Ongoing Work on Automated Verification of Noisy Nonlinear Systems with Ariadne

Luca Geretti, Davide Bresolin, Pieter Collins, Sanja Zivanovic Gonzalez, Tiziano Villa

Ongoing work on automated verification of noisy nonlinear systems with Ariadne

Luca Geretti[1], Davide Bresolin[2], Pieter Collins[3], Sanja Zivanovic Gonzalez[4], and Tiziano Villa[1]

[1] Università di Verona, Verona, Italy {luca.geretti, tiziano.villa}@univr.it
[2] Università di Padova, Padova, Italy davide.bresolin@unipd.it
[3] Maastricht University, Maastricht, The Netherlands
pieter.collins@maastrichtuniversity.nl
[4] Barry University, Miami (FL), USA SZivanovic@barry.edu

Abstract. *Cyber-physical systems* (CPS) are *hybrid systems* that commonly consist of a discrete control part that operates in a continuous environment. Hybrid automata are a convenient model for CPS suitable for formal verification. The latter is based on *reachability analysis* of the system to trace its hybrid evolution and consequently verify its properties. However, when computing reachable states, a challenging task especially for nonlinear noisy systems is to control automatically the numerical precision to obtain meaningful approximations of the reached set. This paper presents the ongoing work and open issues in the automated computation of system evolution when the dynamics is described by differential inclusions. Differential inclusions allow to model noise for hybrid systems and also to decouple the components in a complex system, in order to simplify model-based design and verification. The proposed work aims to extend the capabilities of ARIADNE, a C++ library to perform formal verification of nonlinear hybrid systems.

1 Introduction

Formal verification is concerned with the identification of system properties that are guaranteed to hold for every possible behavior of the system itself. 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, compared to simulation techniques.

In this paper we focus on *hybrid systems*, i.e., dynamical systems that exhibit both a discrete and a continuous behavior. In order to model and specify hybrid systems in a formal way, the notion of *hybrid automaton* has been introduced [1]. Intuitively, a hybrid automaton is a "finite-state automaton" with continuous variables that evolve according to dynamics characterizing each discrete state (called a *location*).

Of particular importance in the analysis of hybrid automata is 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. Many approximation techniques and tools to estimate the reachable set have been proposed in the literature (see [16] for a comprehensive analysis). We recently proposed a development environment for the verification of nonlinear compositional hybrid systems, called ARIADNE [4], which differs from existing tools by being based on the theory of computable analysis [8]. 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 [6]. It also has been successfully used for dominance checking of controllers [3] and even for correct-by-construction code generation [7].

This paper discusses the ongoing work aimed at extending the dynamical model used in ARIADNE to *differential inclusions*, based on the work of [19], 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* [16]: given a complex system, we can replace the actual input of an automaton with an input having *partially defined behavior*. The resulting decoupling of automata ultimately allows to analyze subsystems in isolation, thus trading-off system complexity for precision.

Unfortunately, the introduction of differential inclusions to 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.

In the following, in Sec. 2 we start by presenting the approach used by ARIADNE for verification, in order to better understand how differential inclusions are a valuable addition to the framework. Then, a discussion on differential inclusions is provided in Sec. 3, followed by open issues related to automation aspects in Sec. 4.

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 [9] about functional calculus and [3] 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 *reach-*

able set) and $\mathrm{Sat}(\varphi)$ is the set of states where φ is true. Unfortunately, the reachability problem is not decidable in general [1]. 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 $\underline{\mathrm{S}}$ such that $\underline{\mathrm{S}} \subseteq \mathit{ReachSet}_H(X_0)$ 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 Δt, 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.

At a first glance, evolution may appear to return results similar to those of simulative tools like MathWorks Simulink®. Instead, Ariadne is designed to include all the possible behaviors 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). Analyzing 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 [11] is able to identify subintervals where the constant yields optimal behavior of the system with respect to some metrics.

Since intervals only model a set of constant behaviors, differential inclusions represent the most natural extension to the tool: by using them it is possible to analyze a system in which arbitrary *variations* of quantities within bounded intervals occur. The resulting over-approximation of behaviors 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

The seminal paper [19] that we are working to implement in Ariadne 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, [19] discusses

how to compute the reachable set for nonlinear control systems which are 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 one desires, 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.

Designing numerical algorithms for computing solutions of differential inclusions, both efficiently and with high precision, remains a point of current research. Different techniques and various types of numerical methods have been proposed as approximations to the solution set of a differential inclusion in the past. For example, ellipsoidal calculus was used in [15], a Lohner-type algorithm in [14], grid-based methods in [17] and [5], optimal control in [2], discrete approximations in [10, 12], and optimal control and support vector machines with grids in [18]. However, these algorithms either do not give rigorous over-approximations and so they cannot be used to validate the system, or are low-order approximations, e.g., Euler approximations with a first-order single-step truncation error.

