What will be eventually true of polynomial hybrid automata?*

Martin Fränzle**

Department of Computer Science
Carl-von-Ossietzky Universität Oldenburg
P.O. Box 2503, D-26111 Oldenburg, Germany

Abstract. Hybrid automata have been introduced in both control engineering and computer science as a formal model for the dynamics of hybrid discrete-continuous systems. While computability issues concerning safety properties have been extensively studied, liveness properties have remained largely uninvestigated. In this article, we investigate decidability of state recurrence and of progress properties.
First, we show that state recurrence and progress are in general undecidable for polynomial hybrid automata. Then, we demonstrate that they are closely related for hybrid automata subject to a simple model of noise, even though these automata are infinite-state systems. Based on this, we augment a semi-decision procedure for recurrence with a semi-decision method for length-boundedness of paths in such a way that we obtain an automatic verification method for progress properties of linear and polynomial hybrid automata that may only fail on pathological, practically uninteresting cases. These cases are such that satisfaction of the desired progress property crucially depends on the complete absence of noise, a situation unlikely to occur in real hybrid systems.

Keywords: Hybrid systems; State recurrence; Progress properties; Decidability; Verification procedures

1 Introduction

Embedded digital systems have become ubiquitous in everyday life. Many such systems, and in particular many of the safety-critical ones, operate within or even comprise tightly coupled networks of both discrete-state and continuous-state components. The behaviour of such *hybrid discrete-continuous systems* cannot be fully understood without explicitly modelling the tight interaction of discrete and continuous dynamics. Tools for building such models and for simulating

* This article reflects work that has been partially funded by the Deutsche Forschungsgemeinschaft under contract DFG Da 206/5-2.

** Email: `Martin.Fraenzle@Informatik.Uni-Oldenburg.De`,
Phone/Fax: +49-441-798-3046/2145.

their dynamics are commercially available, e.g. Simulink with the Stateflow extension[1], Matrix$_X$[2], or Statemate MAGNUM[3]. Simulation is, however, inherently incomplete and has to be complemented by *verification*, which amounts to showing that the coupled dynamics of the embedded system and its environment is well-behaved, regardless of the actual disturbance. Basic notions of being well-behaved demand that the system under investigation may never reach an undesirable state (often called *safety*) or that it will eventually reach a desirable state (*progress*).

Verification of hybrid systems calls for integration of discrete and continuous reasoning within a single formal model. The hope is that such combined formalisms, e.g. those proposed in [GNRR93,AKNS95,AHS96], may ultimately help in developing real embedded systems. Among such formalisms, the automata-based ones provide the most immediate prospect for mechanization. Roughly, hybrid automata can be characterized as a *combination of finite automata* whose transitions are triggered by predicates on the continuous plant state *with a description of the evolution of the continuous plant*. The latter consists of a set of real-valued variables that are governed by sets of syntactically restricted differential (in-)equations from which a currently active set is selected depending on the current automaton state. Hybrid automata lend themselves to analysis by state-space exploration methods. The state of the art with respect to state-exploratory *automatic* verification methods for hybrid systems is marked by semi-decision procedure for state reachability, as available in e.g. the HyTech tool [HHWT95]. These can — at least in principle, if complexity does not become prohibitive — be used for effectively *falsifying* safety properties of hybrid system. However, as state reachability has been shown to be undecidable even for very simple hybrid systems [HKPV95], the complementary problem, i.e. *verifying* that no undesirable state may ever be reached, cannot always be done effectively.

However, a closer look at the proof technique used in [HKPV95] for showing undecidability of state reachability in linear hybrid automata reveals that the relevance of these undecidability results to hybrid control is questionable. The proofs use a representation of two-counter machines by hybrid systems which relies on encoding infinite information, namely the set of natural numbers, within a compact interval of continuous states, whereas the ubiquity of noise limits the information content encodable by any bounded continuous variable encountered in real hybrid systems to a finite value. Hence, on simple information-theoretic grounds, the undecidability results thus obtained can be said to be artefacts of an overly idealized formalization. They will not necessarily carry over to operationally more adequate models of hybrid dynamics.

While this implies that the particular proof pattern sketched above lacks physical interpretation, it does not yield any insight as to whether the state reachability problem for hybrid systems featuring noise is decidable or not. Yet,

[1] `http://www.mathworks.com/products/simulink/` and `http://www.mathworks.com/products/stateflow/`
[2] `http://www.windriver.com/products/html/matrixx_ds.html`
[3] `http://www.ilogix.com/products/magnum/statemate_magnum.pdf`

taking noise into account at least simplifies the verification problem, as the author has shown in a companion paper to the current: using a very simple model of noise, in [Frä99] the author devised an automatic verification method for *safety* of hybrid automata that can both falsify and verify automata and that may only fail on pathological, practically uninteresting, cases. It is able to decide safety for all hybrid automata where safety does not depend on the complete absence of noise, i.e. which, if not unsafe, can tolerate some amount of noise without becoming unsafe. In the present paper, we establish similar results for *progress* properties.

Outline. We start our investigation with a definition of polynomial hybrid automata, a generalization of the more frequently analyzed class of linear hybrid automata. The definitions as well as semi-decision procedures for safety and for state recurrence are exposed in the next section; furthermore, undecidability results for state reachability, for state recurrence, and for progress are given. In Section 3 we expand on progress properties. We expose a simple model of noisy automata which allows continuous evolutions to reach a neighborhood of the target state. Given that, we show that state recurrence and progress are closely related for hybrid automata subject to noise, even though these automata are infinite-state systems. Section 4 combines these results with a semi-decision method for length-boundedness of paths, yielding an almost complete automatic verification method for progress properties of polynomial hybrid automata. A sufficient semantic criterion for success of the verification method is also given. Section 5 discusses properties of that criterion, while Section 6 concludes with a review of related work.

