Proceedings of the
1st Workshop on Artificial Intelligence and Formal Verification, Logics, Automata and Synthesis (OVERLAY),
Rende, Italy, November 19–20, 2019

Automated Verification of Noisy Nonlinear Cyber-Physical Systems with Ariadne *

Davide Bresolin[1], Luca Geretti[2], and Tiziano Villa[3]

[1]University of Padova, Italy
[2]University of Verona, Italy
[3]University of Verona, Italy
[1]davide.bresolin@unipd.it
[2]luca.geretti@univr.it
[3]tiziano.villa@univr.it

1 Introduction

A *cyber-physical system* (CPS) consists of a collection of computing devices communicating with each other and interacting with the physical world using sensors and actuators. Such systems are usually represented as a discrete control part that operates in a continuous environment, and are used in many application domains, like automotive, robotics, avionics, autonomous vehicles, process control, real-time and mobile computing systems [28]. The interplay between the discrete and the continuous dynamics causes difficulties in their analysis that are not present in discrete-only or continuous-only systems. This motivates the need for rigorous mathematical models and algorithms to describe and analyse them.

Hybrid automata are a convenient model for CPSs suitable for formal verification [3]. Intuitively, a hybrid automaton is a "finite-state automaton" with continuous variables that evolve according to dynamics characterising each discrete state (called a *location*). An execution of a hybrid automaton alternates *continuous* and *discrete* evolution. In the former, the location does not change, time passes and the evolution of the *state variables* follows some *flow predicate* associated to the current location. A discrete evolution step consists of the activation of a *discrete transition* that can change the current location and the value of the state variables, in accordance with the *reset function* associated to the transition. The interleaving of continuous and discrete evolution is regulated by *invariants*, which must be true for the continuous evolution to continue, while *guards* enable the discrete transitions.

Formal verification aims to identify system properties that are guaranteed to hold for every possible behaviour of the system. Such guarantee is based on the rigorous methodology underlying the computation or deduction of the desired properties. As a consequence, formal verification represents a powerful tool for evaluation of a system that has became standard practice for the development of HW/SW systems, and is becoming a vital aspect in the design of safety-critical CPSs, including robotics and automation systems [23, 24, 28]. More recently, some fruitful connections between AI and formal verification of hybrid systems has emerged. [27] proposes an approach for mining hybrid automata models from execution traces of embedded control systems. [19] uses Bayesian inference and reachability analysis to compute the confidence that a dynamical system with partly unknown dynamics verifies a given property. Both approaches allow formal verification techniques to be applied to black-box systems where the actual model

*This work was partially supported by MIUR, Project "Italian Outstanding Departments, 2018-2022" and by INDAM, GNCS 2019, "Formal Methods for Mixed Verification Techniques".

is unknown or only partially specified. The work in [22] shows how to transform a sigmoid-based neural network into an equivalent hybrid automaton, allowing state-of-the-art reachability tools to be used to verify safety properties of closed-loop systems where the controller is a neural network.

The computation of the *reachable set*, i.e., the set of all states that can be reached under the dynamical evolution starting from a given initial state set, is thus of particular importance in the formal verification of hybrid automata. Many approximation techniques and tools to estimate the reachable set have been proposed in the literature (see [28] for a comprehensive analysis). We recently proposed a development environment for the verification of nonlinear compositional hybrid systems, called Ariadne [7], which differs from existing tools by being based on the theory of computable analysis [12]. Such theory provides a rigorous mathematical semantics for the numerical analysis of dynamical systems, suitable for implementing formal verification algorithms. The tool has been applied mainly to the safety verification of robotic surgery tasks [8]. It also has been successfully used for dominance checking of controllers [6] and even for correct-by-construction code generation [9].

This paper discusses the ongoing work aimed at extending the dynamical model used in Ariadne to *differential inclusions*, based on the work of [33], in order to perform reachability analysis in the presence of noisy inputs. While the most straightforward application of differential inclusions is for modeling system uncertainty, it is worth remarking that they can be used also to support *contract-based design* [28]: given a complex system, we can replace the actual input of a component with an input having *partially defined behaviour*. The resulting decoupling of components ultimately allows to analyse subsystems in isolation, thus trading-off system complexity for precision.

