Proceedings of Machine Learning Research vol 288:1–13, 2025 2nd International Conference on Neuro-symbolic Systems (NeuS)

Formal Synthesis of Lyapunov Stability Certificates for Linear Switched Systems using ReLU Neural Networks

Virginie Debauche[1] VIRGINIE.DEBAUCHE@CS.OX.AC.UK
Alec Edwards[1,3] ALEC.EDWARDS@PHAIDRA.AI
Raphael M. Jungers[2] RAPHAEL.JUNGERS@UCLOUVAIN.BE
Alessandro Abate[1] ALESSANDRO.ABATE@CS.OX.AC.UK
[1] *Department of Computer Science, Oxford University, Oxford, United Kingdom*
[2] *ICTEAM, UCLouvain, Louvain-la-Neuve, Belgium*
[3] *Phaidra Inc*

Editors: G. Pappas, P. Ravikumar, S. A. Seshia

Abstract

This paper presents a neural network-based algorithm with soundness guarantees to study the stability of discrete-time linear switched systems. This algorithm follows a counterexample guided inductive synthesis (CEGIS) architecture: an iterative process alternating between the *learner*, which provides a candidate Lyapunov function, and the *verifier* which checks its validity over the whole domain. We choose a ReLU neural network as learner for its expressivity and flexibility, and a satisfiability module theories (SMT) solver as verifier. In addition, we introduce a post processing step to leverage a valid Lyapunov function from the neural network in case of failure of the CEGIS loop. Several examples demonstrate the algorithm's efficacy.
Keywords: Lyapunov function, Joint spectral radius, Neural network, CEGIS, SMT

1. Introduction

Switched systems provide a simple yet powerful modeling framework to capture many processes and approximate complex systems such as cyber-physical systems. They present both practical and theoretical challenges, including their stability analysis which has been the core subject of many works; see Sun and Ge (2011). Formally, a *discrete-time linear switched system* is of the form

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

where the *switching signal* $\sigma : \mathbb{N} \to \{1, \dots, M\}$ sets the dynamics at each iteration and $A_{\sigma(k)}$ lies in a finite set of square matrices $\Sigma := \{A_1, \dots, A_M\} \subset \mathbb{R}^{n\times n}$ with $n, M \in \mathbb{N}$.

The stability of these systems has been extensively studied. In this work, we focus on the case of *arbitrary switching*, i.e. when there are no constraints on the switching signal. It turns out that in this case stability is characterized by the *Joint Spectral Radius* (JSR) of Σ. First introduced in Rota and Strang (1960), the JSR of a finite set of matrices Σ, denoted by $\rho(\Sigma)$, is defined by

$$\rho(\Sigma) \;:=\; \limsup_{k\to\infty} \left\{ \|A\|^{1/k} : A \in \Sigma^k \right\}, \tag{2}$$

where Σ^k encodes all the possible products of length k of the matrices in Σ. It is well-known that system (1) is asymptotically stable if and only if $\rho(\Sigma) < 1$, see (Jungers, 2009, Corollary 1.1). Despite this appealing characterization, the approximation of the JSR was proven to be NP-hard (Blondel and Tsitsiklis (1997)) and the question of deciding whether "$\rho(\Sigma) < 1$?" is even undecidable (Blondel and Canterini (2008)). This has not, however, prevented the emergence of several approximation techniques (see (Jungers, 2009, Chapter 2.3), for a whole review); Most of which derive from Lyapunov theory, e.g. quadratic functions in Blondel et al. (2005) and Sum-Of-Squares (SOS) polynomials in Parrilo and Jadbabaie (2008), and aim to provide JSR upper bounds.

Recently, several Machine Learning (ML) techniques have tackled the problem of learning a Lyapunov function; see Petridis and Petridis (2006); Richards et al. (2018); Chang et al. (2020); Dawson et al. (2021); Farsi et al. (2022); Zhang et al. (2023); Lechner et al. (2022); Chen et al. (2021). In particular, Abate et al. (2021) proposes a CounterExample-Guided Inductive Synthesis (CEGIS) algorithm where a neural network is trained to represent a Lyapunov function whose validity is soundly checked by a Satisfiability Modulo Theories (SMT) solver thereafter. This approach benefits from the flexibility and expressiveness of neural networks and has shown promising results, though it does not address switched systems. In this paper, we adopt a similar strategy to approximate the JSR. Not only does the neural network have to represent a candidate Lyapunov function, but we must also rely on sample points to provide the best (tightest) upper approximation of the JSR. Therefore, the SMT solver checks the Lyapunov inequalities over the whole domain. The neural network and the SMT solver alternate until a valid approximation is found, or the procedure stops after reaching a maximum number of iterations. In case of failure, we introduce a post processing step; It leverages the network's knowledge of the sample points to derive a valid JSR approximation despite the failure. This contribution addresses a key limitation of many CEGIS-based stability analyses: the lack of guarantees that the procedure will yield a valid JSR bound, which we mitigate through a tailored post-processing step. Finally, we test our new algorithm on several benchmarks. An additional contribution is a well-defined, well-understood benchmark for evaluating neural networks in Lyapunov analysis, which is an emerging trend in AI and automation.

The paper is organised as follows: Section 2 reviews classical JSR approximation techniques and the expressivity power of ReLU neural networks. Section 3 introduces our CEGIS JSR approximation ML technique. Section 3.1 improves previous result linking approximation guarantees to ReLU neural network structure. Section 3.2 presents the algorithm's components and Section 3.3 introduces a post processed approximation method. We conclude with numerical experiments and further discussion in Section 4, before the conclusion.

Notation : Given a square matrix $Q \in \mathbb{R}^{n\times n}$ of dimension $n \in \mathbb{N}$, $Q \succ 0$ denotes that Q is *positive definite*, that is $x^\top Qx > 0$ for any $x \in \mathbb{R}^n_0 := \mathbb{R}^n \setminus \{0\}$. Given a real number $x \in \mathbb{R}$, $\lceil x \rceil$ denotes the *ceiling* of x, i.e. the smallest integer exceeding x.

