S

Safety Guarantees for Hybrid Systems

Raphael M. Jungers [1] and Nikolaos Athanasopoulos [2]
[1]UCLouvain, ICTEAM Institute, Louvain-la-Neuve, Belgium
[2]School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK

Abstract

Hybrid systems describe processes that typically need to satisfy a set of strict physical, computation, and communication constraints. Mission-critical and time-critical cyber-physical systems are a prime example where these constraints play a key role in analysis, controller synthesis, and implementation. On top of classical notions such as stability, safety plays a major role in the control design of hybrid systems. There is a long history of methods related to the safety analysis and safety enforcement for dynamical systems, with the ones concerning linear systems being more mature than the others. Due to the importance and complexity of the underlying problem, several different techniques have been developed for hybrid systems. This entry summarizes the most important approaches and tools, together with references for further reading.

Keywords

Hybrid systems · Safety · Verification · Reachability · Abstraction · Hybrid automata · Switching · Lyapunov methods · Formal methods

Introduction

More and more applications use advanced, data-driven, decentralized, nonlinear control solutions for uncertain or complex models. This complexification leads to unavailability of classical guarantees on the system's behavior. Safety-critical control is a leading challenge in hybrid and cyber-physical systems. Examples can be found in (air) traffic control, medicine, drug administration, logistics and supply chain networks, robotics, planning, the smart grid, autonomous vehicles and autonomous navigation, digital manufacturing and Industry 4.0, control over networks, and edge and cloud computing. Safety analysis of such systems is extremely difficult, with negative complexity results holding even for simple hybrid dynamics, for example, discrete-time switching systems (Jungers 2009) and rectangular hybrid automata (Doyen et al. 2018). Many different approaches have been established to address the problem of safety, often leading to semi-

R.J. and N.A. are supported by the CHIST-ERA 2018 project DRUID-NET "Edge Computing Resource Allocation for Dynamic Networks"

J. Baillieul, T. Samad (eds.), *Encyclopedia of Systems and Control*,
https://doi.org/10.1007/978-1-4471-5102-9_100049-1

algorithms with asymptotic properties. In view of the hardness of the task, it is critical to identify particular features of the system, which could make safety problems easier than in the general case. Indeed, for several special cases of hybrid systems, decidability or semi-decidability results are available. Let us mention reachability for timed automata, or the stability problem for switching systems (see, respectively, (Alur and Dill 1994; Jungers 2009), and (Doyen et al. 2018, Section 30.3) for precise statements). However, as a rule of thumb, one can argue that even elementary safety problems on very simple hybrid systems are often extremely hard.

Models

A *hybrid system* is a dynamical system consisting of both discrete (discrete event) and continuous dynamics, along with the corresponding sets where these two different dynamics are defined, often called *flow* and *jump* sets. A formal definition together with conditions for the existence of solutions is in Goebel et al. (2012, Sec. 1.1), covering, among others, switching and impulsive systems, and hybrid automata. For equivalence results between classes of hybrid systems, see Heemels et al. (2001), and for a formal definition for hybrid automata, see Doyen et al. (2018, Sec. 30.2.2). In this entry we consider the simple, however not restrictive, mathematical model representation

$$\begin{cases} \dot{x} = f(x,u,w), & x \in C, \\ x^+ = g(x,u,w), & x \in D, \end{cases} \tag{1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state vector, defined in a space that may consist of both discrete and continuous variables, $C \subset \mathbb{R}^n$, $D \subset \mathbb{R}^n$, $D \cap C = \emptyset$, are subsets of the state space and $u(t) \in \mathcal{U}$ and $w(t) \in \mathcal{W}$ are the control and exogenous inputs, respectively. We denote the solution of the system at time t^* for an initial condition $x_0 \in \mathcal{X}$ with $x(t^*; x_0)$. Further details on the model, as well as conditions and subtleties on the existence of solutions of (1), can be found in Goebel et al. (2012).