Unfortunately, the introduction of differential inclusions into a nonlinear system represents a challenge in terms of controlling the quality of the computed reachable sets. Such control can be exercised using a number of precision parameters, which should be tuned dynamically for maximum effectiveness. In other words, the successful verification of a noisy system cannot disregard a thorough analysis of such precision parameters and the identification of a proper set of policies for their automated control.

2 Formal verification in the Ariadne framework

In this Section some insight on the approach used in Ariadne is provided, in order to understand the impact of the introduction of differential inclusions. Detailed technical information on the framework can be found in [13] about functional calculus and [6] regarding the reachability routines.

Suppose we wish to verify that a safety property φ holds for a hybrid automaton H; i.e., that φ remains true for all possible executions starting from a set X_0 of initial states, allowing to answer if a system operates within safe operating conditions expressed as a set. If this objective is cast as a reachability analysis problem, then it is necessary to prove that $\mathit{ReachSet}_H(X_0) \subseteq \mathrm{Sat}(\varphi)$, where $\mathit{ReachSet}_H(X_0)$ is the set of states reached by H (also called the *reachable set*) and $\mathrm{Sat}(\varphi)$ is the set of states where φ is true. Unfortunately, the reachability problem is not decidable in general [3]. Nevertheless, formal verification methods can be applied to hybrid automata: suppose we can compute an *outer* approximation $\bar{S}$ such that $\bar{S} \supseteq \mathit{ReachSet}_H(X_0)$. If $\bar{S} \subseteq \mathrm{Sat}(\varphi)$ holds, then also $\mathit{ReachSet}_H(X_0) \subseteq \mathrm{Sat}(\varphi)$ holds, i.e., the automaton H respects the property, or in other terms we *proved* the property. Conversely, if we can compute an *inner* approximation $\underline{S}$ such that $\underline{S} \subseteq \mathit{ReachSet}_H(X_0)$ that turns out to contain at least one point outside $\mathrm{Sat}(\varphi)$, we have proved that H does not respect the safety property φ, i.e., we *disproved* the property.

Clearly, any approximation to the reachable set is bound to the numerical precision used, hence a given quality of approximation may not allow to prove or disprove the property. Computable analysis defines the conditions to construct approximations such that if the precision is progressively increased, a sequence of approximations converging to the reachable set is obtained.

For a given precision, an approximation is obtained by identifying the reached region resulting from the evolution of the system over time. Such evolution is obtained through a sequence of continuous and discrete steps. A continuous step represents time advancement and relies on the integration of a vector field $\dot{x} = f(x)$ for a chosen step size h, where f is nonlinear in general. A discrete step represents a transition, which changes the *hybrid state*, i.e., the pairing of the continuous state and the discrete state, without any time advancement.

Does evolution return results similar to those of simulative tools like MathWorks Simulink®? No, because Ariadne is designed to include all the possible behaviours that result from evolving sets rather than single points. The underlying engine relies on results from *interval analysis*, which supports the definition of constants over intervals (among other things). Analysing a system in this case is equivalent to the simultaneous analysis of the set of singleton instances of the system, each corresponding to a distinct valuation of all constants. In particular, if a given constant represents a design parameter, parametric analysis [17] is able to identify subintervals where the constant yields optimal behaviour of the system with respect to some metrics.

Since intervals only model a set of constant behaviours, differential inclusions represent the most natural extension of the tool: by using them it is possible to analyse a system in which arbitrary *variations* of quantities within bounded intervals occur. The resulting over-approximation of behaviours covered by the noisy model can consequently compensate for an inaccurate system definition, which is a common problem when modelling real systems.

3 Differential inclusions

Differential inclusions (DIs) are dynamic systems of the form $\dot{x} \in F(x)$, where $F(\cdot)$ is a function that gives the *set* of possible values for the derivatives of x. DIs generalise differential equations by having multivalued right-hand sides [4, 32]. They model systems with (bounded, non-stochastic) uncertainties. DIs can be viewed as systems with input (control or noise) in the form $\dot{x} = f(x, v),\ v \in V \subset \mathbb{R}^m$, where $v(\cdot)$ is a measurable function [20]. If the set V of inputs is compact and separable, and $f(x, V)$ is continuous in x and convex for all x, the solution sets of the two systems are equivalent [4].

