IRIS AperTO

UNIVERSITÀ DEGLI STUDI DI TORINO

AperTO - Archivio Istituzionale Open Access dell'Università di Torino

Verification and Control of Probabilistic Rectangular Hybrid Automata

This is the author's manuscript

Original Citation:

Availability:

This version is available http://hdl.handle.net/2318/1568125 since 2017-05-25T15:48:26Z

Publisher:

Springer

Published version:

DOI:10.1007/978-3-319-22975-1_1

Terms of use:

Open Access

Anyone can freely access the full text of works made available as "Open Access". Works made available under a Creative Commons license can be used according to the terms and conditions of said license. Use of all other works requires consent of the right holder (author or publisher) if not exempted from copyright protection by the applicable law.

(Article begins on next page)

20 April 2024

IRIS AperTO

UNIVERSITÀ DEGLI STUDI DI TORINO

This is the author's final version of the contribution published as:

Sproston, Jeremy. Verification and Control of Probabilistic Rectangular Hybrid Automata, in: Proceedings of the 13th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'15), Springer, 2015, 978-3-319-22974-4, pp: 1-9.

The publisher's version is available at:
http://link.springer.com/content/pdf/10.1007/978-3-319-22975-1_1

When citing, please refer to the published version.

Link to this full text:
http://hdl.handle.net/2318/1568125

This full text was downloaded from iris - AperTO: https://iris.unito.it/

iris - AperTO

University of Turin's Institutional Research Information System and Open Access Institutional Repository

Verification and Control of Probabilistic Rectangular Hybrid Automata*

Jeremy Sproston

Dipartimento di Informatica, University of Turin, Italy

Abstract. Hybrid systems are characterised by a combination of discrete and continuous components. In many application areas for hybrid systems, such as vehicular control and systems biology, stochastic behaviour is intrinsic. This has led to the development of stochastic extensions of formalisms, such as hybrid automata, for the modelling of hybrid systems, together with their associated verification and controller synthesis algorithms, in order to allow reasoning about quantitative properties such as "the vehicle's speed will reach 50kph within 10 seconds with probability at least 0.99". We consider probabilistic rectangular hybrid automata, which generalise the well-known class of rectangular hybrid automata with the possibility of representing random behavior of the discrete components of the system, permitting the modeling of the likelihood of faults, choices in randomised algorithms and message losses. We highlight the differences between verification and control problems for probabilistic rectangular hybrid automata and the corresponding problems for non-probabilistic rectangular hybrid automata. Furthermore, we will describe the effect of assumptions on the underlying model of time (discrete or continuous) on the considered verification and control problems. Finally, we will also consider how probabilistic rectangular hybrid automata can be used as abstract models for more general classes of stochastic hybrid systems.

1 Background and Motivation

Verification and Control. The development of correct and reliable computer systems can benefit from formal verification and controller synthesis methods. Both of these kinds of methods necessitate the precise specification of a set of requirements that the system should satisfy: examples are the avoidance of an error state or the repeated completion of a task. Formal verification considers a model of the proposed system, which is then subject to a method for determining whether the model satisfies its requirements. A typical formal verification method is that of model checking [7,4], in which the system is modelled formally as a transition system (or in a high-level modelling formalism which has transition systems as its semantics), the requirements are modelled using temporal logic formulae or

* Supported by the MIUR-PRIN project CINA and the EU ARTEMIS Joint Undertaking under grant agreement no. 332933 (HoliDes).

automata, and an algorithm determines whether the model of the system satisfies its requirements. Controller synthesis considers a partially-specified model of the system, which is subject to a method for restricting the behaviours of the system so that the restricted system satisfies a set of requirements. Controller synthesis is typically solves by representing the system as a two-player game, in which one player takes the role of the controller, which has the objective of restricting system behaviours in order to achieve the requirements, and the other player takes the role of the system's environment [5,24,23]. A winning strategy of the first player constitutes a control mechanism that ideally can be used as a basis of an implementation, so that the system's behaviour is restricted in such a way as to achieve the requirements.

