Verification and Control of Probabilistic Rectangular Hybrid Automata

Jeremy Sproston(✉)

Dipartimento di Informatica, University of Turin, Turin, Italy
sproston@di.unito.it

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 exhibited. 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 verification and control of probabilistic rectangular hybrid automata, which generalise the well-known class of rectangular hybrid automata with the possibility of representing random behaviour of the discrete components of the system, permitting the modelling of the likelihood of faults, choices in randomised algorithms and message losses. Furthermore, 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. A typical formal verification method is that of model checking [4,8], 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 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 solved by representing the system as a two-player game, in which one player takes the role of the controller, which has the objective of

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

S. Sankaranarayanan and E. Vicario (Eds.): FORMATS 2015, LNCS 9268, pp. 1–9, 2015.
DOI: 10.1007/978-3-319-22975-1_1

restricting system behaviours in order to achieve the requirements, and the other player takes the role of the system's environment [5,24,25]. 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 satisfy 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*. *Hybrid automata* [1] have been introduced as a formalism for hybrid systems. 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 value, 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 [17,26].

The semantics of a hybrid automaton is represented by an infinite-state transition system: 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-time 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-time semantics* corresponds to the case in which time-elapse transitions can correspond to natural numbered durations (and duration 0) 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 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, a message loss may be 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 led to the development of the field of *stochastic hybrid systems*. In the literature, a number of formalisms have been considered, of which we mention piecewise-deterministic Markov processes [6,10], controlled discrete-time Markov processes [31] and stochastic hybrid automata [13,16]. In this paper, we consider *proba-*

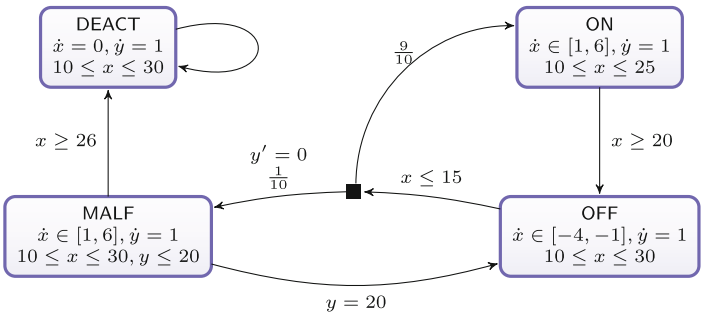

Fig. 1. A probabilistic hybrid automaton modelling a faulty thermostat

bilistic hybrid automata [27,28], which extend the classical hybrid automaton formalism with the possibility to associate probability to the edges of the model's graph. From another perspective, probabilistic hybrid automata 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 variables and constraints. 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. The semantics of a probabilistic hybrid automaton is represented by an infinite-state Markov decision process (when considering verification) or stochastic game (when considering control). As for hybrid automata, either a continuous-time or discrete-time semantics can be considered.

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 node 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 nodes ON and OFF corresponds to non-faulty behaviour, whereas the node 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 this example 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 [3,7], in that setting there are strategies belonging to each player. Hence we determine 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 or stochastic game, which is then analysed using well-established methods. We also mention briefly alternative methods for verification of probabilistic hybrid automata: in [33], "symbolic" search through the state space of a probabilistic hybrid automaton using non-probabilistic methods is performed first, after which a finite-state Markov decision process is constructed and analysed; [14] uses stochastic satisfiability modulo theories to permit the verification of bounded requirements; [9] employs a stochastic semantics for more general stochastic hybrid systems, to which statistical model checking is applied.

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 utilised. 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 [19]). The example in Figure 1 is a probabilistic rectangular automaton. We say that a probabilistic rectangular automaton is *initialised* if the value of a variable is reset when making a transition between nodes with flow constraints on that variable.