Safety

Safety is at the heart of many verification problems of hybrid systems. To introduce the concept, we first define the notion of *reachability*; a set $\mathcal{X}_e$ is *reachable* from x_0 if the state can reach $\mathcal{X}_e$ from x_0 after some finite time t, i.e., if $x(t; x_0) \in \mathcal{X}_e$. The notion of safety is related to *avoiding* regions of the state space; given a set of *unsafe states* $\mathcal{X}_f$ and a set of initial states $\mathcal{X}_0$, a system is *safe* if for any initial state x_0 in the set $\mathcal{X}_0$, there exists a sequence of actions such that it never reaches $\mathcal{X}_f$, i.e., $x(t; x_0) \notin \mathcal{X}_f$, for all $t > 0$ and all $x_0 \in \mathcal{X}_0$. We note that safety is meaningful also in the absence of control inputs, in the context of system analysis.

Safety Verification and Safety-Critical Control

As already mentioned, the literature on safety analysis for hybrid systems is huge, with ramifications in Mathematics, Computer Science, Robotics, etc. In the rest of the note, an attempt is made to address the challenge of classifying and briefly surveying the existing approaches. The following subsection presents perhaps the major and more natural technique, namely, set propagation. Next, optimization-based methods are presented, followed by an exposition of formal methods, which have received a strong influence from the Computer Science literature. The last subsection discusses probabilistic methods that seem particularly promising at the era of data-driven systems and massive computations.

Set Propagation Methods

Safety is connected to non-violation of constraints in the state space. Thus, a natural way of verifying it is to propagate relevant sets, e.g., constraint/initial/target sets, across time. This is called *reachability analysis* or *set propagation*.

Reachability analysis, e.g., Aubin et al. (2011), is strongly related to dynamic programming. In control, the connection has been made obvious and exploited with the model predictive control (MPC) paradigm and the

utilization of invariant sets for the enforcement of recursive feasibility, stability, and optimality for closed-loop control systems. Depending on whether one considers the propagation of dynamics of the system forward or backward in time, reachability may be utilized for different objectives, such as computing invariant sets and designing controllers (Blanchini and Miani 2008, Sections 5, 6), reaching a target set, verifying safety from an initial set, or satisfying more complex temporal specifications and objectives (Belta et al. 2017). We present below two popular instances of forward and backward reachability maps, noting however that several variants exist, which serve slightly different objectives and purposes.

Forward Reachability

Forward reachability can be utilized to study where the trajectories of the system will lie in the future when starting from some set of initial conditions of interest. Here, we restrict to systems with only exogenous inputs $w(\cdot)$, as it is the most common case where these mappings are utilized in the literature. Given an initial set $\mathcal{X}_0$, the approach consists in generating the sequence $\{\mathcal{R}_i\}$ (adapted from Doyen et al. 2018, Sec. 30.4.1) with

$$\mathcal{R}_{i+1} = \mathcal{R}_i \cup \mathcal{F}_C(\mathcal{R}_i) \cup \mathcal{F}_D(\mathcal{R}_i), \qquad (2)$$

$\mathcal{R}_0 = \mathcal{X}_0$, and

$$\mathcal{F}_C(\mathcal{R}_i) = \{x(t^*; x_0) : (\exists x_0 \in \mathcal{R}_i \cap C, \exists w(t) \in \mathcal{W} : x(t; x_0) \in C, t \in [0, t^*))\},$$
$$\mathcal{F}_D(\mathcal{R}_i) = \{g(x_0, w) : x_0 \in \mathcal{R}_i \cap D, w \in \mathcal{W}\}.$$

If all elements $\mathcal{R}_i$ of the generated sequence do not intersect with the set of unsafe states, i.e., $\mathcal{R}_i \cap \mathcal{X}_f = \emptyset$, then the system is safe.

Backward Reachability

Going backwards in time, we can define the sequence (adapted from Aubin et al. 2011, Sec 1.2.1.1, Viability Kernel) $\{\mathcal{C}_i\}$ generated by

$$\mathcal{C}_{i+1} = \mathcal{C}_i \cap (\mathcal{B}_C(\mathcal{C}_i) \cup \mathcal{B}_D(\mathcal{C}_i)), \qquad (3)$$

with $\mathcal{C}_0 = \mathcal{X} \setminus \mathcal{X}_f$, and

$$\mathcal{B}_C(\mathcal{C}_i) = \left\{x_0 \in C : (\exists t^* \geq \kappa, \exists u(t) \in \mathcal{U}, \forall t \in [0, t^*) : x(t; x_0) \in C \setminus \mathcal{X}_f, x(t^*; x_0) \in \mathcal{C}_i)\right\},$$
$$\mathcal{B}_D(\mathcal{C}_i) = \{x_0 \in D : (\exists u \in \mathcal{U} : g(x_0, u, w) \in \mathcal{C}_i)\}.$$