2. Preliminaries

2.1. Classical and newer approximation techniques

Notwithstanding several theoretical limitations, the approximation of the JSR has been the subject of numerous studies. Most of them rely on the following characterisation and amount to finding a common contractive norm along all the matrices. In this section, we first review the approximation techniques which derive from the following result.

Theorem 1 (Proposition 1 in Rota and Strang (1960)) *For any finite set of matrices Σ such that $\rho(\Sigma) \neq 0$, the joint spectral radius of Σ can be defined as*

$$\rho(\Sigma) \;:=\; \inf_{\|\cdot\|\in\mathcal{N}} \max_{A\in\Sigma} \|A\|\,, \tag{3}$$

where $\mathcal{N}$ refers to the family of submultiplicative norms[1].

[1] A matrix norm $\|\cdot\| : \mathbb{R}^{n\times n} \to \mathbb{R}_{\geq 0}$ is *submultiplicative* if for all $A, B \in \mathbb{R}^{n\times n}$, $\|AB\| \leq \|A\|\,\|B\|$.

Note that one can restrict the infimum over the matrix norms induced by a vector norm. Theorem 1 implies in particular that any submultiplicative norm $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ provides an upper bound on the JSR, defined as the maximum norm over the matrices in Σ denoted by $\rho(\Sigma, \|\cdot\|)$, i.e.

$$\rho(\Sigma) \;\leq\; \max_{A\in\Sigma} \|A\| \;:=\; \rho(\Sigma, \|\cdot\|).$$

Moreover, if there exists a norm which realizes the infimum in Theorem 1 (which is not always the case), it will be called an *extremal norm*. For any $\varepsilon > 0$, there exists a norm $\|\cdot\|_\varepsilon$, which will be called ε-extremal, that satisfies

$$\forall A \in \Sigma, \forall x \in \mathbb{R}^n : \; \|Ax\|_\varepsilon \;\leq\; (\rho(\Sigma) + \varepsilon)\, \|x\|. \tag{4}$$

In order to provide an upper bound on the JSR that would be as tight as possible, one usually looks over a sufficiently wide set of norms, called a *template* for which the computation can be easily done. The most classical family of norms is the set of *ellipsoidal* norms defined by $\|x\|_Q := \sqrt{x^\top Q x}$ where $Q = Q^\top$ and $Q \succ 0$. Theorem 14 in Blondel et al. (2005) proves that the best ellipsoidal approximation of the JSR, i.e. $\rho_{\mathcal{Q}}(\Sigma) := \inf_{Q\succ 0}\; \max_{A\in\Sigma}\; \|A\|_Q$, satisfies some guarantees, namely that

$$\frac{1}{\tau_{\mathcal{Q}}}\, \rho_{\mathcal{Q}}(\Sigma) \;\leq\; \rho(\Sigma) \;\leq\; \rho_{\mathcal{Q}}(\Sigma), \tag{5}$$

where $\tau_{\mathcal{Q}} := \sqrt{n}$. Moreover, they proved that $\rho_{\mathcal{Q}}(\Sigma)$ can be computed efficiently as the solution of a convex optimisation program.

Not only norms may be used to derive an upper bound on the JSR. One can indeed relax the convexity property, and look for a *candidate (common) Lyapunov function*, i.e. a continuous, positive definitive and homogeneous function (Ahmadi et al., 2014, Theorem 2.4). If such a function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ satisfies the following *Lyapunov inequalities*:

$$\forall A \in \Sigma,\; \forall x \in \mathbb{R}^n : \; V(Ax) \;\leq\; \gamma\, V(x), \tag{6}$$

then $\rho(\Sigma) \leq \gamma$. For example, Parrilo and Jadbabaie (2008) introduce the JSR approximation using *Sum-Of-Squares* (SOS) polynomials (which are not necessarily norms) for which they also provide a bound on the quality of the approximation (see (Parrilo and Jadbabaie, 2008, Theorem 3.4)).

In this work, we consider the universal template of *continuous piecewise linear (CPWL) functions* as candidate Lyapunov functions. Since their sublevel sets are polytopes, we will refer to the *polytopic approximation* of the JSR. Similarly to John's Theorem (John (1948)) for the ellipsoidal approximation of convex sets, the following theorem provides an upper bound on the minimum number of vertices for a polytope to approximate a convex set with some prespecified error τ.

Theorem 2 (Theorem 1.1 in Barvinok (2013)) *Let n and k be two positive integers and $\tau > 1$ be a real number such that*

$$\left(\tau - \sqrt{\tau^2 - 1}\right)^k + \;\left(\tau + \sqrt{\tau^2 - 1}\right)^k \;\geq\; 6\, D(n,k)^{1/2}, \tag{7}$$

where

$$D(n,k) \;:=\; \sum_{m=0}^{\lfloor k/2 \rfloor} \binom{n + k - 1 - 2m}{k - 2m}. \tag{8}$$

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

$$P \;\subseteq\; K \;\subseteq\; \tau P. \tag{9}$$

Using this theorem, Debauche et al. (2024) derive some approximation guarantees using CPWL functions whose polytopic sublevel sets have a fixed number of vertices, as follows.

Theorem 3 (Theorem 6 in Debauche et al. (2024)) *Let $\rho(\Sigma)$ be the joint spectral radius of a finite set of matrices Σ of dimension $n \in \mathbb{N}$. For any $\tau > 1$ and $k_\tau \in \mathbb{N}$ such that relation (7) is satisfied, the following relation holds:*

$$\frac{1}{\tau}\rho_{\mathcal{P}}(\Sigma) \leq \rho(\Sigma) \leq \rho_{\mathcal{P}}(\Sigma), \tag{10}$$

where $\rho_{\mathcal{P}}$ is the optimum solution of (3) where the norms are restricted to CPWL norms with at most $8D(n, k_\tau)$ vertices.