We focus first on verification problems with the continuous-time semantics. Many verification problems for *probabilistic timed automata* [15,21], which are probabilistic rectangular automata for which variables increase at the same rate as real-time, are decidable. In particular, reachability verification for probabilistic timed automata is EXPTIME-complete [21,23]. These results can be generalised in the following way: letting $\mathcal{H}$ be a class of (non-probabilistic) hybrid automata, and letting $\mathcal{P}$ be 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 [30]. 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 requirements expressed as ω-regular properties or as probabilistic temporal logic formulae) for probabilistic hybrid automata that are probabilistic extensions of initialised multisingular automata [19] (a subclass of probabilistic rectangular automata), and also for classes incomparable to rectangular automata, such as o-minimal automata [22] and STORMED hybrid automata [32]. This result relies crucially on the fact that the hybrid automata considered 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 [19], which results in a language equivalent and not necessarily bisimilar timed automaton, can be adapted to the probabilistic case [27,28], but obtains an over-approximate model, rather than a faithful representation of the original initialised probabilistic rectangular automaton.

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, but without the assumption of initialisation, it is possible to obtain a finite probabilistic bisimulation relation of the model, and hence a finite-state Markov decision process or stochastic game [29]. 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 [12], in which a game version of probabilistic timed automata is considered.

3 Approximation with Probabilistic 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 [11,18]. 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 automata, in order to transform a model of a certain class of stochastic hybrid automata to a model of an more 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 [18] 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 [13,16,20] 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 [13,16,20] to analyse a stochastic hybrid automaton S is to construct over-approximating probabilistic hybrid automaton P, where P is obtained from S by replacing the probabilistic choice involved in a probabilistic reset (over a continuous domain) by a discrete probabilistic choice (over a finite domain) between a number of intervals that cover the support of the probabilistic reset. After a probabilistic choice between the intervals, a nondeterministic choice is

made 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.), S and P are identical, then P over-approximates S.

The framework of over-approximation of probabilistic resets, with the aim of obtaining probabilistic rectangular automata, has been considered in [30]. 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. With regard to the example of Figure 1, in node ON, rather than increase nondeterministically with a rate in $[1, 6]$, we could consider that the rate of increase of the temperature is chosen on entry to the node from the normal distribution with mean 3.5 and standard deviation 1, truncated to the interval $[1, 6]$. It is shown that such stochastic hybrid automata can be over-approximated by probabilistic rectangular automata.

References

1. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. TCS **138**(1), 3–34 (1995)
2. Assouramou, J., Desharnais, J.: Analysis of non-linear probabilistic hybrid systems. In: Proc. QAPL 2011. EPTCS, vol. 57, pp. 104–119 (2011)
3. Baier, C., Größer, M., Leucker, M., Bollig, B., Ciesinski, F.: Controller synthesis for probabilistic systems (extended abstract). In: Levy, J.-J., Mayr, E.W., Mitchell, J.C. (eds.) TCS 2004. IFIP, vol. 155, pp. 493–506. Springer, Heidelberg (2004)
4. Baier, C., Katoen, J.-P.: Principles of model checking. MIT Press (2008)
5. Büchi, J.R., Landweber, L.H.: Solving sequential conditions by finite-state strategies. Transactions of the AMS **138**, 295–311 (1969)
6. Bujorianu, M.L., Lygeros, J.: Reachability questions in piecewise deterministic Markov processes. In: Maler, O., Pnueli, A. (eds.) HSCC 2003. LNCS, vol. 2623, pp. 126–140. Springer, Heidelberg (2003)
7. Chatterjee, K., Henzinger, T.A.: A survey of stochastic omega-regular games. J. Comput. Syst. Sci. **78**(2), 394–413 (2012)
8. Clarke, E., Grumberg, O., Peled, D.: Model checking. MIT Press (2001)
9. David, A., Du, D., Larsen, K.G., Legay, A., Mikucionis, M., Poulsen, D.B., Sedwards, S.: Statistical model checking for stochastic hybrid systems. In: Proc. HSB 2012. EPTCS, vol. 92, pp. 122–136 (2012)
10. Davis, M.H.A.: Markov Models and Optimization. Chapman and Hall (1993)
11. Doyen, L., Henzinger, T.A., Raskin, J.-F.: Automatic rectangular refinement of affine hybrid systems. In: Pettersson, P., Yi, W. (eds.) FORMATS 2005. LNCS, vol. 3829, pp. 144–161. Springer, Heidelberg (2005)