Hybrid Automata. For the development of a wide range of computer systems, ranging from domestic appliances to vehicular controllers to medical devices, the interaction between the discrete behaviour of the digital system and the continuous behaviour of the environment in which the system operates is vital to understanding and reasoning about the overall system. Such systems are termed *hybrid systems*. Both verification and controller synthesis require a formal model of the system: in the setting of hybrid systems, *hybrid automata* [1] have been introduced for this purpose. A hybrid automaton consists of a finite directed graph and a set of real-valued variables, representing the discrete and continuous parts of the system, respectively. The discrete and continuous parts interact according to constraints on the continuous variables, and on their first derivatives, that label the nodes and edges of the graph. We refer to constraints on the first derivatives of variables as *flow constraints*. Variables can be *reset* to new values when an edge is traversed: resets are expressed as a relation between the previous value of the variable and its new one, and may involve nondeterministic choice (for example, a variable may be reset to any value in the interval $[1, 2]$ after taking an edge, where the choice of the new value is nondeterministic). For more information on hybrid automata, see [16,25].

The semantics of hybrid automata are presented by infinite-state transition systems: each state comprises a node and a valuation, that is a function associating a real value to each variable, and the transitions between states either correspond to the elapse of time or to the traversal of an edge of the hybrid automaton's graph. We say that the *continuous semantics* of hybrid automata corresponds to the case in which the durations of time-elapse transitions are taken from the set of non-negative reals, whereas the *discrete semantics* corresponds to the case in which time-elapse transitions can correspond to natural numbered durations only.

Probabilistic Hybrid Automata. In many application contexts for hybrid systems, system behaviours may have dramatically varying degrees of likelihood. Examples of events that typically have a low probability of occurring include faults, message losses or extreme meteorological conditions. In such settings, the traditional formulation of verification and control problems, with their Boolean view of system correctness, is insufficient: for example, message losses may be

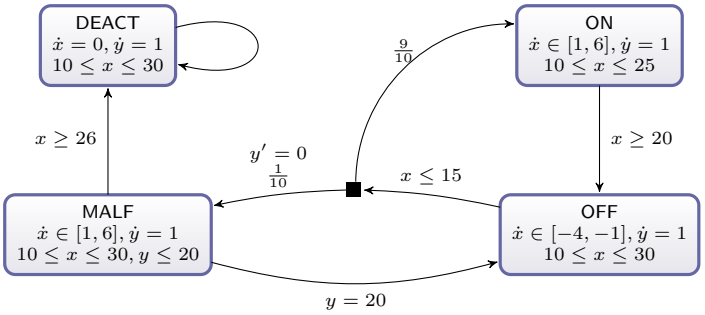

Fig. 1. A probabilistic hybrid automaton modelling a faulty thermostat

acceptable if the message can eventually be delivered within a specified deadline with high probability, and a lengthier journey of an automated vehicle may be acceptable in the case of uncharacteristically inclement weather. These facts have lead to the development of the field of *stochastic hybrid systems*. In the literature, formalisms including piecewise-deterministic Markov processes [8], controlled discrete-time Markov processes [30] and stochastic hybrid automata [12,15] have been considered. In this paper, we consider *probabilistic hybrid automata* [26,27], which extend the classical hybrid automaton formalism with the possibility to associate probability to the edges of the model's graph; from another perspective, they can be viewed as finite-state Markov decision processes (for verification) or finite-state stochastic games (for control) extended with continuous variables and their associated constraints, in the same way that hybrid automata can be seen as finite-state graphs extended with continuous variables. Probabilistic hybrid automata allow the modelling of probabilistic phenomena associated with the discrete part of the system, such as randomised choice between a finite number of alternatives of a digital controller, or the occurrence of a fault at the moment in which a discrete action is performed. We will discuss later in the paper how more general stochastic phenomena that considers the continuous behaviour of the system may be approximated using probabilistic hybrid automata.

An example of a probabilistic hybrid automaton modelling a faulty thermostat is shown in Figure 1. The ambient temperature is represented by the variable x, and variable y is a timer. When the heater is on (node ON or location MALF), the temperature increases at a rate between 1 and 6; when the heater is off (location OFF), the temperature changes at a rate between -4 and -1. The locations ON and OFF corresponds to non-faulty behaviour, whereas the location MALF corresponds to the heater being on in the presence of a fault in the temperature sensor that means that the measurement of the temperature is temporarily unavailable. The system passes from ON to OFF, with probability 1, when the temperature is between 20 and 25, and from OFF to ON, with probability $\frac{9}{10}$, or to MALF, with probability $\frac{1}{10}$, when the temperature is between 10 and 15. The sensor fault means that the temperature can increase to

a higher level in MALF than in ON. After a malfunction, either the system is deactivated if the temperature reaches an excessive level (location DEACT), or the system times-out exactly 20 time units after the location MALF was entered, in which case the heater is switched off. All edges of the probabilistic hybrid automaton correspond to reaching a certain location with probability 1, apart from the probabilistically branching edge from OFF.

