Formal Verification of State and Temporal Properties of Neural Network-Controlled Systems

Antoine Besset[1], Joris Tillet[1] and Julien Alexandre dit Sandretto[1]

[1] ENSTA Paris, Institut Polytechnique
U2IS
Palaiseau, France
{antoine.besset,joris.tillet,alexandre}@ensta.fr

Keywords: Signal Temporal Logic, Verification of Neural Network, Interval Methods, Cyber-Physical Systems

Ensuring the safety of Neural Network Controlled Systems (NNCS) remains a major challenge due to the opaque nature of neural networks, especially when temporal properties are involved. This paper presents an interval analysis-based framework for verifying both state and temporal properties of NNCS using Signal Temporal Logic (STL) specifications [1–4]. We introduce an STL monitoring algorithm based on interval analysis, featuring adaptive time sampling and formal guarantees of satisfaction over continuous domains. The STL formalism allows a rich temporal property specification while our approach is broadly applicable to neural networks when activation functions can be expressed as Ordinary or Differential Algebraic Equations (ODEs/DAEs). Reachability analysis, following the differential approach of [5], employs an ODE solver with affine arithmetic to ensure tight enclosures and dependency tracking. We demonstrate the effectiveness of the method on two case studies, involving both NNCS and systems with temporal constraints.

To express temporal properties, we adopt a temporal logic formalism known as Signal Temporal Logic (STL) [1, 2]. It has been applied in the domains of robotics and control. STL formulas allow the expression of various temporal properties using explicit time bounds, combining logical connectives with bounded Until temporal operators ($U_{[a,b]}$) [1]. The syntax of STL is defined recursively as follows:

$$\phi := \mu \mid T \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \, U_{[a,b]} \, \phi_2. \tag{1}$$

We extend the verification of predicate (μ) with an inclusion predicate ($\mathcal{X}^\mu$) to verify properties on reachable tubes $y(t,[y_0]) \subseteq ([\tilde{y}],t),\ \forall t \in [t_0,T]$, [3,4]. A reachable set at t is $[\tilde{y}](t)$.

$$([\tilde{y}],t) \vDash \mu_i := \begin{cases} 1, & \text{if } [\tilde{y}](t) \subset \mathcal{X}^\mu, \\ 0, & \text{if } [\tilde{y}](t) \cap \mathcal{X}^\mu = \emptyset, \\ [0,1], & \text{otherwise.} \end{cases} \tag{2}$$

To evaluate satisfaction, we extend the STL syntax with the Boolean Interval Arithmetic [6, 7], enabling sound reasoning under uncertainty.

To conduct reachability analysis of an NNCS, one effective approach is to exploit the differential properties of its activation functions [5]. For instance, in the case of the sigmoid activation function $\sigma(x)$, it can be represented in the form of an ordinary differential equation (ODE) as follows:

$$\frac{d\sigma}{dx}(x) = \sigma(x)(1-\sigma(x)). \tag{3}$$

This formulation enables the use of ODE solvers within an affine arithmetic framework to compute guaranteed enclosures of the solutions while preserving the dependencies between individual neurons in the network. Supporting a broad spectrum of neural network architectures and expressive temporal logic specifications, the framework enables formal verification of practical NNCS scenarios. Comparative analysis with a Monte Carlo-based method highlights its precision and formal soundness.

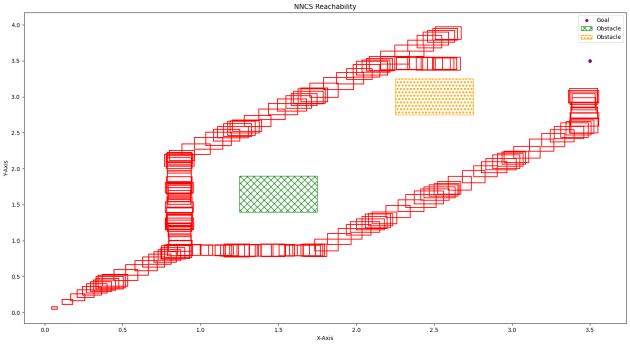

Figure 1: The 20-second simulation illustrates branching in the neural network output, with reachable tubes depicted in red. The axes indicate the position of the NN-controlled robot in meters. Branching arises from uncertainty in output classification. Left deviations around obstacles result in longer trajectories to the goal.

Acknowledgement

The authors acknowledge support from the French Interdisciplinary Center for Defense and Security (CIEDS) with the STARTS project.

References

[1] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, ser. Lecture Notes in Computer Science, vol. 3253. Springer, 2004, pp. 152–166.

[2] E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Ničković, and S. Sankaranarayanan, "Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications," *Lecture Notes in Computer Science*, vol. 10457, pp. 135–175, 2018.

[3] F. Lercher and M. Althoff, "Using four-valued signal temporal logic for incremental verification of hybrid systems," in *Proc. of Computer Aided Verification (CAV)*, vol. 14683. Springer Nature Switzerland, 2024, pp. 259–281.

[4] J. Tillet, A. Besset, and J. Alexandre dit Sandretto, "Guaranteed satisfaction of a signal temporal logic formula on tubes," *Acta Cybernetica*, 2025, accepted.

[5] R. Ivanov, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Verisig: Verifying safety properties of hybrid systems with neural network controllers," 2018.

[6] J. Alexandre dit Sandretto and A. Chapoutot, "Logical differential constraints based on interval boolean tests," in *Fuzzy Techniques: Theory and Applications*. Springer International Publishing, 2019, vol. 1000, pp. 788–792.

[7] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, "Applied interval analysis," in *Applied Interval Analysis*, L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, Eds. Springer, 2001, pp. 11–43.