In the formula above, κ is an arbitrary positive constant, essentially limiting the dwell time for the continuous-time dynamics. Let us add that it is assumed in the formula above that the value u of the controller is valid for any possible value of w; however one can model different situations, where, e.g., the controller knows the input noise. If the limit of the generated sequence exists and is not empty, then the system is safe within this limit set, in the sense that all trajectories starting from it can be controlled such that they do not enter the unsafe region. Together with answering the safety problem, a by-product of the sequence is the establishment of safe, set-valued controllers, see, e.g., (Tomlin et al. 2003; De Santis et al. 2004). Moreover, many connections to Lyapunov theory through the notions of invariance and set contractivity have been identified (Blanchini and Miani 2008; Belta et al. 2017). We also remark that in the literature, different notions of safety/invariance have been developed, depending on the knowledge of the controller of the states, outputs, and exogenous inputs. Moreover, problems closely related to safety, such as the

reach-avoid problem, can be addressed by changing the set $\mathcal{C}_0$.

Computations

Reachability analysis produces sequences of sets generated by recursive updates. Thus, it is crucial to work with set parameterizations that can be described in a computer program with finite memory, and such that all operations described above are practically implementable. Polyhedral sets, especially in cases where linear dynamics and convex constraints are involved, are particularly appealing, since they are closed under Minkowski addition, intersection, convex union, and affine transformation. There is extensive literature tackling a great deal of settings involving linear dynamics and polyhedral sets, including linear parameter varying systems and switching systems. Nevertheless, even for the linear-convex case, computational complexity of the resulting sets can explode, especially for Minkowski sums, projections, minimal representations, and conversions between half-space and vertex representations, see, e.g., Fukuda (2004). A number of works deal with this issue by providing efficient alternatives using, e.g., zonotopes (Girard 2005), support function representations, and other classes of parameterized polytopes/template polyhedra. On the other hand, semialgebraic sets can also be employed for reachability analysis, such as ellipsoids, sublevel sets of SOS polynomials, or polynomials in the Bernstein representation. Moreover, one can leverage combinatorial techniques to reduce computational complexity in reachability analysis (Athanasopoulos and Jungers 2018).

Software

A number of fairly robust software solutions are available, often accompanied by modeling interfaces that allow easy formulation analysis and synthesis problems for hybrid systems and complex specifications. For toolboxes related to reachability analysis, we mention *SpaceEx, HyPro, CORA, d/dT, and JuliaReach* and note that the *MPT3* and *PPL* toolboxes are suitable for polyhedral computations.

Lyapunov-Based Optimization Techniques

The development of interior point optimization methods, enabling the efficient resolution of *linear matrix inequalities* (LMIs), has unlocked a huge literature in control. Linear matrix inequalities are particular algebraic optimization problems involving linear functions of the variables, and these variables can be nonnegative real numbers or positive definite matrices. In the 2000s, it was realized that the ability to efficiently solve such optimization problems actually allows to solve much more general ones, where the unknown variables are multivariate polynomials (of a fixed bounded degree), involving linear functions of these polynomials together with constraints requiring nonnegativity of these polynomials. This is the so-called sum-of-squares (SOS) programming (Parrilo 2000).

