the original problem (16) comes from the number of decision variables. Without loss of generality, we consider an affine piece C_k . By inspecting the tessellation validation test (17), if a data point $(x_i, f(x_i))$ does not contain any vertices of C_k in $\mathcal{B}(x_i, r_i)$ (i.e. $C_k \ni v_{j,k} \notin \mathcal{B}(x_i, r_i)$), then $\forall \tilde{g} \in \mathbb{R}^{n_x}$, following inequality holds

$$0 \leq \tilde{g}^\top f_i + M \|\tilde{g}\| \|v_{j,k} - x_i\|.$$

Hence, this data point x_i cannot help enforce the strict negative Lyapunov decreasing condition (16f), and can therefore be neglected in the constraints defined on affine piece C_k . Accordingly, the set of data points relevant to C_k are:

$$D_k := \{(x_i, f_i) \mid i \in I_k\} \quad \text{with} \quad I_k := \{i \in \mathbb{N}_{N_D} \mid \exists j \in \mathbb{N}_{|C_k|}, v_{j,k} \in \mathcal{B}(x_i, r_i)\}$$

The condition defining this set essentially states the point should at least be possible to enforce strict negativity on constraint (16f) at one vertex. This technique can significantly reduce the computational cost. To see this, we consider a homogeneous tessellation within a unit hypercube centred at $\mathbf{0}$ within which data points scatter uniformly. We further assume that the affine pieces of the tessellation are hypercubes with edge width *rel*. Based on the Lipschitz condition (13), each data point will at most get involved in $O\left(\left(\frac{r_i}{rel}\right)^{n_x}\right) \ll N_C$. Notice that $r_i \leq .5$ (see Section 5.1), the number of decision variables are reduced to roughly $O\left(\frac{1}{2^{n_x}}\right)$ of the problem defined by the whole data set. In the numerical example we consider in Section 6, we observe on average an 81% reduction in the number of decision variables, which makes the problem tractable on a Laptop without memory overflow.

Explicit SOCP formulations

In the numerical implementation, it is critical to convert the inequality constraint (16f) into a set of second-order cones, or Lorentz cone in particular [38]:

$$\begin{aligned} s_{j,k} &\geq \sum_{i \in I_k} \tilde{g}_{i,k}^\top f_i + t_{i,j,k} \\ t_{i,j,k} &\geq \|\tilde{g}_{i,k}\| M \|v_{j,k} - x_i\|, \end{aligned}$$

where $|I_k|$ auxiliary decision scalar variables $\{t_{i,j,k}\}$ are introduced per vertex $v_{j,k}$ in order to define the Lorentz cone. The resulting computational complexity per iteration in an interior

point algorithm is $O(N_D N_C n_x^2)$ [52]. In comparison, without this reformulation, this inequality constraint will be directly handled by a block diagonal positive semi-definite matrix, whose computational complexity per iteration in an interior point algorithm is $O(|C_k| (N_D n_x)^3)$ [53, Chapter 1].

The Recast Problem

After introducing of the reformulation techniques, the learning problem we solved becomes:

$$s_{\alpha, \epsilon}^* := \min_{\substack{\{g_k, b_k\}, \{\tilde{g}_{i,k}\} \\ \{s_{j,k}\}, \{t_{i,j,k}\}}} \sum_{k=1}^{N_C} \sum_{j=1}^{|C_k|} s_{j,k} \quad (18a)$$

$$\forall k, \ell \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, s_{j,k} \geq -\epsilon \quad (18b)$$

$$v_{j,k} \in C_\ell \implies (g_k - g_\ell)^\top v_{j,k} = b_\ell - b_k \quad (18c)$$

$$v_{j,k} \in \mathcal{X} \setminus \partial\mathcal{X} \implies g_k^\top v_{j,k} + b_k \leq \alpha - \epsilon \quad (18d)$$

$$v_{j,k} \in \partial\mathcal{X} \implies g_k^\top v_{j,k} + b_k = \alpha \quad (18e)$$

$$\sum_{i \in I_k} \tilde{g}_{i,k} = g_k \quad (18f)$$

$$\sum_{i \in I_k} \tilde{g}_{i,k}^\top f_i + t_{i,j,k} \leq s_{j,k} \quad (18g)$$

$$\forall i \in I_k, \|\tilde{g}_{i,k}\| \|M\| \|v_{j,k} - x_i\| \leq t_{i,j,k} \quad (18h)$$