Essentially, the only algorithms mentioned above that could give arbitrarily accurate error estimates are the ones that use grids. However, higher-order discretization of a state space greatly affects the efficiency of the algorithm. In fact, it was noted in [5] that if one tries to obtain higher-order error estimates on the solution set of differential inclusions then grid methods should be avoided.

A recent publication [13] proposes a method for computing outer approximations of reachable sets for nonlinear control systems by constructing convex polyhedral enclosures of reachable sets; it produces upper and lower bounds via polyhedra and demonstrates the efficiency of the proposed algorithm through several examples. Since all the examples are input-affine systems, we plan to compare this approach to the implementation of [19] within Ariadne.

Finally, in terms of theoretical extensions of the current approach, a desirable objective is to explore even higher-order error estimates. Additionally, we plan to use constraints for set representation, which allow for pseudo-affine inputs and inputs defined via more general convex sets. The ultimate goal however is the ability to handle differential inclusions which are nonlinear in the inputs.

4 Open issues for automation

The presence of differential inclusions introduces additional issues for continuous evolution, which require specific operations to be performed:

- **Reduction of auxiliary parameters.** Each continuous step increases the number of parameters by $2m$, where m is the dimension of the noise space. Consequently it is important to identify when some parameters can be lumped into a uniform error term δ, in order to reduce the dimensionality of the problem.
- **Reconditioning of the set.** When the uniform error term of the representation of the set becomes too large in respect to the set radius, it is beneficial to convert it into an additional parameter for the representation itself. Again, it is necessary to lump periodically one or more parameters into δ for scalability purposes. While reconditioning is a necessary operation in general, differential inclusions make its automation even more critical.
- **Splitting and recombining sets.** Additional precision can be obtained by splitting a large set over one parameter and evolving the split parts separately. However, the problem of identifying the conditions for an effective splitting is not trivial. Additionally, it is ultimately necessary to recombine split sets periodically to avoid an exponential explosion of the number of evolved sets. The problem is that recombination should introduce a small over-approximation error, in order to justify splitting in the first place.
- **Tuning of the continuous step size.** There is a trade-off to investigate between a large step size, which is unable to provide an accurate reachable set, and a small step size, which results in high complexity of the evolved set along with longer verification time.

In general, it is clear that local dynamics greatly affect the approximation error introduced in a single continuous step. As a consequence, a manual tuning phase at the beginning of the reachability routine has a very limited capability to identify a (sub)optimal strategy for evolution.

A reasonable approach relies on a *pre-analysis* of the system using *point-based simulation*. In this case, we drop the guarantees given by set-based evolution with the objective of gaining valuable local information on the system evolution in a significantly shorter verification time. 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 critical from the numerical viewpoint. Given such pre-analysis of the system, preemptive policies can be enacted to tune numerical parameters in order to trade between precision and verification time.

Summarizing, it appears that 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, with the objective of providing as much automation as possible regardless of the dynamics involved.

References

1. 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.
2. 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.
3. 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.
4. 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.
5. W.-J. Beyn and J. Rieger. Numerical fixed grid methods for differential inclusions. *Computing*, 81(1):91–106, 2007.
6. D. Bresolin, L. Di 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.
7. D. Bresolin, L. Di 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.
8. P. Collins. Semantics and computability of the evolution of hybrid systems. *SIAM J. Control Optim.*, 49:890–925, 2011.
9. 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.
10. T. Dontchev. Euler approximation of nonconvex discontinuous differential inclusions. *An. Stiint. Univ. Ovidius Constanta Ser. Mat.*, 10(1):73–86, 2002.
11. 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 9658–9663, July 2017.
12. G. Grammel. Towards fully discretized differential inclusions. *Set-Valued Anal.*, 11(3):1–8, 2003.
13. 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.
14. T. A. Kapela and P. Zgliczynski. A lohner-type algorithm for control systems and ordinary differential inclusions. *Discrete Contin. Dyn. Syst. Ser. B*, 11(2):365–385, March 2009.
15. A. Kurzhanski and I. Valyi. *Ellipsoidal calculus for estimation and control*. Systems and Control: Foundations and Applications. Birkhäuser, Basel, CH, 1997.
16. 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.
17. A. Puri, V. Borkar, and P. Varaiya. ϵ-approximation of differential inclusions. In *Proc. of the 34th IEEE Conference on Decision and Control*, pages 2892–2897, New Orleans, LA, USA, 1995. IEEE.
18. M. Rasmussen, J. Rieger, and K. Webster. Approximation of reachable sets using optimal control and support vector machines. *Journal of Computational and Applied Mathematics*, 311:68–83, February 2017.
19. S. Zivanovic and P. Collins. Numerical solutions to noisy systems. In *49th IEEE Conference on Decision and Control (CDC)*, pages 798–803, Dec 2010.