2.2. Expressivity power of ReLU neural networks

In this work, we use a feedforward neural network with ReLU activation functions as template.

Definition 1 (ReLU neural network) *A* Rectified Linear Units (ReLU) feedforward neural network *with $k \in \mathbb{N}$ hidden layers with parameter $\theta = [n_0, \dots, n_{k+1}]$ is defined by k affine transformations $T^{(j)} : \mathbb{R}^{n_{j-1}} \to \mathbb{R}^{n_j}$, $x \mapsto W^{(j)}x + b^{(j)}$ for $j \in \{1, \dots, k\}$, and a linear transformation $T^{(k+1)} : \mathbb{R}^{n_k} \to \mathbb{R}^{n_{k+1}}$, $x \mapsto W^{(k+1)}x$. The network* represents *the function $NN_\theta : \mathbb{R}^{n_0} \to \mathbb{R}^{n_{k+1}}$ given by*

$$NN_\theta := T^{(k+1)} \circ \sigma \circ \cdots \circ T^{(2)} \circ \sigma \circ T^{(1)}, \tag{11}$$

where $\sigma(x) = (\max\{0, x_1\}, \dots, \max\{0, x_n\})$. The matrices $W^{(l)}$ and the vectors $b^{(l)}$ are respectively the weights *and the* biases *of the l-th layer while n_l is the* width *of the l-th layer. The maximum width of all the hidden layers is called the* width *of the neural network, and the* depth *is $k + 1$.*

Any function represented by a ReLU neural network is a continuous piecewise affine function.

Definition 2 (Continuous piecewise affine/linear function) *We say that a function $f : \mathbb{R}^n \to \mathbb{R}$ is a* continuous piecewise affine (resp. linear) function *(CPWA/L function) if there exists a finite set of polyhedra[2] whose union is $\mathbb{R}^n$, and furthermore if f is affine (resp. linear) over each polyhedron. The* number of pieces *of f is the mininum number of polyhedra necessary to express f as above.*

Conversely, any CPWL function can be represented by a ReLU neural network, and it is possible to bound the depth and the width with respect to its number of pieces. Moreover, the represented function is homogeneous if and only if the biases are zero, see (Hertrich et al., 2021, Proposition 2.3).

Theorem 4 (Theorems 4.2 and 4.4 in Hertrich et al. (2021)) *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a CPWL function with p affine pieces. Then f can be represented by a ReLU neural network with depth $\lceil \log_2(n+1) \rceil + 1$ and width $\mathcal{O}(p^{2n^2+3n+1})$. If f is a convex CPWL function, then the width is of $\mathcal{O}(p^{n+1})$.*

Theorem 4 means that, given a CPWL function f, there exists a ReLU neural network, some weight matrices and some biases such that the represented function is f. Nothing ensures that one will reach this function during the training process though.

3. Formal approximation of the JSR using ReLU neural networks

In this section, we introduce a new formal method to approximate the JSR of a finite set of matrices using ML algorithms. We consider a CEGIS architecture as implemented in the Fossil tool developed in Abate et al. (2021); Edwards et al. (2023, 2024). This synthesis architecture involves the alternating interaction of two main components, namely a *learner* and a *verifier*. The learner

[2] A *polyhedron* is the intersection of a finite number of halfspaces. A *polytope* is a bounded polyhedron.

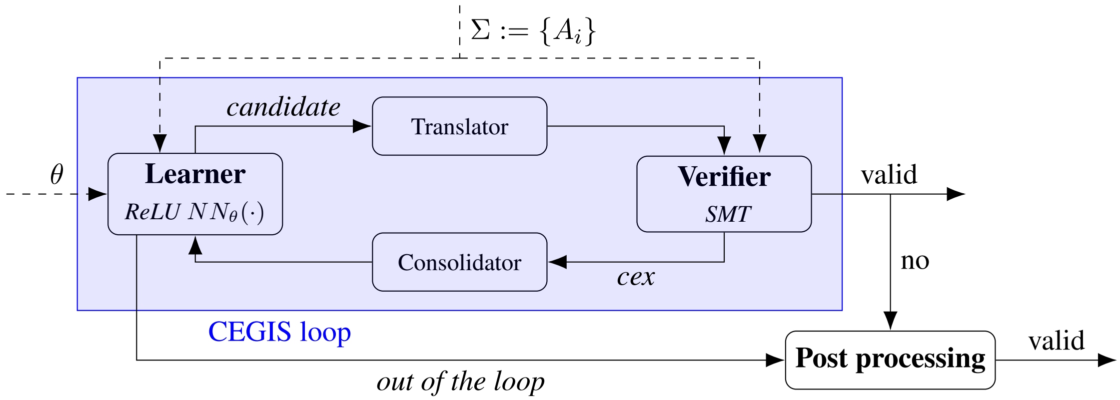


Figure 1: Illustration of the CEGIS architecture of our method to provide sound upper approximation of the JSR of a finite set of matrices $\Sigma \subset \mathbb{R}^{n\times n}$.

seeks to submit a candidate Lyapunov function and a sample-based JSR approximation to the verifier. Subsequently, the verifier either validates or disproves the candidate stability certificate – in the latter case providing new samples to the learner. The communication between the two components is facilitated by the *translator* and the *consolidator*; further information can be found in Abate et al. (2021). Although the algorithm is sound, it is not complete, as the CEGIS loop may not always terminate. To address this, we introduce a post-processing step that leverages the information acquired by the network over the sample points, generalizes it to the entire state space, and generates a valid upper bound of the JSR, even if the CEGIS loop fails. Figure 1 illustrates the full architecture.