Remark 13. Learning without knowing M : It is possible to consider M as a decision variable, and determine the largest Lipschitz constant for which a Lyapunov function can be found given a particular set of data. More specifically, constraint (18h) is recast to a positive semi-definite constraint by

$$\begin{bmatrix} M I_{n_x} & \tilde{g}_{i,k} \\ \tilde{g}_{i,k}^\top & t_{i,j,k} \end{bmatrix} \in \mathcal{S}^+, \quad (19)$$

where \mathcal{S}^+ denotes the set of positive semi-definite matrices. However, once this formulation is used, the resulting optimization problem becomes a semidefinite programming (SDP) and the data refinement technique proposed at the beginning of this Section 5.2 cannot be applied. Even though the particular sparsity structure in (19) can be exploited to improve the computational efficiency, the computational cost of this optimization still drastically increase in comparison with the SOCP problem (16).

5.3 Algorithms

The final learning algorithm is summarized in Algorithm 1. Even though we use the standard tessellation algorithm in algorithm 1, generating a good tessellation is vital but non-trivial. Existing works mostly focus on the link between a convex liftable tessellation and the power diagram (see e.g. [54, 55]). However, as the Lyapunov function studied in this paper is not necessarily convex, hence we leave the study of this topic in the future research. And we use the standard Delaunay triangulation in this work [56].

Sequential Space Partition

Based on the aforementioned strategies, the scalability of the learning problem (16) can still be improved by partitioning the region of interest \mathcal{X} into a sequence of subset, such that

$$\mathcal{X}_s \subset \mathcal{X}_1 \subset \mathcal{X}_2 \cdots \subset \mathcal{X}.$$

Algorithm 1

Input: RoA prior \mathcal{X}_s , negativity tolerance $-\epsilon$, Lipschitz overestimate M , set level α ,
Output: Lyapunov function $L(x)$
Refine a tessellation $\{C_k\}_{k=1}^{N_c}$ until it satisfies (17)
if tessellation is valid **then**
 Solve optimization (18)
 if Optimal value solution satisfies $s_{\alpha,\epsilon}^* = -\epsilon \sum_{k=1}^{N_c} |C_k|$ **then**
 Return
 end if
else
 Return cannot learn $L(x)$.
end if

Note that the logic behind the proposed algorithm is an augmentation of the a-priori knowledge in \mathcal{X}_s to \mathcal{X} , this allows us to further improve the computational efficiency. The key idea is to gradually augment the volume of the RoA, and in each iteration, it is only necessary to learn the LF on the set $\mathcal{X}_i \setminus \mathcal{X}_{i-1}$. This sequential learning algorithm is summarized in Algorithm 2. It is noteworthy that, if one needs to recover the whole LF on $\mathcal{X} \setminus \mathcal{X}_s$, then it is necessary to impose the continuity condition on the boundary of $\partial\mathcal{X}_i$ between the i -th iteration and the $i+1$ -th iteration. The corresponding algorithm is summarized in Algorithm 2.

Algorithm 2

Input: Subset sequence $\{\mathcal{X}_i\}$ and the initial basin $\mathcal{X}_0 = \mathcal{X}_s$
Output: Lyapunov function $L(x)$
 $i = 1$
while $\{\mathcal{X}_s \neq \mathcal{X}\}$ **do**
 Run Algorithm. 1 with \mathcal{X}_s
 if Algorithm 1 failed **then**
 Return cannot learn $L(x)$.
 end if
 $i \leftarrow i + 1$
 $\mathcal{X}_s \leftarrow \mathcal{X}_i$
end while