This new optimization technique had a great impact on *state-space methods* in control: State-space methods encode the objective function and the constraints directly in terms of the time, and of variables in the state space, by opposition to frequency domain techniques. In view of the intrinsic nonlinear and nonstandard behavior of hybrid systems, it is not surprising that such methods are well-adapted to solve control problems for hybrid systems (Parrilo 2000; Coogan and Arcak 2012). Even more, polynomials turn out to be efficient tools for representing *sets of points* in the state space. As a simple example, if, for some given multivariate polynomial $p(x)$, one defines the set

$$\mathcal{X}_p := \{x \in \mathbb{R}^n : p(x) \geq 0\},$$

then one can encode a sufficient condition for a safety constraint of the type $\forall x \in \mathcal{X}_p,\ Ax \in \mathcal{X}_q = \{x \in \mathbb{R}^n : q(x) \geq 0\}$ in the following formula:

$$\forall x \in \mathbb{R}^n, q(Ax) \geq p(x).$$

The above formula is *LMI-representable*, being composed of linear functions of the polynomials p, q. By combining such constraints, one can encode complicated control objectives in large SOS programs, which are then solved by standard

LMI solvers (e.g., MOSEK, SeDuMi, or SDPT3). Last, we should mention the closely related class of barrier methods; see, e.g., Prajna and Jadbabaie (2004), Sloth et al. (2012) and Maghenem and Sanfelice (2019).

Technicalities

Classical results from algebraic geometry imply that polynomials are universal, in the sense that they can approximate arbitrarily well any set, at the price of increasing the degree of the polynomial; see Henrion and Korda (2014) and references therein. This directly translates to universality of the SOS approach for representing sets in the state space.

It is important to mention that inequalities of the type $\forall x,\, p(x) \geq 0$ are in fact *not* efficiently solvable by LMI solvers. Instead, one classically proceeds to a *SOS relaxation*, requiring the weaker condition that the polynomial can be expressed as the sum of squares of other polynomials. This latter condition can be solved efficiently. Even though this relaxation induces conservativeness, one can show that by increasing the degree of the polynomial, the SOS relaxation is asymptotically non-conservative (Fig. 1).

Interior point methods are relatively efficient at solving sum-of-squares programs, as they essentially rely on the Newton method for convex functions optimization and hence have a cubic complexity at every step; see Boyd and Vandenberghe (2004, Section 11.5) for a detailed analysis. However, one should note that in practice, they suffer scalability issues for problems of moderate to large size. Even more, the size of the SOS program depends on the number of monomials of the maximal degree chosen for the unknown polynomials. Since this number of monomials grows exponentially with the degree, in practice, one is bound to limit the degree to a reasonable value (say, from 8 to 14, depending on the problems considered).