This section is organised as follows: Section 3.1 improves Theorem 8 in Debauche et al. (2024) which provides some insight on the network structure needed to achieve a given precision. Section 3.2 describes in details the CEGIS architecture and Section 3.3 covers the derivation of a valid upper bound even if the CEGIS iterations end without a valid certificate.

3.1. Upper bounds on the structure of the network

In Debauche et al. (2024), we use the upper bound Theorem from McMullen (1970) to bridge the gap between Theorems 2 and 4. However, this auxiliary result artificially increases the theoretical bound. To overcome this problem, we prove a corollary of Theorem 2 using duality to get a bound on the required number of facets for the approximating polytope, rather than the vertices.

Proposition 1 *Let n, k be positive integers, and $\tau > 1$ be a real number such that (7) is satisfied. Then, for any symmetric, compact and convex set $K \subset \mathbb{R}^n$ with non-empty interior and containing the origin, there exists a symmetric polytope $P \subset \mathbb{R}^n$ with at most $8D(n, k)$ facets such that*

$$P \subseteq K \subseteq \tau P. \tag{12}$$

Proof Consider a symmetric, compact and convex set K with non empty interior in dimension $n \in \mathbb{N}$, a precision $\tau > 1$ and an integer $k \in \mathbb{N}$ which satisfies relation (7). The polar[3] of K, denoted by K^*, retains its original properties. By Theorem 2, there exists $\widetilde{P} \subset \mathbb{R}^n$ with at most $8D(n, k)$ vertices such that $\widetilde{P} \subseteq K^* \subseteq \tau\widetilde{P}$. Since duality reverses the inclusion, we have

$$(\tau\widetilde{P})^* \subseteq K^{**} \subseteq \widetilde{P}^* \Leftrightarrow \frac{1}{\tau}\widetilde{P}^* \subseteq K \subseteq \widetilde{P}^*$$

where $\widetilde{P}^*$ has as many facets as $\widetilde{P}$ has vertices. Posing $P := \frac{1}{\tau}\widetilde{P}^*$ ends the proof. ■

[3] Given any subset A of $\mathbb{R}^n$, the *polar* of A, denoted by A^*, is defined as $A^* := \{b \in \mathbb{R}^n \mid a^\top b \leq 1,\ \forall a \in A\}$.

Figure 2 represents the evolution of the precision error τ on the JSR approximation as a function of the number of facets as stated in Proposition 1. For comparison purpose, we display the previous bound obtained in Debauche et al. (2024). One can see that we drastically reduce the number of facets required to achieve a given precision. Moreover, the difference is even more pronounced as the dimension increases.

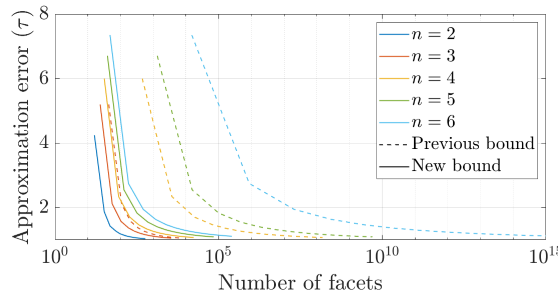


Figure 2: Evolution of the precision error (τ) of $\rho_{\mathcal{P}}$ with the number of facets of the corresponding polytopic sublevel set for $n = 1, \ldots, 6$. Dashed lines represent previous bounds, whereas solid lines show the improved bounds proposed in the present work.

We can now derive theoretical guarantees on the JSR approximation using CPWL functions with a bounded number of pieces.

Theorem 5 (Improvement on Theorem 4) *Let $\rho(\Sigma)$ be the joint spectral radius of a finite set of matrices Σ of dimension $n \in \mathbb{N}$. For any $\tau > 1$ and $k_\tau \in \mathbb{N}$ such that relation (7) is satisfied, the following relation holds:*

$$\frac{1}{\tau}\,\rho_{\mathcal{P}}(\Sigma) \;\leq \rho(\Sigma) \;\leq\; \rho_{\mathcal{P}}(\Sigma), \tag{13}$$

where $\rho_{\mathcal{P}}$ is the optimum solution of (3) where the template is restricted to CPWL norms with at most $8D(n,k)$ facets.

Proof Consider a finite set of matrices $\Sigma \subset \mathbb{R}^{n\times n}$ of dimension $n \in \mathbb{N}$, and $\rho(\Sigma)$ its joint spectral radius. By Theorem 1, for any $\varepsilon > 0$, there exists an ε-extremal norm $\|\cdot\|_\star$ such that expression (4) is satisfied. This norm defines a convex set $K \subset \mathbb{R}^n$ (which contains the origin) that can be approximated by a polytope. By Proposition 1, for any positive integer k and any real number $\tau > 1$ satisfying (7), there exists a symmetric polytope $P \subset \mathbb{R}^n$ with at most $8D(n,k)$ facets such that

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

Therefore, the (homogeneous) Minkowski function $V(\cdot)$ whose 1-level set is τP, satisfies that for all $x \in \mathbb{R}^n$, $V(x) \leq \|x\|_\star \leq \tau\, V(x)$, and then for any $A \in \Sigma$ and any $x \in \mathbb{R}^n$:

$$V(Ax) \;\leq\; \|Ax\|_\star \;\leq\; (\rho(\Sigma)+\varepsilon)\;\|x\|_\star \;\leq\; (\rho(\Sigma)+\varepsilon)\,\tau\, V(x).$$

Then, $V(A) \leq (\rho(\Sigma)+\varepsilon)\tau$ for all $A \in \Sigma$ and for any $\varepsilon > 0$, where $V(A)$ denotes the extension of V to matrices, analogous to a matrix norm induced by a vector norm. Therefore, we have that

$$\frac{1}{\tau}\,\rho_{\mathcal{P}}(\Sigma) \;\leq\; \rho(\Sigma) \;\leq\; \rho_{\mathcal{P}}(\Sigma),$$

where the second inequality is direct since we consider a subset of norms. ∎

One can therefore bypass McMullen's theorem, and directly combine Theorems 5 and 6 to derive a tighter bound on the network's structure to represent a Lyapunov function with a specified error.