5.4 Learning \mathcal{X}

Up to this point, we assume knowledge of \mathcal{X} within which the stability analysis is conducted. However, such prior knowledge/assumption is not necessarily available. Instead, the users may only have access to a set of data and hope to find out a ROA based on this dataset. In this case,

we will need to solve the Learning problem (16) without the boundary condition:

$$\begin{aligned}
s_\epsilon^* &:= \min_{\{g_k, b_k\}, \{\tilde{g}_{i,k}\}, \{s_{j,k}\}} \sum_{k=1}^{N_C} \sum_{j=1}^{|C_k|} s_{j,k} \\
\forall k, \ell \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, & \quad s_{j,k} \geq -\epsilon \\
v_{j,k} \in C_\ell & \implies (g_k - g_\ell)^\top v_{j,k} = b_\ell - b_k \\
v_{j,k} \in \mathcal{X} \setminus \partial\mathcal{X} & \implies g_k^\top v_{j,k} + b_k \leq \alpha - \epsilon \\
& \quad \sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k \\
& \quad \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \|v_{j,k} - x_i\| \leq s_{j,k}
\end{aligned} \tag{20}$$

Based on the solution to this problem, it is possible to determine a ROA by the following Corollary:

Corollary 10. *If the solution to problem (20) satisfies $s_\epsilon^* = -\epsilon \sum_{k=1}^{N_C} |C_k|$, then any sublevel set $L_{\leq \alpha} \subset \mathcal{X}$ with $\alpha \in \mathbb{R}$ is attracted to 0 under the unknown dynamics f .*

Proof. If the solution to (20) satisfies $s_\epsilon^* = -\epsilon \sum_{k=1}^{N_C} |C_k|$, by choosing $\alpha \in \mathbb{R}$ such that $L_{\leq \alpha} \subset \mathcal{X}$, then the optimal solution to (20) is also an optimal solution to:

$$\begin{aligned}
& \min_{\{g_k, b_k\}, \{\tilde{g}_{i,k}\}, \{s_{j,k}\}} \sum_{k=1}^{N_C} \sum_{j=1}^{|C_k|} s_{j,k} \\
\forall k, \ell \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, & \quad s_{j,k} \geq -\epsilon \\
v_{j,k} \in C_\ell \cap L_{\leq \alpha} & \implies (g_k - g_\ell)^\top v_{j,k} = b_\ell - b_k \\
v_{j,k} \in L_{\leq \alpha} \setminus \partial L_{\leq \alpha} & \implies g_k^\top v_{j,k} + b_k \leq \alpha - \epsilon \\
v_{j,k} \in \partial L_{\leq \alpha} & \implies g_k^\top v_{j,k} + b_k = \alpha \\
& \quad \sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k \\
& \quad \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \|v_{j,k} - x_i\| \leq s_{j,k},
\end{aligned}$$

whose tessellation is given by $\{C_k \cap L_{\leq \alpha}\}_k^{N_C}$. Hence, Theorem 6 holds in $L_{\leq \alpha}$, which summarizes the proof. \square

Based on this corollary, we are able to learn the ROA within \mathcal{X} by applying the algorithm summarized in Algorithm 3.

6 Numerical Results

In this part, we going to evaluate the proposed learning schemes in two different examples. In particular, we will make use of Algorithm 1, 2 and 3 in the first example and use Algorithm 3 in the second one. All the following results are implemented on a laptop with Intel i7-11800H and 32G memory, and the `Mosek` is used for solving the SOCP problem.

Algorithm 3

Input: RoA prior \mathcal{X}_s , negativity tolerance $-\epsilon$, Lipschitz overestimate M ,

Output: Lyapunov function $L(x)$, ROA $L_{\leq\alpha}$

Refine a tessellation $\{C_k\}_{k=1}^{N_c}$ until it satisfies (17)

if tessellation is valid **then**

 Solve optimization (20)

if Optimal value solution satisfies $s_\epsilon^* = -\epsilon \sum_{k=1}^{N_c} |C_k|$ **then**