Scalability Improvements

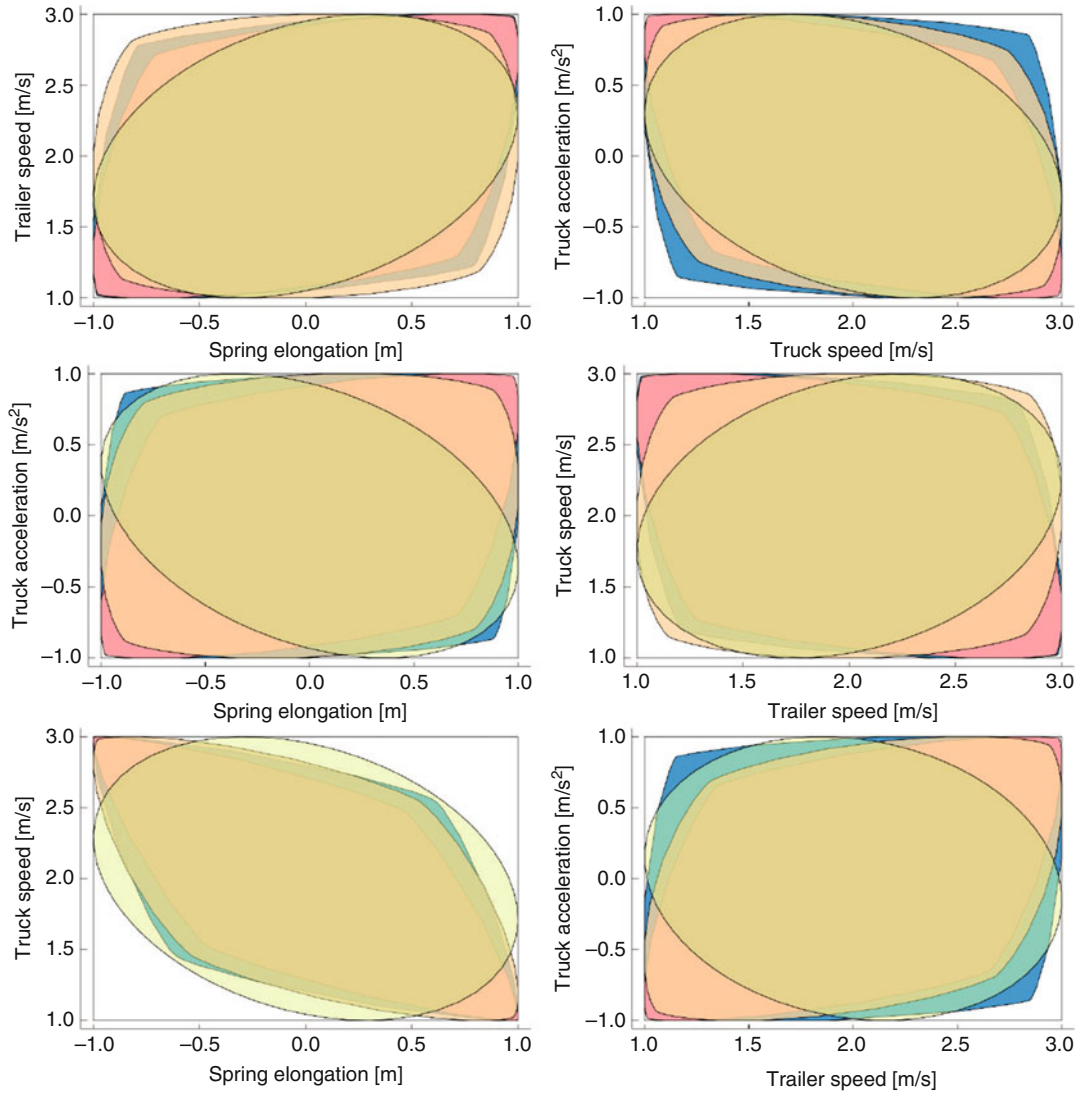
Recent work has focused on addressing the scalability issues. Of particular relevance for hybrid systems is the *path-complete approach* which, rather than increasing the degree of the polynomial, makes use of a combinatorial tool in order to enhance the representing power of polynomials by limiting their degree to a small number (Philippe et al. 2019). See Legat et al. (2018) for an application to safety-critical control of hybrid systems. Let us also mention the DSOS approach (Ahmadi and Majumdar 2014) which restricts SOS problems to particular subfamilies of polynomials, on which solvers perform better.

Software

The past decade has seen the creation of efficient and easy-to-use software for encoding polynomial constraints in a high-level language that are then automatically transformed in LMI programs and solved. See Gloptipoly3, Jump, cvx, SOSTOOLS, MOSEK, SeDuMi, and SDPT3, along with the parsers *JumP* and *YALMIP*.

Formal Methods and Abstraction

In view of the difficulty and criticality of safety problems, formal approaches have been proposed, influenced by the Computer Science and Model Checking communities. In these approaches, one formulates the safety problem in terms of rigorous predicates in a well-defined syntax corresponding to a particular logic. Typically, *first-order temporal logic* (Pnueli 1977) is well suited to formulate safety problems for dynamical systems, in particular hybrid systems. Then one needs to solve the logical equations formulated to verify safety of a given system or to construct an appropriate controller. To solve such complicated equations, one typically proceeds by *abstraction*, that is, by partitioning the state space into cells and the set of admissible inputs into small intervals, ending up with a finite-state automaton representing the dynamical system. Relying on standard continuity assumptions, many classical control problems (reachability, safety, reference tracking) can then be solved in a finite time, provided that the discrete abstraction satisfies *simulation* properties with respect to the actual system. The simulation property ensures that the behavior of the discrete, abstract system is a truthful representation of the original system. See Alur et al. (2000) for precise definitions. Thanks



Safety Guarantees for Hybrid Systems, Fig. 1 Example of the increasing performance of the SOS technique for a safe set computation problem, when one increases the degree of the unknown polynomial (in yellow for quadratic, orange for quartic, red for sextic, and blue for octic) (Legat et al. 2018). The problem is concerned with finding the range of safe speeds for a truck with trailers. The pictures are projections of the safe set in the four-dimensional state space, onto coordinate planes

to the finiteness of the obtained abstraction, the above problems reduce to a simple graph-theoretic one. For example, one can just find (or optimize) a path in the obtained automaton in order to solve the control problem. This confers a great advantage to the method in theory. See Haghverdi et al. (2005) and Alur (2011) for good introductions to formal methods.