Theorem 6 *Let $\rho(\Sigma)$ be the joint spectral radius of a finite set of matrices Σ of dimension n. For any real $\tau > 1$ and $k_\tau \in \mathbb{N}$ satisfying relation (7), there exists a CPWL function represented by a bias-free ReLU neural network of depth $\lceil \log_2(n+1) \rceil + 1$ and width*

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

which approximates $\rho(\Sigma)$ with a precision of τ.

Proof This theorem directly results from the consecutive application of Theorems 2 and 4. ∎

3.2. CEGIS approach with formal verification

Since the search for a Lyapunov function can be written as a second order logical formula, we propose a CEGIS implementation: an inductive loop between the learner which seeks a candidate Lyapunov function with an optimised sample-based estimate of the JSR, and the verifier which checks the validity of the Lyapunov inequalities over the whole domain for the candidate function.

3.2.1. LEARNER

The first component of our CEGIS architecture is the *learner* which trains a ReLU neural network over a set of sample points $\mathcal{S}$ according to hyper-parameters θ of the network. The outcome of this learning procedure is twofold since it instantiates a candidate Lyapunov function $NN_\theta : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and provides a sample-based approximation of the JSR of Σ, denoted by $\hat{\rho}_{NN}(\Sigma, \mathcal{S})$. The loss function of the network is tuned to find the best approximation of the JSR (see (14) below). We set the biases to zero, and thus, it is easily seen that any function encoded by the network is continuous, radially unbounded and positively homogeneous by construction. We enforce the positivity of the function by taking the absolute value of the output weights. Then, the gradient descent performs the minimization of the JSR approximation provided by the network, i.e. the loss function is

$$\hat{\rho}_{NN}(\Sigma, \mathcal{S}) \ := \ \max_{A \in \Sigma} \ \max_{x \in \mathcal{S}} \ \frac{NN_\theta(Ax)}{NN_\theta(x)}. \tag{14}$$

This quantity corresponds to the maximum sample-based approximation of the value of $NN_\theta(A)$ over all the matrices $A \in \Sigma$. Note that this quantity might be an invalid upper bound on $\rho(\Sigma)$ since we compute it over a finite subset $\mathcal{S}$ of the state space, which is why a verification step is introduced.

3.2.2. VERIFIER

By construction, we know that the function represented by the network, that is $NN_\theta(\cdot)$, is a candidate Lyapunov function. Moreover, it follows from the loss function in (14) that, for all the sample points in the training set $\mathcal{S}$, the Lyapunov inequalities in (6) are satisfied by $NN_\theta(\cdot)$ and the sample-based approximation of the JSR $\hat{\rho}_{NN}(\Sigma, \mathcal{S})$. However, these inequalities must be satisfied over the whole state space to be able to derive a valid upper bound on the JSR. Therefore, we use an SMT solver to check their validity and ensure soundness. If the SMT solver provides a counterexample, meaning a point where at least one of the Lyapunov inequalities is violated, this point (along with a few other points supplied by the consolidator, as we discuss above) is added to the training sample set for the next CEGIS iteration. Otherwise, the SMT solver confirms that there is no counterexample, and $\hat{\rho}_{NN}(\Sigma, \mathcal{S})$ is a valid upper bound on the JSR. Then, the CEGIS process stops.

Note that by linearity of the dynamics and homogeneity of the Lyapunov function, it suffices to check the Lyapunov inequalities on the unit ball. For computational efficiency, we use the infinity-norm unit ball, as it admits a linear description, unlike the 2-norm which requires nonlinear constraints.

3.3. Post Processing

The verification step can be quite challenging and may not end successfully; here, we introduce a novel technique to ameliorate an unsuccessful verification step in order to obtain a useful result.

In our setting, we generally rely on a small training sample set to speed up neural networm computations. However this can lead to poor generalization beyond the training data, where the neural network potentially fails to satisfy the desired properties outside the training data. As a result, in such cases, the SMT solver might return a counterexample at each iteration and the CEGIS loop might end up without valid approximation on the JSR. This limitation motivates the introduction of a norm that is constructed using the information learned from the sample points, and for which the corresponding JSR approximation can be easily computed.

Proposition 2 *Given a finite set of pairs* $\mathcal{S} = \{(x_k, y_k)\}_{k\in I} \subset \mathbb{R}^n \times \mathbb{R}_{\geq 0}$, *there exists a unique polytopic norm* $\|\cdot\|_{\mathcal{S}} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ *such that* $\forall k \in I,\ \|x_k\|_{\mathcal{S}} = y_k$. *Moreover, given a finite set of matrices* $\Sigma := \{A_1, \dots, A_M\} \subset \mathbb{R}^{n\times n}$, *the JSR approximation provided by this norm, denoted by* $\rho_{\mathcal{P},\mathcal{S}}(\Sigma)$ *can be computed by solving Linear Programs.*

Proof In what follows, we describe the construction of this norm, as illustrated in Figure 3.

