Formal Synthesis of Path-Complete Lyapunov Functions on Neural Templates

Virginie Debauche[1], Alec Edwards[2], Stella Simic[2], Raphaël M. Jungers[1,2], and Alessandro Abate[2]

[1]ICTEAM Institute, UCLouvain, B-1348 Louvain-la-Neuve, Belgium
[2]Department of Computer Sciences, University of Oxford, United Kingdom

1 Introduction

We study the stability analysis of discrete-time switched systems on $M \in \mathbb{N}$ modes of the form

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

where the state $x(k) \in \mathbb{R}^n$ for any $k \in \mathbb{N}$, and the switching signal $\sigma : \mathbb{N} \to \{1, \ldots, M\}$ determines the current dynamics at each time step. In particular, we study the stability *under arbitrary switching*, i.e. when there is no constraints on the switching signal. In this work, we merge two recently introduced tools; the path-complete Lyapunov framework [1] (PCLF) which is a graph-based generalization of the multiple Lyapunov function approach for switched systems, and FOSSIL [2] a software for the formal synthesis of Lyapunov functions based on neural networks.

On one side, the PCLF framework generalizes multiple Lyapunov functions for switched systems, and involves two structural parameters. First, a labeled and directed *graph* whose edges encode the Lyapunov inequalities, and a *template* which defines the search space for the Lyapunov functions as quadratic functions for instance. This framework provides in particular a characterization of the validity of a graph-based certificate thanks to the *path-completeness* property. Therefore, this formalism offers a large range of stability certificates, and the question of finding the best ones among them is still the focus of active research [3].
On the other side, FOSSIL [2] is a sound counterexample-guided inductive synthesis (CEGIS) method which is used to find one Lyapunov function for general dynamical systems. This software features two main components as illustrated in Figure 1: as a first step, the *learner* suggests a candidate Lyapunov function valid over a finite sets of points to the *verifier* which checks the validity of the function over the whole domain. If the verifier finds a point where the Lyapunov constraints are not satisfied, the point is added to the training set and the algorithm loops. Otherwise, the algorithm stops and produces a valid Lyapunov stability certificate. In FOSSIL, the learner trains a neural network while the verifier is an SMT solver.

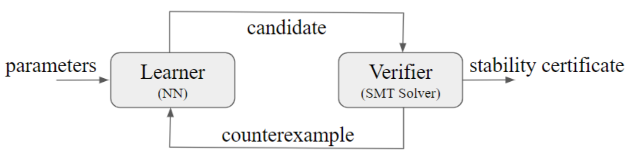

Figure 1: CEGIS loop of the FOSSIL software [2].

2 Results

In this work, we managed to adapt the FOSSIL architecture to graph-based stability certificates for switched systems in order to synthesize path-complete Lyapunov functions where the template is a neural network.

Concluding, the adaptation of the software FOSSIL to PCLF certificates allows to fully exploit the flexibility of both formalisms. In the future, we plan to leverage this flexibility in order to develop further the comparison of PCLF certificates in terms of performance by studying the modifications on the Lyapunov function when we add new layers, new neurons or when we modify the graph.

Acknowledgements

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.

References

[1] A.A. Ahmadi, R.M. Jungers, P.A. Parrilo and M. Roozbehani (2014). *Joint spectral radius and path-complete graph Lyapunov functions*. SIAM Journal on Control and Optimization, 52(1), pp. 687–717.

[2] A. Abate, D. Ahmed, A. Edwards, M. Giacobbe, & A. Peruffo (2021). *FOSSIL: a software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks*. Proceedings of the 24th ACM International Conference on Hybrid Systems: Computation and Control, pp. 1–11.

[3] V. Debauche, M. Della Rossa and R.M. Jungers, (2022). *Comparison of path-complete Lyapunov functions via template-dependent lifts*. Nonlinear Analysis: Hybrid Systems, vol. 26, p.101237.