The formal approach as applied until now suffers from important scalability problems. Indeed, a straightforward discretization of the state space in small cells inevitably leads to a huge set of nodes/edges in the automaton. More precisely, the size of the discretization step must be small, because the method relies on a local property, namely, continuity; and for a fixed discretization step, the number of cells grows exponentially

with the dimension of the state space. For this reason, except for systems of very low dimension (3 or 4), the approach does not work in practice. Nevertheless, these techniques are considered as among the most promising opportunities for complex problems on hybrid systems and are at the center of an intense research effort for the moment.

Software

Several software solutions are being developed and constantly improved, as, for instance, *Pessoa*, *CoSyMa*, *Tulip*, *SCOTS*, *Hytech*, *HSolver*, *Flow**, *PHAVer*, and *PHAVerLite* for general systems, *Uppaal*, *Kronos* for timed-automata oriented tools, and *Stochy* or *Prism* for probabilistic hybrid systems.

Probabilistic Techniques

Many models of hybrid systems are probabilistic. The techniques above can be adapted to such models, while specialized techniques have also been developed for the analysis of probabilistic hybrid systems; see Katoen (2016).

This subsection is not about such systems. Here we refer to probabilistic *techniques* that provide safety guarantees on either deterministic or probabilistic hybrid systems. Such techniques have recently received increasing attention as alternatives to the aforementioned methods, and the reason is obvious; due to the extreme computational cost, one may want to resort to probabilistic, Monte Carlo-like approaches to tackle the safety analysis problem while controlling at the same time the computational cost. In the community of systems and control, probabilistic techniques like the *scenario approach* (Calafiore and Campi 2006) are taking increasing importance. To our knowledge, the application of such approaches to hybrid systems has remained embryonic until now, but we expect a fast-growing body of work along these lines in the hybrid systems literature in the forthcoming years. Among such attempts, let us mention (Kenanian et al. 2019), which provides theoretical guarantees on probabilistic stability analysis for the particular case of switching systems, and (Julius and Pappas 2008) which makes use of stochastic bisimulation functions to provide a partial verification for more general hybrid systems.

In view of the intrinsic difficulty of coming with firm safety guarantees with probabilistic techniques, the state of the art of software implementation is less mature too. Let us mention S-Taliro which uses several randomized techniques for critical trajectory generation.

Summary and Future Directions

The problem of providing safety guarantees on a control system, called verification or model-checking in the Computer Science literature, is a paradigmatic problem that has been of paramount importance in applications for a long time. Today, the complexification of (data-driven, embedded, networked, etc.) control systems to hybrid systems makes it a challenge more than ever. It also makes it more important than ever, in view of the many emerging control systems interacting with our daily lives. In view of the theoretical computational barriers (undecidability, NP hardness, and others), one cannot hope that a single off-the-shelf technique could solve the problem, and as of today, ad hoc techniques have to be developed in order to cope with, and leverage, the specificities of the particular control problem at stake. Nevertheless, engineers have at their disposal several sound theoretical approaches, with solid software toolboxes, which can provide easily implementable solutions on simple models. We believe that in the future, these solutions will keep improving in performance and generality, as testified by the constant improvement on the state of the art over the past decades.

Cross-References

- Discrete Event Systems and Hybrid Systems, Connections Between
- Feedback Control of Hybrid Dynamical Systems
- Stability Theory for Hybrid Dynamical Systems

Bibliography

Ahmadi AA, Majumdar A (2014) DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization. In: 2014 48th annual conference on information sciences and systems (CISS). IEEE, pp 1–5

Alur R (2011) Formal verification of hybrid systems. In: 2011 proceedings of the ninth ACM international conference on embedded software (EMSOFT). IEEE, pp 273–278

Alur R, Dill DL (1994) A theory of timed automata. Theor Comput Sci 126(2):183–235