We define the norm induced by $\mathcal{S}$ as the (unique and homogeneous) Minkowski function $\|\cdot\|_{\mathcal{S}} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ whose 1-sublevel set is given by

$$\overline{\mathcal{B}_{\|\cdot\|_{\mathcal{S}}}} := conv\left(\left\{v_k := \frac{x_k}{y_k} \;\middle|\; x_k \in \mathcal{S}\right\}\right).$$

By construction, the norm satisfies that $\|x_k\|_{\mathcal{S}} = y_k$ for any pair (x_k, y_k) in $\mathcal{S}$. Elsewhere, computing the norm $\|x\|_{\mathcal{S}}$ for $x \in \mathbb{R}^n$ amounts to solving the following Linear Program

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

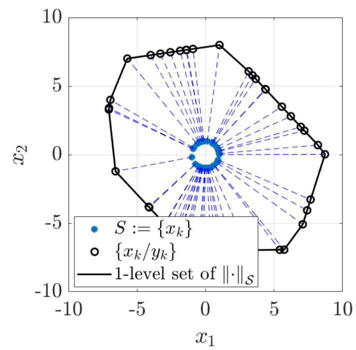


Figure 3: Illustration of the polytopic norm in Proposition 2.

and $\|x\|_{\mathcal{S}} = 1/\lambda^*$. In turn, the JSR approximation provided by $\|\cdot\|_{\mathcal{S}}$ can be computed as the maximum induced matrix norm $\rho_{\mathcal{P},\mathcal{S}}(\Sigma) := \max_{A\in\Sigma} \|A\|_{\mathcal{S}}$, where $\|A\|_{\mathcal{S}}$ is the maximum norm of Av_k over all the vertices v_k of unit ball $\overline{\mathcal{B}_{\|\cdot\|_{\mathcal{S}}}}$. ■

In our case, we consider the training sample set $\mathcal{S}$ and the corresponding outputs of the neural network on these points, i.e.,

$$\mathcal{S}_{NN} := \{(x, NN_\theta(x)) \mid x \in \mathcal{S}\}.$$

Since the training process optimizes not only $NN_\theta(x)$ but also $NN_\theta(Ax)$, it is meaningful to include these points as well. To that end, we define the augmented sample set as

$$\Sigma\mathcal{S}_{NN} := \{(A_i x, NN_\theta(A_i x)) \mid 0 \leq i \leq M, x \in \mathcal{S}\}$$

where $A_0 := I_n$, the n-dimensional identity matrix. The corresponding JSR approximations are denoted by $\rho_{\mathcal{P},\mathcal{S}_{NN}}(\Sigma)$ and $\rho_{\mathcal{P},\Sigma\mathcal{S}_{NN}}(\Sigma)$, respectively. In this setting, the network and the norm coincide both on the original sample points and their images under the system matrices. We expect the latter norm to yield a tighter upper bound on the JSR, at the cost of increased computational effort due to the larger number of vertices involved.

Crucially, the computation of these induced norms allow us to provide correct upper bounds on the JSR without the usage of SMT-solving. SMT-problems are in general NP-hard, so we provide a fail-safe in case of a timeout failure in the verification step.

4. Tool evaluation with benchmarks

We test our algorithm on several switched systems and compare its precision with the ellipsoidal approximation. All the experiments have been run on an Intel i7 laptop with 4 cores and 8GB of RAM. The neural networks are implemented using PyTorch and trained using AdamW (Loshchilov and Hutter (2017)), while we use Z3 (de Moura and Bjørner (2008)) as SMT solver.

We start with a 2-dimensional example with 2 matrices.

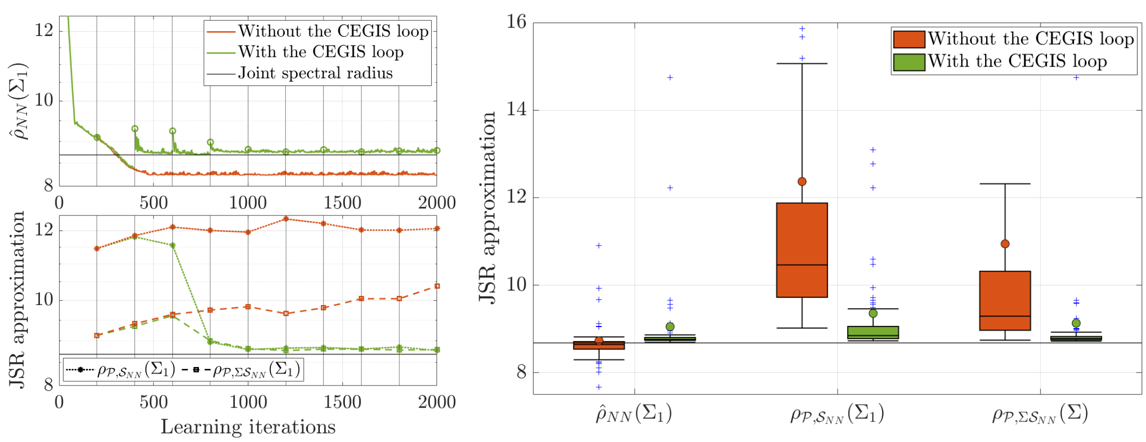
Example 1 *We consider the* 2*-dimensional linear switched system* $\Sigma_1 := \{A_1, A_2\} \subset \mathbb{R}^{2\times 2}$ *in* (*Debauche et al., 2024, Example 1*) *with 2 modes. The JSR is* 8.6881 *while* $\rho_{\mathcal{Q}}(\Sigma_1)$ *is* 9.5868*.*

Using our CEGIS algorithm, we approximate $\rho(\Sigma_1)$ *with a ReLU neural network with 1 hidden layer of 5 neurons, starting with 100 sample points and adding 20 after each SMT solver failure. The process stopped after 10 loops, and the SMT solver validated the approximation of 8.7090.*

*We motivate the CEGIS loop by training a ReLU neural network (*1 *single hidden layer,* 6 *neurons) with* 20 *initial sample points across* 100 *different seeds. We perform up to* 10 *loops of* 200 *learning iterations, adding* 10 *points after each SMT failure. For comparison, we let the network learn for* 2000 *iterations without any verification step, and we compute the post processed JSR approximations based on the last neural approximation. The results are summarized in Figure 4.*