When considering verification problems of formalisms based on Markov decision processes, such as probabilistic hybrid automata, we must take into account the fact that there are two types of choice in the model, namely nondeterministic choice and probabilistic choice. A *strategy* is a function that, given a finite execution of the model, returns the next action to be performed from the set of possible actions that can be chosen nondeterministically in the final state of the execution. Hence, a strategy resolves the nondeterministic choice of the model, but not the probabilistic choice. Given a particular system requirement and a particular strategy, we can then reason about the probability of satisfying the requirement when the nondeterministic choice of the model is resolved by the strategy. In particular we are interested computing in the maximum or minimum probability of satisfying a requirement. Hence verification typically takes the form of considering a requirement φ (for example reachability, which specifies that a state with a node F is eventually reached, but more generally an ω-regular property), and a threshold $\lambda \in [0, 1]$, and then relies on determining whether the maximum probability of satisfying φ is at least λ. Controller synthesis approaches take a similar form although, recalling that control of probabilistic systems is typically stated in terms of a stochastic game [3,6], in that setting there are strategies belonging to each player, and relies in determining whether the controller player can guarantee that φ is satisfied with probability at least λ, regardless of the behaviour of the environment player.

2 Probabilistic Rectangular Hybrid Automata

Methods for the verification and control of probabilistic hybrid automata, like the associated methods for classical hybrid automata, must take into account the fact that the underling state space of the model is infinite; more precisely, the semantics of a probabilistic hybrid automaton is described in terms of an infinite-state Markov decision process or stochastic game. We can identify a number of techniques for the verification and control of probabilistic hybrid automata: in this paper, we will focus mainly on the approach of constructing a finite-state Markov decision process, which is then analysed using classical methods for Markov decision processes. We also mention briefly two alternative methods. In [32], a "symbolic" search through the state space using non-probabilistic methods is performed first, after which a finite-state Markov decision process is constructed and analysed. Instead, [13,10] uses stochastic satisfiability modulo theories to permit the verification of bounded properties.

Recalling from above that subclasses of hybrid automata are generally characterised in terms of the form of the constraints associated with the nodes and

edges of the graph. In a similar way, we can characterise subclasses of probabilistic hybrid automata in terms of the form of constraints associated with the states and actions of the Markov decision process. In particular, a *probabilistic rectangular automaton* is a probabilistic hybrid automaton for which the constraints on continuous variables take the form of conjunctions of comparisons of a variable with a constant, and flow constraints take the form of conjunctions of comparisons of a first derivative of a variable with a constant (that is, the constraints of probabilistic rectangular automata have the same form as the constraints used in non-probabilistic rectangular automata [18]).

We now consider decidability results in the context of probabilistic rectangular automata, focusing first on verification problems with the continuous-time semantics. We first note that many verification problems for *probabilistic timed automata* [14,20], which are probabilistic hybrid automata for which variables are restricted to increasing at the same rate as real-time (or, equivalently, at the same rate in all nodes) are decidable. In particular, reachability problems for probabilistic timed automata are EXPTIME-complete [20,22]. These results can be generalised in the following way: letting $\mathcal{H}$ be a class of (non-probabilistic) hybrid automata, and letting $\mathcal{P}$ being the associated class of probabilistic hybrid automata (where $\mathcal{H}$ being associated with $\mathcal{P}$ means that the constraints used for both classes are of the same form), if the hybrid automata of the class $\mathcal{H}$ have finite bisimulation relations, then the probabilistic automata of the class $\mathcal{P}$ will have finite probabilistic bisimulation relations [29]. Given that, for an infinite-state Markov decision process with a finite probabilistic bisimulation relation, we can construct a finite-state Markov decision process that is equivalent with respect to a wide range of verification problems, this result means that a number of bisimulation-based decidability results for verification in the hybrid automata setting can be lifted to the probabilistic hybrid automata setting. For example, we can establish the decidability of a number of verification problems (including verification of ω-regular properties or properties of the probabilistic temporal logic PCTL*) for probabilistic hybrid automata that are probabilistic extensions of initialised multisingular automata [18], o-minimal automata [21] and STORMED hybrid automata [31]. This results relies crucially on the fact that the hybrid automata have finite bisimulation relations: intuitively, bisimulation takes into account the branching structure of the system, which then allows results on bisimulation for hybrid automata to be lifted to the probabilistic case. Indeed, we note that the reduction from initialised rectangular automata to timed automata presented in [18], which results in a language equivalent and not necessarily bisimilar timed automaton, can be adapted to the probabilistic case, as in done in [26,27], but results in an over-approximate model, rather than a faithful representation of the original initialised probabilistic rectangular automata.