Alur R, Henzinger TA, Lafferriere G, Pappas GJ (2000) Discrete abstractions of hybrid systems. Proc IEEE 88(7):971–984

Athanasopoulos N, Jungers RM (2018) Combinatorial methods for invariance and safety of hybrid systems. Automatica 98:130–140

Aubin JP, Bayen AM, Saint-Pierre P (2011) Viability theory: new directions. Springer, Heidelber/Dordrecht/London/New York

Belta C, Yordanov B, Gol EA (2017) Formal methods for discrete-time dynamical systems. In: Studies in systems, decision and control. Springer, Heidelberg

Blanchini F, Miani S (2008) Set-theoretic methods in control. Birkhäuser, Boston

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge

Calafiore GC, Campi MC (2006) The scenario approach to robust control design. IEEE Trans Autom Control 51:742–753

Coogan S, Arcak M (2012) Guard synthesis for safety of hybrid systems using sum of squares programming. In: 2012 IEEE 51st annual conference on decision and control (CDC). IEEE, pp 6138–6143

De Santis E, Di Benedetto MD, Berardi L (2004) Computation of maximal safe sets for switching systems. IEEE Trans Autom Control 49(2):184–195

Doyen L, Frehse G, Pappas GJ, Platzer A (2018) Verification of hybrid systems. In: Handbook of model checking. Springer, Cham, pp 1047–1110

Fukuda K (2004) Frequently asked questions in polyhedral computation. Technical report, Swiss Federal Institute of Technology

Girard A (2005) Reachability of uncertain linear systems using zonotopes. In: Proceedings of the international workshop on hybrid systems: computation and control, pp 291–305

Goebel R, Sanfelice RG, Teel AR (2012) Hybrid dynamical systems: modeling stability, and robustness. Princeton University Press, Princeton

Haghverdi E, Tabuada P, Pappas GJ (2005) Bisimulation relations for dynamical, control, and hybrid systems. Theor Comput Sci 342(2–3):229–261

Heemels WP, De Schutter B, Bemporad A (2001) Equivalence of hybrid dynamical models. Automatica 37:1085–1091

Henrion D, Korda M (2014) Convex computation of the region of attraction of polynomial control systems. IEEE Trans Autom Control 59(2):297–312

Julius AA, Pappas GJ (2008) Probabilistic testing for stochastic hybrid systems. In: 2008 47th IEEE conference on decision and control. IEEE, pp 4030–4035

Jungers R (2009) The joint spectral radius: theory and applications, vol 385. Springer Science & Business Media, Berlin

Katoen JP (2016) The probabilistic model checking landscape. In: Proceedings of the 31st annual ACM/IEEE symposium on logic in computer science. ACM, pp 31–45

Kenanian J, Balkan A, Jungers RM, Tabuada P (2019) Data driven stability analysis of black-box switched linear systems. Automatica, 109, p. 108533

Legat B, Tabuada P, Jungers RM (2018) Computing controlled invariant sets for hybrid systems with applications to model-predictive control. arxiv preprint: https://arxivorg/abs/180204522

Maghenem M, Sanfelice RG (2019) Characterizations of safety in hybrid inclusions via barrier functions. In: 22nd ACM international workshop on hybrid systems: computation and control, pp 109–118

Parrilo PA (2000) Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. Ph.D. thesis, California Institute of Technology

Philippe M, Athanasopoulos N, Angeli D, Jungers RM (2019) On path-complete lyapunov functions: geometry and comparison. IEEE Trans Autom Control 64:1947–1957

Pnueli A (1977) The temporal logic of programs. In: 18th annual symposium on foundations of computer science (sfcs 1977). IEEE, pp 46–57

Prajna S, Jadbabaie A (2004) Safety verification of hybrid systems using barrier certificates. In: International workshop on hybrid systems: computation and control. Springer, Berlin, pp 477–492

Sloth C, Pappas GJ, Wisniewski R (2012) Compositional safety analysis using barrier certificates. In: International workshop on hybrid systems: computation and control, pp 15–24

Tomlin CJ, Mitchell I, Bayen AM, Oishi M (2003) Computational techniques for the verification of hybrid systems. Proc IEEE 91(7):986–1001