Figure 4(a) represents the evolution of the neural and post processed JSR approximation over the learning iterations for one seed. In this example, the CEGIS loop does not stop but recursively integrating counterexamples into the sample set prevents the network from reaching an invalid JSR approximation. Then, the neural approximation is more reliable, even without a formal proof from the SMT. The CEGIS approach also improves the post processing; after 10 CEGIS iterations, $\rho_{\mathcal{P},\mathcal{S}_{NN}}(\Sigma_1)$ *and* $\rho_{\mathcal{P},\Sigma\mathcal{S}_{NN}}(\Sigma_1)$ *almost collapse and are* 8.77265 *and* 8.79399 *respectively, compared to* 12.07067 *and* 10.38091 *without the CEGIS loop.*

Figure 4(b) shows the box plot of neural and post processed JSR approximations with and without the CEGIS approach for 100 *different seeds. The results support the observations in Figure 4(a), showing that the CEGIS approach seems to reduce (and ultimately prevent) overfitting, resulting in a net decrease in the standard deviation of the JSR approximation (both neural and post processed). In addition, the network always generates valid JSR approximations (even if the SMT fails) and the precision of the post processed approximations is greatly improved. This is mainly due to their sensitivity to overfitting, with higher values of* $\rho_{\mathcal{P},\mathcal{S}_{NN}}(\Sigma_1)$ *and* $\rho_{\mathcal{P},\Sigma\mathcal{S}_{NN}}(\Sigma_1)$ *often correlating to invalid* $\hat{\rho}_{NN}(\Sigma_1)$*. Less overfitting brings the post-processed approximations closer to* $\hat{\rho}_{NN}(\Sigma_1)$*.*



(a) (b)

Figure 4: Comparison of the neural and post processed approximations of the JSR of system Σ_1 in Example 1 with (in green) and without (in red) the CEGIS approach. The horizontal black line represents $\rho(\Sigma_1)$. (a) Evolution of $\hat{\rho}_{NN}(\Sigma_1)$ (above), $\rho_{\mathcal{P},\mathcal{S}_{NN}}(\Sigma_1)$ and $\rho_{\mathcal{P},\Sigma\mathcal{S}_{NN}}(\Sigma_1)$ (below) with learning iterations for one seed, with vertical lines marking the completion of each CEGIS loop. (b) Box plot of $\hat{\rho}_{NN}(\Sigma_1)$, $\rho_{\mathcal{P},\mathcal{S}_{NN}}(\Sigma_1)$ and $\rho_{\mathcal{P},\Sigma\mathcal{S}_{NN}}(\Sigma_1)$ for 100 different seeds. The dots represent the mean. For clarity, outliers (+) above 16 are not displayed.

System				Ellipsoidal approx.	Neural approximation						Post processing	
Σ	n	M	$\rho(\Sigma)$	$\rho_{\mathcal{Q}}(\Sigma)$	$\hat{\rho}_{NN}(\Sigma)$	θ_1	Iters	$\lvert\mathcal{S}\rvert$	Loops	Time	$\rho_{\mathcal{P},\mathcal{S}_{NN}}(\Sigma)$	$\rho_{\mathcal{P},\Sigma\mathcal{S}_{NN}}(\Sigma)$
Σ_1	2	2	8.6881	9.5868	8.7090	[5]	400	100	10	19 [16]	8.7141	8.7107
					8.7298	[15]	400	300	0	7 [2]	8.7346	8.7298
Σ_2	2	3	4.5340	4.7794	4.5367	[8]	200	100	5	12 [6]	4.5402	4.5439
					4.5463	[10]	200	500	1	8 [2]	4.5471	4.5478
Σ_3	2	4	1.000	1.000	1.0014	[7]	300	500	10	23 [20]	1.0007	1.0007
					1.0016	[8]	300	500	0	3 [2]	1.0004	1.0004
Σ_4	3	2	0.9506	1.0171	$0.99^\star$	[10]	500	800	6	46 [17]	1.0063	1.0049
					$1.01^\star$	[9]	400	900	2	17 [6]	1.0471	0.9989
Σ_5	3	3	0.9194	0.9411	$0.94^\star$	[10]	450	1000	4	223 [15]	0.9526	0.9394
					$0.95^\star$	[10]	200	1000	3	168 [5]	0.9508	0.9438

Table 1: Evaluation of the numerical benchmarks of Section 4. For each system, the first line (white background) outlines the scheme with the best approximation of the JSR while the second line (gray background) outlines a sample execution with a good precision-computation time trade off. n is the dimension of the system, $\theta := [n, \theta_1, 1]$ are the parameters of the neural network, *Iters* is the number of learning iterations for each CEGIS loop, $|\mathcal{S}|$ is the initial number of sample points, *Loops* is the number of CEGIS loops and *Time* is the computation time for the whole CEGIS algorithm with the total learning time in brackets. The symbol $\star$ means that the neural JSR approximation has been rounded up to 0.01 accuracy and increased by 0.01 before the validation check by the SMT solver.

Finally, we tested our algorithm on a variety of switched systems of varying complexity, by increasing the dimension and the number of matrices. We consider the switched systems Σ_2 in (Della Rossa and Jungers, 2022, Example 3), Σ_3 in (Guglielmi et al., 2005, Example 6.4) Σ_4 in (Athanasopoulos and Jungers, 2019, Example 2) and Σ_5 in (Guglielmi and Zennaro, 2008, Section 7). The results, summarized in Table 1, show that our neural approach outperforms the ellipsoidal approximation in most cases, notably proving stability for system Σ_4, where the ellipsoidal method is inconclusive. While the CEGIS loop terminates, we also compute the post-processed JSR to assess its conservativeness. In practice, the difference between the post-processed and the neural approximation is typically on the order of 10^{-2}, highlighting the usefulness of post-processing.