Next we consider both verification and control of probabilistic rectangular automata with the discrete-time semantics. In this case, with the assumption that each variable is either non-decreasing (as time elapses) or remains within a bounded range throughout the model's execution, it is possible to obtain a finite

probabilistic bisimulation relation of the model, and hence a finite-state Markov decision process [28]. This result also allows controller synthesis algorithms to be applied. Hence verification of many requirements, such as reachability, safety and ω-regular properties, is EXPTIME-complete, whereas controller synthesis of the same classes of requirements can be done in NEXPTIME $\cap$ coNEXPTIME.

Instead, the problem of controller synthesis of probabilistic rectangular automata with the continuous-time semantics has received little attention so far. A notable exception is [11], in which a game version of probabilistic timed automaton is considered.

3 Approximation with Probabilistic Rectangular Hybrid Automata

A well-established approach in the field of modelling and verification is the construction of models that are amenable to verification and that over-approximate more faithful, but more difficult-to-verify models. In the context of classical hybrid automata, over-approximation generally consists of constructing a model whose set of observable behaviours contains all those of the original model. For example, rectangular automata have been used to approximate hybrid automata with more complex dynamics, in particular with respect to the constraints on the first derivatives of the variables [17,9]. In the context of probabilistic hybrid automata, or more general types of stochastic hybrid system, over-approximation generally consists of constructing a model for which, for any strategy σ of the original model, there exists a strategy σ' of the over-approximating model such that σ and σ' assign the same probability to observable events. This means that the maximum (minimum) probability of satisfying a certain requirement in the over-approximating model is no less than (no greater than) the probability of satisfying the requirement in the original model: that is, the maximum and minimum probabilities of a requirement in the over-approximating model bound those of the original model. Such an approach has been applied in the context of stochastic hybrid systems, in order to transform a model of a certain class of stochastic hybrid system to a model of an (ideally) easily-analysed class of probabilistic hybrid automata. The applications of the approach have taken two forms. over-approximation of flows (which extends the results of [17] to the probabilistic setting) [2], and over-approximation of probabilistic resets. We concentrate our attention on the latter.

Recall that, in the probabilistic hybrid automaton framework described above, a variable can be reset when traversing an edge. The mechanism of resetting variables is generalised in [19,12,15] to allow the possibility to reset variables according to continuous probability distributions, such as the uniform or normal distributions, and thus allowing the modelling of an increased range of probabilistic phenomena, such as measurement errors and uncertain times of events. The resulting formalism is called *stochastic hybrid automata*. The approach taken in [19,12,15] to analyse a stochastic hybrid automaton $\mathcal{S}$ is to construct over-approximating probabilistic hybrid automaton $\mathcal{P}$, where $\mathcal{P}$ is obtained from $\mathcal{S}$

by replacing the probabilistic choice involved in a probabilistic reset (over a continuous domain) by a discrete probabilistic choice (over a finite domain) over a number of intervals that cover the domain of the probabilistic reset, followed by a nondeterministic choice within the chosen interval. For example, consider the probabilistic reset in which a variable x is updated according to a uniform distribution over $[1, 3]$. The probabilistic reset can be replaced by a discrete probabilistic choice over (for example) the intervals $[1, 2]$ and $[2, 3]$, each of which correspond to probability $\frac{1}{2}$, in accordance with the original uniform distribution, and which is then followed by a nondeterministic choice over the chosen interval. If, in all other respects (nodes, flows etc.), $\mathcal{S}$ and $\mathcal{P}$ are identical, then $\mathcal{P}$ over-approximates $\mathcal{S}$.

The framework of over-approximation of probabilistic resets, with the aim of obtaining probabilistic rectangular automata, has been considered in [29]. In this context, stochastic hybrid automata are restricted as having rectangular-like constraints on flows and variables, although flows of the form $\dot{x} = y$, where y is constant as time passes, are allowed: this permits the modelling of situations in which the flow of a variable within a node is chosen according to a continuous probability distribution on entry to the node. It is shown that such stochastic hybrid automata can be over-approximated by probabilistic rectangular automata.