DIs arise in applications in engineering (e.g., robotics), physical and biological sciences in a variety of ways. They are used to model differential equations with discontinuities, by taking the closed convex hull of the right-hand side [15]. They can also be used to model systems with uncertain time-varying parameters, by replacing them with the whole set to which they could belong, as in [10, 30]. However, the most important use cases arise from the analysis of complex systems. One approach to analysis is to apply model-reduction techniques to replace a high-order system of differential equations $\dot{x} = f(x)$ by a low-order system of the form $\dot{z} = g(z) + e$, where $|e| \leq \epsilon$ represents the error introduced in simplifying the model (see [16]). Another way to analyse complex systems is to analyse separately their components. When the components depend on one another, we can decouple them by replacing an input from another component by noise varying over the range of possible values, again resulting in smaller but uncertain systems (see [11]). Note that stochastic models are not appropriate in these cases since inputs are not random.

To analyse reliably the behaviour and properties of a system, notably safety, uncertainties in the system must be taken into account when modelling, and rigorous numerical methods are necessary in order to provide guaranteed correct solutions. Designing numerical algorithms for computing solutions of DIs rigorously, efficiently and with high precision remains a point of current research. Different techniques and various types of numerical methods have been proposed in the literature. Some of the recent algorithms include ellipsoidal methods in [25], an interval Taylor-model approach in [26] and [10], Lohner-type algorithms in [30], optimal control in [5], set-based approximations to the Peano-Baker series in [2], hybrid bounding methods in [29], and a set-oriented method in [14]. However, none of these algorithms give both validated enclosures and guaranteed convergence with high-order precision for nonlinear DIs and thus are not of direct interest for comparison. We mention some closely related algorithms implemented by publicly available tools: [11] implemented within the tool Flow*, [31] implemented on top of CORA 2015, [21] implemented on top of CVODE and finally CORA 2018 (see [1]).

Our algorithm, whose initial version was presented in [33], considers a system with dynamics

$$\dot{x}(t) = f(x(t), v(t))\,,\ x(t) \in \mathbb{R}^n\,,\ v(t) \in V \subset \mathbb{R}^m \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a smooth function, V is a compact set and $v(t)$ is a measurable function known as the disturbance input. In particular, [33] discusses how to compute the reachable set for nonlinear control systems which are *input-affine*, i.e., affine with respect to noisy inputs. Also, a reasonable assumption in practice is that noisy inputs are elements of a box whose vector components are intervals.

The numerical approach focuses on (a) using an auxiliary function system to account for the input during a continuous step of evolution, then (b) adding the high-order theoretical error between the given system and the auxiliary one. Such approach is formally correct since it yields an over-approximation of the reachable set. However, the higher the order desired, the greater the number of parameters for the auxiliary system required for each continuous step, which clearly affects the efficiency of the algorithm. The question remains if the auxiliary system approach yields the best trade-off between precision and efficiency for computing reachable sets. The answer is not straightforward and most likely depends on the system itself.

The work presented in [18] illustrates how we implemented the algorithm from [33] in Ariadne, by analysing a benchmark of ten systems, whose results are then compared with Flow* and CORA 2018. The paper shows the importance of higher-order methods as more accuracy was achieved with two-parameter approximations on nine out of ten systems considered. Moreover, we found that Ariadne yields tighter set bounds, as the nonlinearity increases, compared with the other tools. Although no analysis of the order of the method is given in [10], we believe that Flow* has a local error $O(h^2)$, where h is the integration step size, so the global error is intrinsically first-order. Hence a higher quality is to be expected from Ariadne, since the proposed methodology is able to achieve third-order local errors. On the other hand, our approach introduces extra parameters at each step in the representation of the evolved set, causing a growth in complexity, whereas Flow* and CORA have a fixed complexity. As a result, the computational cost increases with both the noise level and the total number of steps taken. A comparison with the state-of-the-art using a common time budget indeed suggests that Ariadne currently provides better bounds for highly nonlinear systems.

4 Open issues and current work

The presence of differential inclusions introduces additional issues and opportunities in the analysis of continuous and hybrid systems, for which we are working on several improvements and extensions to the framework:

- **Improve reconditioning of the set.** The addition of parameters on each continuous step ultimately requires to simplify the set in order to keep the computation time reasonable. This operation necessarily introduces an additional over-approximation error which should be controlled in order to return a trajectory with acceptable quality. At the same time, adding new parameters from the uniform error allows to improve the computation of the flow set. Therefore an interesting open problem in the literature is the identification of the conditions that guarantee a given order of convergence with respect to the reconditioning policy.