5. Conclusion

This paper presents an automatic and sound algorithm to study the stability of linear switched systems by approximating the joint spectral radius of the corresponding set of matrices. This method leverages the strengths of two key components: the flexibility of neural networks and the soundness of SMT solvers. Through several numerical examples, we show that our algorithm performs competitively in low dimensions with the most common techniques, which is a first for a simulation-driven technique. In addition, we address a classical limitation of CEGIS-based methods, namely the absence of guarantees for procedure termination and thereby the supply of valid certificates, by introducing a post processing step. These two approaches seem to complement each other since our experiments reveal that each CEGIS loop enhances the post processed approximations, even when the SMT fails. We conjecture that our approach, with a smart sampling technique and potentially a tailored verifier, could significantly outperform current model-based approximation methods.

Acknowledgments

RJ is a FNRS honorary Research Associate. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 864017 - L2C, from the Horizon Europe programme under grant agreement No101177842 - Unimaas, the ARC (French Community of Belgium) - project name: SIDDARTA, and from the Advanced Research + Invention Agency (ARIA).

References

A. Abate, D. Ahmed, A. Edwards, M. Giacobbe, and A. Peruffo. Fossil: A software tool for the formal synthesis of Lyapunov functions and barrier certificates using Neural Networks. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, HSCC '21. Association for Computing Machinery, 2021.

A.A. Ahmadi, R.M. Jungers, P.A. Parrilo, and M. Roozbehani. Joint spectral radius and path-complete graph Lyapunov functions. *SIAM Journal on Control and Optimization*, 52(1), 2014.

N. Athanasopoulos and R.M. Jungers. Polyhedral path-complete Lyapunov functions. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019.

A. Barvinok. Thrifty Approximations of Convex Bodies by Polytopes. *International Mathematics Research Notices*, 2014(16), 04 2013.

V.D. Blondel and V. Canterini. Undecidable problems for probabilistic automata of fixed dimension. *Theory of Computing Systems*, 36, 2008.

V.D. Blondel and John N. Tsitsiklis. When is a pair of matrices mortal? *Information Processing Letters*, 63(5), 1997.

V.D. Blondel, Y. Nesterov, and J. Theys. On the accuracy of the ellipsoid norm approximation of the joint spectral radius. *Linear Algebra and its Applications*, 394, 2005.

Y.-C. Chang, N. Roohi, and S. Gao. Neural Lyapunov Control. In *arXiv:2005.00611 [Cs, Eess, Stat]*, December 2020.

S. Chen, M. Fazlyab, M. Morari, G. Pappas, and V. Preciado. Learning Lyapunov functions for hybrid systems. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, HSCC '21, New York, NY, USA, 2021. Association for Computing Machinery.

C. Dawson, Z. Qin, S. Gao, and C. Fan. Safe nonlinear control using robust Neural Lyapunov-barrier functions. In *Conference on Robot Learning*, 2021.

L. de Moura and N. Bjørner. Z3: An efficient SMT solver. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

V. Debauche, A. Edwards, A. Abate, and R.M. Jungers. Stability analysis of switched linear systems with neural Lyapunov functions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

M. Della Rossa and R.M. Jungers. Memory-based Lyapunov functions and path-complete framework: Equivalence and properties. In *2022 10th International Conference on Systems and Control (ICSC)*, 2022.

A. Edwards, A. Peruffo, and A. Abate. A General Framework for Verification and Control of Dynamical Models via Certificate Synthesis, 2023. arXiv:2309.06090.

A. Edwards, A. Peruffo, and A. Abate. Fossil 2.0: Formal Certificate Synthesis for the Verification and Control of Dynamical Models. In *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC, New York, NY, USA, 2024. Association for Computing Machinery.

M. Farsi, Y. Li, Y. Yuan, and J. Liu. A piecewise learning framework for control of unknown nonlinear systems with stability guarantees. In *Learning for Dynamics and Control Conference, L4DC 2022, 23-24 June 2022, Stanford University, Stanford, CA, USA*, volume 168 of *Proceedings of Machine Learning Research*. PMLR, 2022.

N. Guglielmi and M. Zennaro. An algorithm for finding extremal polytope norms of matrix families. *Linear Algebra and its Applications*, 428, 2008.

N. Guglielmi, F. Wirth, and M. Zennaro. Complex polytope extremality results for families of matrices. *SIAM Journal on Matrix Analysis and Applications*, 27(3), 2005.

C. Hertrich, A. Basu, M. Di Summa, and M. Skutella. Towards lower bounds on the depth of relu neural networks. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021.

F. John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays : Courant Anniversary Volume*. John Wiley & Sons : Interscience Division, New York, 1948.

R.M. Jungers. *The Joint Spectral Radius: Theory and Applications*, volume 385 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2009.

M. Lechner, Đ. Žikelić, K. Chatterjee, and T. Henzinger. Stability verification in stochastic control systems via Neural Network supermartingales. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7), June 2022.

I. Loshchilov and F. Hutter. Fixing weight decay regularization in adam. *CoRR*, 2017.

P. McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17(2), 1970.

P.A. Parrilo and A. Jadbabaie. Approximation of the joint spectral radius using sum of squares. *Linear Algebra and its Applications*, 428(10), 2008. Special Issue on the Joint Spectral Radius: Theory, Methods and Applications.

V. Petridis and S. Petridis. Construction of Neural Network based Lyapunov functions. 2006.

Spencer M. Richards, Felix Berkenkamp, and Andreas Krause. The Lyapunov Neural Network: Adaptive stability certification for safe learning of dynamical systems. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*. PMLR, 29–31 Oct 2018.

G.-C. Rota and W.G. Strang. A note on the joint spectral radius. In *Proceedings of the Netherlands Academy*, 1960.

Z. Sun and S. Ge. *Stability Theory of Switched Dynamical Systems*. Springer, London, 2011.

S. Zhang, Y. Xiu, G. Qu, and C. Fan. Compositional neural certificates for networked dynamical systems. In *Learning for Dynamics and Control Conference, L4DC 2023, 15-16 June 2023, Philadelphia, PA, USA*, volume 211 of *Proceedings of Machine Learning Research*. PMLR, 2023.