 find $\alpha \in \mathbb{R}$ such that $L_{\leq\alpha} \subset \mathcal{X}$

 Return

end if

else

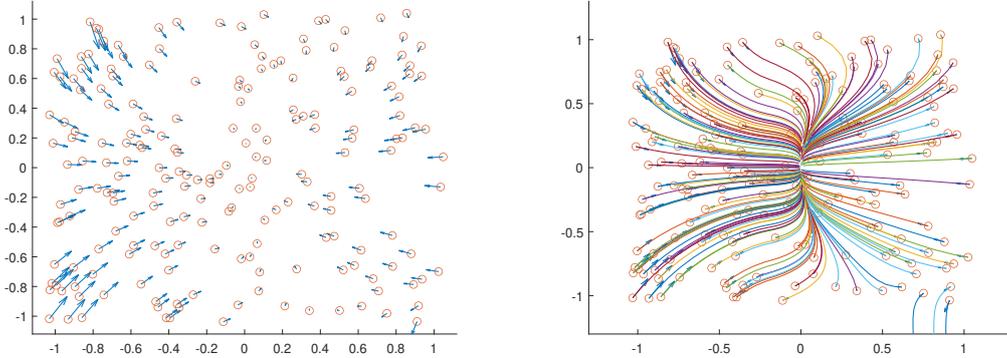
 Return cannot learn $L(x)$.

end if

6.1 Non Polynomial dynamic system

We consider a two-dimensional nonlinear dynamic system:

$$\begin{aligned}\dot{x}_1(t) &= -0.9 \sin(x_1(t)) \cos(x_2(t)) + 0.2x_1(t)x_2(t) + 0.25x_2(t)^2 \\ \dot{x}_2(t) &= -1 \sin(x_2(t))(|x_1(t) + 0.2|) + 0.5 \frac{x_1(t)x_2(t)}{\cos(x_2(t)) - 0.3x_1(t)}\end{aligned}$$



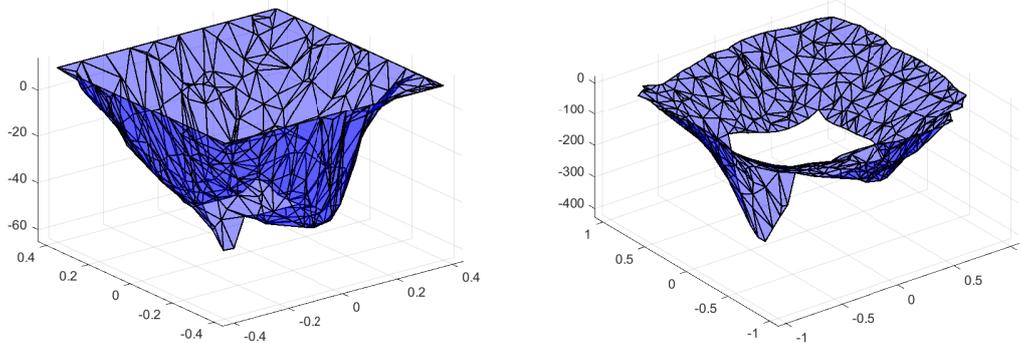
(a) The data used for the learning scheme.

(b) Underlying dynamics (not used for learning).

Figure 3: Representations of the considered nonpolynomial dynamic system.

We assume that we know a ROA $\mathcal{X}_s = [-0.1, 0.1]^2$. A dataset with only 200 samples within $[-1, 1]^2 \subset \mathbb{R}^2$ is used to learn the underlying Lyapunov function: the positions $\{x_i\}$ and speeds $\{f(x_i)\}$ of these samples are plotted in Figure 3a, from which we can observe that this dataset is relatively sparse in $[-1, 1]^2$. Judging by the speed sample, the dynamic system seem stable within the box $[-0.4, 0.4]^2$, while stability within the region $[-1, 1]^2 \setminus [-0.4, 0.4]^2$ is unclear because of the speed samples in the lower right corner in Figure 3a. Hence, we ran sequential space partition

scheme (Algorithm 2). In particular, we first use Algorithm 1 in the region $[-0.4, 0.4]^2$ with $\mathcal{X}_s = [-0.1, 0.1]^2$. After we justify that $[-0.4, 0.4]^2$ is a positively invariant subset of the ROA, then we further apply Algorithm 3 to $[-1, 1]^2$ with $\mathcal{X}_s = [-0.4, 0.4]^2$. In both sub-problems, the negativity tolerances ϵ are set to 10^{-3} and the tessellation are both randomly generated by Delaunay triangulation [56].



(a) $\mathcal{X}_s = [-0.1, 0.1]^2 \subset \mathcal{X} = [-0.4, 0.4]^2$.

(b) Learning a ROA estimate within $[-1, 1]^2$.

Figure 4: Visualization of the learnt Lyapunov functions.