- **Automation of accuracy parameter.** A manual, fixed tuning phase at the beginning of the reachability routine has a very limited capability to identify a (sub)optimal strategy for evolution. Instead, we are exploring a reasonable approach relying on a *pre-analysis* of the system by *point-based simulation*. In this case, we drop the guarantees given by set-based evolution, in order to gain valuable local information on the system evolution in a significantly shorter verification time vs. manual refinement through multiple analyses varying the accuracy parameters. The resulting information necessarily comes with no guarantees of correctness, meaning that the obtained evolution may include spurious transitions or miss some transitions. Still, for sufficiently well-behaved dynamics this approach is able to identify reached regions where evolution is numerically critical. Given such pre-analysis of the system, preemptive policies can be enacted that tune locally numerical parameters, with the goal to trade-off precision vs. verification time.

- **Extension to non-input-affine inclusions and more expressive input bounds.** The ability to decompose a system into several subsystems implies that we should be able to replace any variable of a differential equation with an input. Such operation may return a non-input-affine inclusion, which needs to be handled on an ad-hoc basis under our theoretical framework. Additionally, when a set of inputs represents the reachable set from another subsystem, its representation using a box introduces an excessive over-approximation. To address this issue, we are exploring the extension to polytope/zonotope bounds with nonlinear constraints.

Summarising, dealing with noisy nonlinear systems requires both local and global strategies in order to allow evolution to progress with bounded over-approximation error and reasonable efficiency of computation. Future work will focus on improving such strategies and on extending the tool to handle more general differential inclusions, with the overall objective of providing an automated way of analysing large-scale hybrid systems.

References

[1] M. Althoff, D. Grebenyuk, and N. Kochdumper. Implementation of Taylor Models in CORA 2018. In *Proc. of the 5th International Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 145–173, 2018.

[2] M. Althoff, C. L. Guernic, and B. H. Krogh. Reachable set computation for uncertain time-varying linear systems. In *Hybrid Systems: Computation and Control*, pages 93–102, 2011.

[3] R. Alur, C. Courcoubetis, T. A. Henzinger, and P. H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, volume 736 of *LNCS*, pages 209–229, Lyngby, DK, 1993. Springer.

[4] J. Aubin and A. Cellina. *Differential inclusions*, volume 264 of *Fundamental Principles of Mathematical Sciences*. Springer, 2002.

[5] R. Baier and M. Gerdts. A computational method for non-convex reachable sets using optimal control. In *Proceedings of the European Control Conference 2009*, pages 97–102, Budapest, HU, 2009. IEEE.

[6] L. Benvenuti, D. Bresolin, P. Collins, A. Ferrari, L. Geretti, and T. Villa. Ariadne: Dominance checking of nonlinear hybrid automata using reachability analysis. In *Reachability Problems*, volume 7550 of *LNCS*, pages 79–91. Springer, 2012.

[7] L. Benvenuti, D. Bresolin, P. Collins, A. Ferrari, L. Geretti, and T. Villa. Assume-guarantee verification of nonlinear hybrid systems with Ariadne. *Int. J. Robust. Nonlinear Control*, 24(4):699–724, 2014.

[8] D. Bresolin, L. D. Guglielmo, L. Geretti, R. Muradore, P. Fiorini, and T. Villa. Open problems in verification and refinement of autonomous robotic systems. In *15th Euromicro Conf. on Digital System Design (DSD)*, pages 469–476, Sept 2012.

[9] D. Bresolin, L. D. Guglielmo, L. Geretti, and T. Villa. Correct-by-construction code generation from hybrid automata specification. In *7th Int. Wireless Communications and Mobile Computing Conf. (IWCMC)*, pages 1660–1665, July 2011.

[10] X. Chen. *Reachability Analysis of Non-Linear Hybrid Systems Using Taylor Models*. PhD thesis, Aachen University, 2015.

[11] X. Chen and S. Sankaranarayanan. Decomposed reachability analysis for nonlinear systems. In *2016 IEEE Real-Time Systems Symposium (RTSS)*, pages 13–24, Nov 2016.

[12] P. Collins. Semantics and computability of the evolution of hybrid systems. *SIAM J. Control Optim.*, 49:890–925, 2011.