References

1. R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *TCS*, 138(1):3–34, 1995.
2. J. Assouramou and J. Desharnais. Analysis of non-linear probabilistic hybrid systems. In *Proc. QAPL'11*, volume 57 of *EPTCS*, pages 104–119, 2011.
3. C. Baier, M. Größer, M. Leucker, B. Bollig, and F. Ciesinski. Controller synthesis for probabilistic systems. In *Proc. TCS'04*, volume 155 of *IFIP*, pages 493–506. Kluwer/Springer, 2004.
4. C. Baier and J.-P. Katoen. *Principles of model checking.* MIT Press, 2008.
5. J. R. Büchi and L. H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the AMS*, 138:295–311, 1969.
6. K. Chatterjee and T. A. Henzinger. A survey of stochastic omega-regular games. *J. Comput. Syst. Sci.*, 2011. To appear.
7. E. Clarke, O. Grumberg, and D. Peled. *Model checking.* MIT Press, 2001.
8. M. H. A. Davis. *Markov Models and Optimization.* Chapman and Hall, 1993.
9. L. Doyen, T. A. Henzinger, and J-.F. Raskin. Automatic rectangular refinement of affine hybrid systems. In *Proc. FORMATS'05*, volume 3829 of *LNCS*, pages 144–161. Springer, 2005.
10. C. Ellen, S. Gerwinn, and M. Fränzle. Confidence bounds for statistical model checking of probabilistic hybrid systems. In *Proc. FORMATS'12*, volume 7595 of *LNCS*, pages 123–138. Springer, 2012.
11. V. Forejt, M. Kwiatkowska, G. Norman, and A. Trivedi. Expected reachability-time games. In *Proc. FORMATS'10*, volume 6246 of *LNCS*, pages 122–136. Springer, 2010.

12. M. Fränzle, E. M. Hahn, H. Hermanns, N. Wolovick, and L. Zhang. Measurability and safety verification for stochastic hybrid systems. In *Proc. HSCC'11*, pages 43–52. ACM, 2011.
13. M. Fränzle, T. Teige, and A. Eggers. Engineering constraint solvers for automatic analysis of probabilistic hybrid automata. *J. Log. Algebr. Program.*, 79(7):436–466, 2010.
14. H. Gregersen and H. E. Jensen. Formal design of reliable real time systems. Master's thesis, Department of Mathematics and Computer Science, Aalborg University, 1995.
15. E. M. Hahn. *Model checking stochastic hybrid systems.* Dissertation, Universität des Saarlandes, 2013.
16. T. A. Henzinger. The theory of hybrid automata. In *Proc. LICS'96*, pages 278–292. IEEE, 1996.
17. T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE TSE*, 43:540–554, 1998.
18. T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *J. Comput. Syst. Sci.*, 57(1):94–124, 1998.
19. M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Verifying quantitative properties of continuous probabilistic timed automata. In *Proc. CONCUR'00*, volume 1877 of *LNCS*, pages 123–137. Springer, 2000.
20. M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *TCS*, 286:101–150, 2002.
21. G. Lafferriere, G. Pappas, and S. Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems*, 13(1):1–21, 2000.
22. F. Laroussinie and J. Sproston. State explosion in almost-sure probabilistic reachability. *IPL*, 102(6):236–241, 2007.
23. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. POPL'89*, pages 179–190. ACM Press, 1989.
24. P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230, 1987.
25. J.-F. Raskin. An introduction to hybrid automata. In *Handbook of Networked and Embedded Control Systems*, pages 491–518. Birkhäuser, 2005.
26. J. Sproston. Decidable model checking of probabilistic hybrid automata. In *Proc. FTRTFT'00*, volume 1926 of *LNCS*, pages 31–45. Springer, 2000.
27. J. Sproston. *Model Checking for Probabilistic Timed and Hybrid Systems.* PhD thesis, School of Computer Science, University of Birmingham, 2001.
28. J. Sproston. Discrete-time verification and control for probabilistic rectangular hybrid automata. In *Proc. QEST'11*, pages 79–88. IEEE, 2011.
29. J. Sproston. Exact and approximate abstraction for classes of stochastic hybrid systems. In *Proc. AVOCS'14*, Electronic Communications of the EASST, pages 79–88, 2014.
30. I. Tkachev, A. Mereacre, J.-P. Katoen, and A. Abate. Quantitative automata-based controller synthesis for non-autonomous stochastic hybrid systems. In *Proc. HSCC'13*, pages 293–302. ACM, 2013.
31. V. Vladimerou, P. Prabhakar, M. Viswanathan, and G. E. Dullerud. STORMED hybrid systems. In *Proc. ICALP'08*, volume 5126 of *LNCS*, pages 136–147. Springer, 2008.
32. L. Zhang, Z. She, S. Ratschan, H. Hermanns, and E. M. Hahn. Safety verification for probabilistic hybrid systems. *European Journal of Control*, 18(6):572–587, 2012.