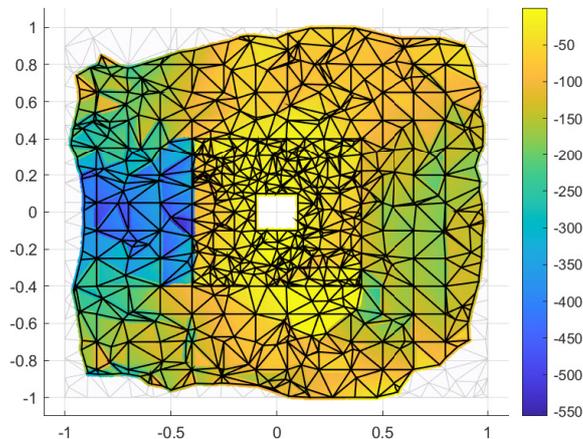


Figure 5: Evaluation of $f(x)^\top \partial_{C1} L(x)$ on the learnt ROA, the gray triangulation in the background is the tessellation used to solve Problem (20), while the coloured region in the front is our ROA estimate.

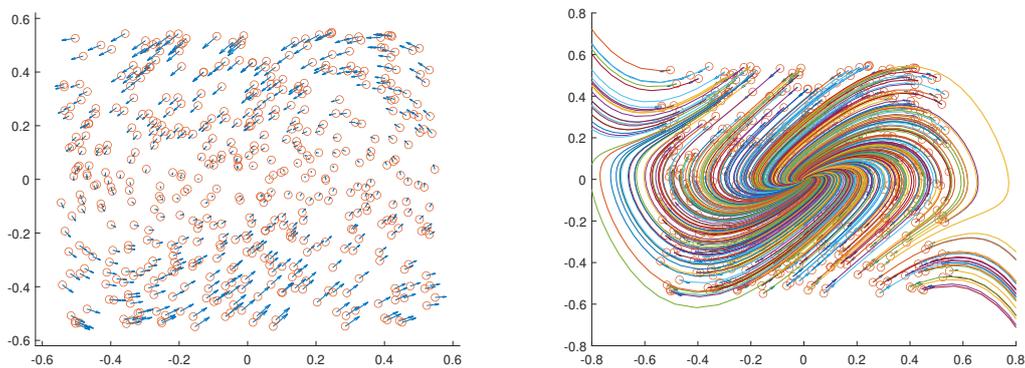
The learnt Lyapunov function in $[-0.4, 0.4]^2$ is shown in Figure 4a, while the ROA we finally end up with is shown in Figure 5. Moreover, the Lyapunov function learnt from Algorithm 3 in $\mathcal{X}_s = [-1, 1]^2 \setminus [-0.4, 0.4]^2$ is shown in Figure 4b. Figure 5 also shows the evaluation of $f(x)^\top \partial_{C1} L(x)$ with respect to the underlying dynamic system, whose maximal evaluation is $-1.525 \times 10^{-2} < 0$, as expected from a Lyapunov function. In accordance with our guess, the

learnt ROA in Figure 5 cuts off the lower right corner, because this region does not seem to be stable. To see that, we simulate the underlying dynamic system by setting the initial states to points in our dataset. The simulated trajectories are plotted in Figure 3b; please note that these trajectories are not used in the learning scheme at all.

6.2 Reverse Time Van Del Pol Oscillator

In this part, we consider the reverse time Van Del Pol oscillator:

$$\begin{aligned}\dot{x}_1(t) &= -2x_2(t) \\ \dot{x}_2(t) &= -0.8x_1(t) - 10(x_1(t)^2 - 0.21)x_2(t).\end{aligned}$$



(a) The data used for the learning scheme. (b) Simulated trajectories (not used for learning).