[13] P. Collins, D. Bresolin, L. Geretti, and T. Villa. Computing the evolution of hybrid systems using rigorous function calculus. In *Proc. of the 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS12)*, pages 284–290, Eindhoven, The Netherlands, June 2012.

[14] M. Dellnitz, S. Klus, and A. Ziessler. A set-oriented numerical approach for dynamical systems with parameter uncertainty. *SIAM J. on Applied Dynamical Systems*, 16(1):120–138, 2017.

[15] A. F. Filippov. *Differential Equations with Discontinuous Right-Hand Sides*, volume 18 of *Mathematics and its Applications*. Kluwer, 1988.

[16] L. Fortuna, G. Nunnari, and A. Gallo. *Model Order Reduction Techniques with Applications in Electrical Engineering.* Springer, 1992.

[17] L. Geretti, R. Muradore, D. Bresolin, P. Fiorini, and T. Villa. Parametric formal verification: the robotic paint spraying case study. In *Proceedings of the 20th IFAC World Congress*, pages 9248–9253, July 2017.

[18] L. Geretti, S. Živanović Gonzalez, P. Collins, D. Bresolin, and T. Villa. Rigorous continuous evolution of uncertain systems. In *Proc. 12th International Workshop on Numerical Software Verification (NSV2019)*, volume 11652 of *LNCS*, pages 60–75. Springer, 2019.

[19] S. Haesaert, P. M. V. den Hof, and A. Abate. Data-driven and model-based verification via Bayesian identification and reachability analysis. *Automatica*, 79:115 – 126, 2017.

[20] Z. Han, X. Cai, and J. Huang. *Theory of Control Systems Described by Differential Inclusions.* Springer Tracts in Mechanical Engineering. Springer-Verlag, 2016.

[21] S. Harwood and P. Barton. Efficient polyhedral enclosures for the reachable set of nonlinear control systems. *Mathematics of Control, Signals, and Systems*, 28(8), March 2016.

[22] R. Ivanov, J. Weimer, R. Alur, G. J. Pappas, and I. Lee. Verisig: Verifying safety properties of hybrid systems with neural network controllers. In *22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 169–178, 2019.

[23] T. L. Johnson. Improving automation software dependability: A role for formal methods? *Control Engineering Practice*, 15(11):1403–1415, 2007.

[24] H. Kress-Gazit. Robot challenges: Toward development of verification and synthesis techniques [from the guest editors]. *Robotics Automation Magazine, IEEE*, 18(3):22–23, Sept 2011.

[25] A. Kurzhanski and I. Valyi. *Ellipsoidal calculus for estimation and control.* Systems and Control: Foundations and Applications. Birkhäuser, 1997.

[26] Y. Lin and M. A. Stadtherr. Validated solutions of initial value problems for parametric ODEs. *Applied Numerical Mathematics*, 57(10):1145 – 1162, 2007.

[27] R. Medhat, S. Ramesh, B. Bonakdarpour, and S. Fischmeister. A framework for mining hybrid automata from input/output traces. In *2015 International Conference on Embedded Software (EMSOFT)*, pages 177–186, Oct 2015.

[28] P. Nuzzo, A. L. Sangiovanni-Vincentelli, D. Bresolin, L. Geretti, and T. Villa. A platform-based design methodology with contracts and related tools for the design of cyber-physical systems. *Proceedings of the IEEE*, 103(11):2104–2132, 2015.

[29] N. Ramdani, N. Meslem, and Y. Candau. A hybrid bounding method for computing an over-approximation for the reachable set of uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 54(10):2352–2364, October 2009.

[30] M. Rungger and G. Reissig. Arbitrarily precise abstractions for optimal controller synthesis. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1761–1768, Dec 2017.

[31] M. Rungger and M. Zamani. Accurate reachability analysis of uncertain nonlinear systems. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week), HSCC 2018, Porto, Portugal, April 11-13, 2018*, pages 61–70, 2018.

[32] G. V. Smirnov. *Introduction to the theory of differential inclusions*, volume 41 of *Graduate Studies in Mathematics.* American Mathematical Society, 2002.

[33] S. Zivanovic and P. Collins. Numerical solutions to noisy systems. In *IEEE Conf. on Decision and Control (CDC)*, pages 798–803, Dec 2010.