12. Forejt, V., Kwiatkowska, M., Norman, G., Trivedi, A.: Expected reachability-time games. In: Chatterjee, K., Henzinger, T.A. (eds.) FORMATS 2010. LNCS, vol. 6246, pp. 122–136. Springer, Heidelberg (2010)
13. Fränzle, M., Hahn, E.M., Hermanns, H., Wolovick, N., Zhang, L.: Measurability and safety verification for stochastic hybrid systems. In: Proc. HSCC 2011, pp. 43–52. ACM (2011)
14. Fränzle, M., Teige, T., Eggers, A.: Engineering constraint solvers for automatic analysis of probabilistic hybrid automata. J. Log. Algebr. Program. **79**(7), 436–466 (2010)
15. Gregersen, H., Jensen, H.E.: Formal design of reliable real time systems. Master's thesis, Department of Mathematics and Computer Science, Aalborg University (1995)
16. Hahn, E.M.: Model checking stochastic hybrid systems. Dissertation, Universität des Saarlandes (2013)
17. Henzinger, T.A.: The theory of hybrid automata. In: Proc. LICS 1996, pp. 278–292. IEEE (1996)
18. Henzinger, T.A., Ho, P.-H., Wong-Toi, H.: Algorithmic analysis of nonlinear hybrid systems. IEEE TSE **43**, 540–554 (1998)
19. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? J. Comput. Syst. Sci. **57**(1), 94–124 (1998)
20. Kwiatkowska, M., Norman, G., Segala, R., Sproston, J.: Verifying quantitative properties of continuous probabilistic timed automata. In: Palamidessi, C. (ed.) CONCUR 2000. LNCS, vol. 1877, pp. 123–137. Springer, Heidelberg (2000)
21. Kwiatkowska, M., Norman, G., Segala, R., Sproston, J.: Automatic verification of real-time systems with discrete probability distributions. TCS **286**, 101–150 (2002)
22. Lafferriere, G., Pappas, G., Sastry, S.: O-minimal hybrid systems. Mathematics of Control, Signals, and Systems **13**(1), 1–21 (2000)
23. Laroussinie, F., Sproston, J.: State explosion in almost-sure probabilistic reachability. IPL **102**(6), 236–241 (2007)
24. Pnueli, A., Rosner, R.: On the synthesis of a reactive module. In: Proc. POPL 1989, pp. 179–190. ACM Press (1989)
25. Ramadge, P.J., Wonham, W.M.: Supervisory control of a class of discrete-event processes. SIAM Journal of Control and Optimization **25**(1), 206–230 (1987)
26. Raskin, J.-F.: An introduction to hybrid automata. In: Handbook of Networked and Embedded Control Systems, pp. 491–518. Birkhäuser (2005)
27. Sproston, J.: Decidable model checking of probabilistic hybrid automata. In: Joseph, M. (ed.) FTRTFT 2000. LNCS, vol. 1926, pp. 31–45. Springer, Heidelberg (2000)
28. Sproston, J.: Model Checking for Probabilistic Timed and Hybrid Systems. PhD thesis, School of Computer Science, University of Birmingham (2001)
29. Sproston, J.: Discrete-time verification and control for probabilistic rectangular hybrid automata. In: Proc. QEST 2011, pp. 79–88. IEEE (2011)
30. Sproston, J.: Exact and approximate abstraction for classes of stochastic hybrid systems. In: Proc. AVOCS 2014. Electronic Communications of the EASST, pp. 79–88 (2014)
31. Tkachev, I., Mereacre, A., Katoen, J.-P., Abate, A.: Quantitative automata-based controller synthesis for non-autonomous stochastic hybrid systems. In: Proc. HSCC 2013, pp. 293–302. ACM (2013)

32. Vladimerou, V., Prabhakar, P., Viswanathan, M., Dullerud, G.E.: STORMED hybrid systems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 136–147. Springer, Heidelberg (2008)
33. Zhang, L., She, Z., Ratschan, S., Hermanns, H., Hahn, E.M.: Safety verification for probabilistic hybrid systems. European Journal of Control **18**(6), 572–587 (2012)