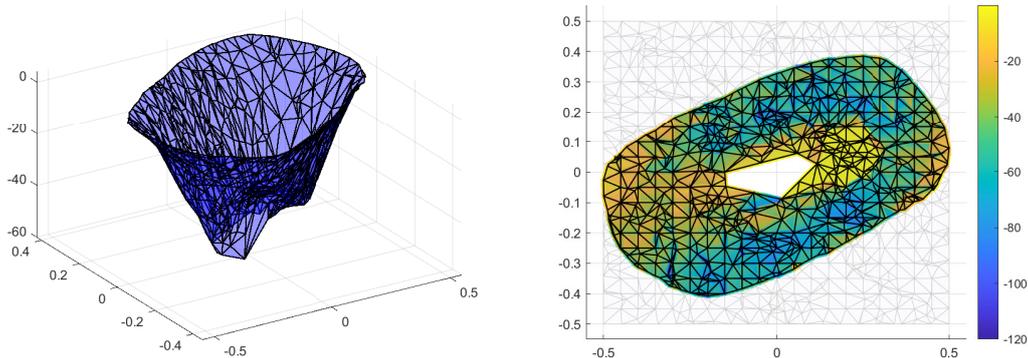
Figure 6: The reverse time Van Del Pol oscillator.

We know an a-priori polytopic ROA \mathcal{X}_s , which is plotted in the center of Figure 7b. A dataset with only 400 samples within $[-0.5, 0.5]^2 \subset \mathbb{R}^2$ is used to learn the underlying Lyapunov function: the positions $\{x_i\}$ and speeds $\{f(x_i)\}$ of these samples are plotted in Figure 6a. Similar to what we did in the last example, we simulate these data forward in Figure 6b, while these trajectories are not used in the learning scheme. We can observe that both the lower right corner and the upper left corner in Figure 6b correspond to regions of unstable states. Even with only the access to the data in Figure 6a, we can not give a clear idea about which region is safe, hence we apply Algorithm 3 to $[-0.5, 0.5]^2$. In particular, the negativity tolerance ϵ is set to 10^{-3} and the tessellation is randomly generated by Delaunay triangulation [56]. The learnt Lyapunov function and its evaluation of $f(x)^\top \partial_{C_1} L(x)$ on the learnt ROA are respectively plotted in Figure 7a and Figure 7b. In particular, the maximal evaluation of $f(x)^\top \partial_{C_1} L(x)$ on the learnt ROA is $-1.947 \times 10^{-2} < 0$.

7 Conclusion

Our results

In this work, we went all the way from proving a variant of stability theorem with non-smooth Lyapunov functions (LF), to actually implementing an algorithm for data-based region of attraction (RoA) estimation with unknown dynamic system. In the process, we went through proving a



(a) Visualization of the Lyapunov function. (b) Evaluation of $f(x)^\top \partial_{C1} L(x)$ on the learnt ROA.

Figure 7: The Lyapunov function learnt from a Van der Pol dataset.

On Figure 7b, the gray triangularization in the background is the tessellation used to solve Problem (20), while the coloured region in the front is ROA. The polytopic hole in the middle is \mathcal{X}_s

theorem for piecewise affine (PWA) LF computation and deriving a convex optimization program for computing such LF.

The originality of the method we propose is that it only requires a fixed dataset to compute an estimate of the RoA, from which it allows the user to deduce global information from local data and knowledge on the Lipschitz constant of the dynamic system. Hence, it can be used to study systems whose dynamic system cannot be easily sampled at will, through a relatively simple optimization problem that can be handled with interior point methods.

Future works

In this work, we proposed a learning scheme that can learn a robust Lyapunov function from a fixed dataset. However, with such minimal knowledge of the underlying dynamic system comes a big convex optimization even for a low-dimensional dynamic system. In future works, we plan on investigating how side information, particularly the RKHS structure [57], or other a-priori knowledge of the underlying dynamic system can be incorporated into the learning scheme, so that it can handle a higher-dimensional dynamic system.

References

- [1] A. M. Lyapunov, “The general problem of the stability of motion,” *International journal of control*, vol. 55, no. 3, pp. 531–534, 1992.
- [2] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, “Learning an approximate model predictive controller with guarantees,” *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, 2018.
- [3] S. Gao, S. Kong, and E. M. Clarke, “dreal: An smt solver for nonlinear theories over the reals,” in *International conference on automated deduction*, pp. 208–214, Springer, 2013.

- [4] Y.-C. Chang, N. Roohi, and S. Gao, “Neural lyapunov control,” *Advances in neural information processing systems*, vol. 32, 2019.
- [5] H. Dai, B. Landry, M. Pavone, and R. Tedrake, “Counter-example guided synthesis of neural network lyapunov functions for piecewise linear systems,” in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 1274–1281, IEEE, 2020.
- [6] J. Kapinski, J. V. Deshmukh, S. Sankaranarayanan, and N. Aréchiga, “Simulation-guided lyapunov analysis for hybrid dynamical systems,” in *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pp. 133–142, 2014.
- [7] S. Gao, J. Avigad, and E. M. Clarke, “ δ -complete decision procedures for satisfiability over the reals,” in *International Joint Conference on Automated Reasoning*, pp. 286–300, Springer, 2012.
- [8] V. I. Zubov, *Methods of AM Lyapunov and their application*. P. Noordhoff, 1964.
- [9] P. Giesl, *Construction of global Lyapunov functions using radial basis functions*, vol. 1904. Springer, 2007.
- [10] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.
- [11] A. Oustry, M. Tacchi, and D. Henrion, “Inner approximations of the maximal positively invariant set for polynomial dynamical systems,” *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 733–738, 2019.
- [12] D. Henrion and M. Korda, “Convex computation of the region of attraction of polynomial control systems,” *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 297–312, 2013.
- [13] M. Tacchi, C. Cardozo, D. Henrion, and J. B. Lasserre, “Approximating regions of attraction of a sparse polynomial differential system,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3266–3271, 2020.
- [14] A. A. Ahmadi and A. Majumdar, “Dsos and sdsos optimization: more tractable alternatives to sum of squares and semidefinite optimization,” *SIAM Journal on Applied Algebra and Geometry*, vol. 3, no. 2, pp. 193–230, 2019.
- [15] Z. Sun, “Stability of piecewise linear systems revisited,” *Annual Reviews in Control*, vol. 34, no. 2, pp. 221–231, 2010.
- [16] H. Lin and P. J. Antsaklis, “Stability and stabilizability of switched linear systems: a survey of recent results,” *IEEE Transactions on Automatic control*, vol. 54, no. 2, pp. 308–322, 2009.
- [17] A. Alessio and A. Bemporad, “A survey on explicit model predictive control,” in *Nonlinear model predictive control*, pp. 345–369, Springer, 2009.
- [18] M. Johansson and A. Rantzer, “Computation of piecewise quadratic lyapunov functions for hybrid systems,” in *1997 European Control Conference (ECC)*, pp. 2005–2010, IEEE, 1997.
- [19] H. Ravanbakhsh and S. Sankaranarayanan, “Learning control lyapunov functions from counterexamples and demonstrations,” *Autonomous Robots*, vol. 43, no. 2, pp. 275–307, 2019.

- [20] R. Schwan, C. N. Jones, and D. Kuhn, “Stability verification of neural network controllers using mixed-integer programming,” *arXiv preprint arXiv:2206.13374*, 2022.
- [21] A. Solar-Lezama, L. Tancau, R. Bodik, S. Seshia, and V. Saraswat, “Combinatorial sketching for finite programs,” in *Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*, pp. 404–415, 2006.
- [22] A. Solar-Lezama, *Program synthesis by sketching*. University of California, Berkeley, 2008.
- [23] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado, “Learning lyapunov functions for hybrid systems,” in *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, pp. 1–11, 2021.
- [24] A. Abate, D. Ahmed, M. Giacobbe, and A. Peruffo, “Formal synthesis of lyapunov neural networks,” *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 773–778, 2020.
- [25] T. A. Johansen, “Computation of lyapunov functions for smooth nonlinear systems using convex optimization,” *Automatica*, vol. 36, no. 11, pp. 1617–1626, 2000.
- [26] P. Giesl, B. Hamzi, M. Rasmussen, and K. N. Webster, “Approximation of lyapunov functions from noisy data,” *arXiv preprint arXiv:1601.01568*, 2016.
- [27] S. F. Marinósson, “Lyapunov function construction for ordinary differential equations with linear programming,” *Dynamical Systems: An International Journal*, vol. 17, no. 2, pp. 137–150, 2002.
- [28] P. M. Julian, *A high-level canonical piecewise linear representation: Theory and applications*. PhD thesis, Universidad Nacional del Sur (Argentina), 1999.
- [29] R. Baier, L. Grüne, and S. F. Hafstein, “Linear programming based lyapunov function computation for differential inclusions,” *Discrete & Continuous Dynamical Systems-B*, vol. 17, no. 1, p. 33, 2012.
- [30] S. F. Hafstein, “An algorithm for constructing lyapunov functions,” *Electronic Journal of Differential Equations*, vol. 2007, 2007.
- [31] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317. Springer Science & Business Media, 2009.
- [32] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of convex analysis*. Springer Science & Business Media, 2004.
- [33] F. H. Clarke, *Optimization and Nonsmooth Analysis*. Classics in Applied Mathematics, SIAM, 1990.
- [34] J. P. La Salle, *The stability of dynamical systems*. SIAM, 1976.
- [35] M. Della Rossa, *Non-Smooth Lyapunov Functions for Stability Analysis of Hybrid Systems*. PhD thesis, Institut National des Sciences Appliquées de Toulouse, 2020.
- [36] H. L. Royden and P. Fitzpatrick, *Real analysis*, vol. 32. Macmillan New York, 1988.
- [37] F. Aurenhammer, “A criterion for the affine equivalence of cell complexes in \mathbb{R}^d and convex polyhedra in \mathbb{R}^{d+1} ,” *Discrete & Computational Geometry*, vol. 2, no. 1, pp. 49–64, 1987.

- [38] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [39] A. Ben-Tal and A. Nemirovski, “Robust optimization—methodology and applications,” *Mathematical programming*, vol. 92, no. 3, pp. 453–480, 2002.
- [40] J.-P. Calliess, S. J. Roberts, C. E. Rasmussen, and J. Maciejowski, “Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control,” *Automatica*, vol. 122, p. 109216, 2020.
- [41] S. Saitoh and Y. Sawano, *Theory of reproducing kernels and applications*. Springer, 2016.
- [42] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.
- [43] P. Scharnhorst, E. T. Maddalena, Y. Jiang, and C. N. Jones, “Robust uncertainty bounds in reproducing kernel hilbert spaces: A convex optimization approach,” *arXiv preprint arXiv:2104.09582*, 2021.
- [44] L. Sabug Jr, F. Ruiz, and L. Fagiano, “Smgo: A set membership approach to data-driven global optimization,” *Automatica*, vol. 133, p. 109890, 2021.
- [45] M. Canale, L. Fagiano, and M. Milanese, “Set membership approximation theory for fast implementation of model predictive control laws,” *Automatica*, vol. 45, no. 1, pp. 45–54, 2009.
- [46] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.
- [47] S. F. Hafstein*, “A constructive converse lyapunov theorem on asymptotic stability for nonlinear autonomous ordinary differential equations,” *Dynamical Systems*, vol. 20, no. 3, pp. 281–299, 2005.
- [48] P. Giesl and S. Hafstein, “Review on computational methods for lyapunov functions,” *Discrete & Continuous Dynamical Systems-B*, vol. 20, no. 8, p. 2291, 2015.
- [49] A. Subbaraman and A. R. Teel, “A matrosov theorem for strong global recurrence,” *Automatica*, vol. 49, no. 11, pp. 3390–3395, 2013.
- [50] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.
- [51] H. Krieg, *Modeling and solution of continuous set covering problems by means of semi-infinite optimization*. Fraunhofer Verlag, 2019.
- [52] E. D. Andersen, C. Roos, and T. Terlaky, “On implementing a primal-dual interior-point method for conic quadratic optimization,” *Mathematical Programming*, vol. 95, no. 2, pp. 249–277, 2003.
- [53] L. Vandenberghe and M. S. Andersen, “Chordal graphs and semidefinite optimization,” *Foundations and Trends in Optimization*, vol. 1, no. 4, pp. 241–433, 2015.
- [54] K. Rybnikov, *Polyhedral partitions and stresses*. Queen’s University at Kingston, 2000.
- [55] F. Aurenhammer, “Power diagrams: properties, algorithms and applications,” *SIAM Journal on Computing*, vol. 16, no. 1, pp. 78–96, 1987.

- [56] B. Delaunay *et al.*, “Sur la sphere vide,” *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, vol. 7, no. 793-800, pp. 1–2, 1934.
- [57] P.-C. Aubin-Frankowski and Z. Szabó, “Hard shape-constrained kernel machines,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 384–395, 2020.