2 Hybrid automata

We start our investigation by providing a formalization of hybrid automata. The class of hybrid automata we will deal with goes beyond linear hybrid automata in that we will allow polynomial (instead of only linear) activities and polynomial (instead of linear) predicates for state invariants, transition guards, and transition effects. The reasons for adopting a more general class than usual are twofold. First, all our results can be shown for this more general class without any extra effort. Second, all definitions, statements, and proofs concerning this class of hybrid automata are more compact as no need to keep state invariants separate from activity predicates and transition guards separate from transition effects arises. Instead, every state and transition can be described by just one polynomial predicate, formalised through the first-order logic over the real-closed field, denoted $\mathrm{FOL}(\mathbb{R}, +, \cdot)$ in the remainder.

Therefore, within this article, a *(polynomial) hybrid automaton of dimensionality d* ($d \in \mathbb{N}$) is a five-tuple

$$(\Sigma, \mathbf{x}, (act_\sigma)_{\sigma \in \Sigma}, (trans_{\sigma \to \sigma'})_{\sigma, \sigma' \in \Sigma}, (initial_\sigma)_{\sigma \in \Sigma}) \ ,$$

where Σ is a finite set, representing the discrete states, and $\mathbf{x} = (x_1, \ldots, x_d)$ is a vector of length d of variable names, the continuous variables of the hybrid system.[4] $(act_\sigma)_{\sigma \in \Sigma}$ is a Σ-indexed family of formulae from $\mathrm{FOL}(\mathbb{R}, +, \cdot)$ representing the continuous activities and corresponding state constraints, and $(trans_{\sigma \to \sigma'})_{\sigma, \sigma' \in \Sigma}$ is a doubly Σ-indexed family of formulae from $\mathrm{FOL}(\mathbb{R}, +, \cdot)$ representing the discrete transitions and their guarding conditions. Finally, the initial states of the automaton are given by the Σ-indexed family $(initial_\sigma)_{\sigma \in \Sigma}$ of formulae from $\mathrm{FOL}(\mathbb{R}, +, \cdot)$. The free variables of the $initial_\sigma$ predicates are from $\mathbf{x}$, while the free variables of the act_σ and the $trans_{\sigma \to \sigma'}$ predicates are from $\overleftharpoonup{\mathbf{x}}$ and $\mathbf{x}$, as these predicates relate pre-states, which are denoted by the hooked variables $\overleftharpoonup{\mathbf{x}}$, to post-states, which are denoted by undecorated variables $\mathbf{x}$.

The interpretation is as follows:

- An *activity predicate* act_σ defines the possible evolution of the continuous state while the system is in discrete state σ. Hooked variable names in the predicate refer to the values of the corresponding real-valued system variables before the activity, while undecorated variable names refer to the values thereafter. A satisfying valuation $I_{\overleftharpoonup{\mathbf{x}}, \mathbf{x}}$ of its free variables $\overleftharpoonup{\mathbf{x}}, \mathbf{x}$ is interpreted as: if the system is in state σ and its continuous variables have values $I_{\overleftharpoonup{\mathbf{x}}}$ then the continuous variables may evolve to $I_{\mathbf{x}}$ while staying in state σ.
  Note that this single predicate thus generalizes both the state invariants and the activity functions of classical hybrid automata. E.g. the activity predicate
  $$\exists \Delta t \,.\; \overleftharpoonup{x} + 3 \cdot \Delta t \leq x \,\wedge\, x < \overleftharpoonup{x} + 5 \cdot \Delta t \,\wedge\, 4 < x \,\wedge\, x \leq 10$$
  encodes both the linear activity $\dot{x} \in [3, 5)$, where $\dot{x}$ denotes the first derivative of x over time, and the state invariant $x \in (4, 10]$.
- A *transition predicate* $trans_{\sigma \to \sigma'}$ defines when the system may evolve from state σ to state σ' by a discrete transition (i.e., it specifies the transition guard) and what the effect of that transition on the continuous variables is. A satisfying valuation $I_{\overleftharpoonup{\mathbf{x}}, \mathbf{x}}$ of its free variables $\overleftharpoonup{\mathbf{x}}, \mathbf{x}$ is interpreted as: if the system is in state σ and its continuous variables have values $I_{\overleftharpoonup{\mathbf{x}}}$ then the system may evolve to state σ' with its continuous variables taking the new values $I_{\mathbf{x}}$.
  Assignments as present in hybrid automata are simple to represent in this framework: e.g. for a hybrid system with continuous variables $(x_1, \ldots, x_8)$ the assignment $x_7 := 5$ (where all other continuous variables are left unchanged) is encoded by the predicate $x_7 = 5 \wedge \bigwedge_{i \in \{1, \ldots, 6, 8\}} x_i = \overleftharpoonup{x_i}$. Accordingly, guards are simply encoded through predicates over the hooked variables: e.g. the guard $x_3 > 11$ is represented by $\overleftharpoonup{x_3} > 11$. A transition with guard $x_3 > 11$ and assignment $x_7 := 5$ is thus encoded by the predicate
  $$\overleftharpoonup{x_3} > 11 \,\wedge\, x_7 = 5 \,\wedge\, \bigwedge_{i \in \{1, \ldots, 6, 8\}} x_i = \overleftharpoonup{x_i} \quad .$$

[4] Here and in the following, we use the convention to print vectors of variables or values in boldface. All these vectors have length d.

Multiple different transitions between the same state pair can be represented by disjunction of their encodings.

It should be noted that there is no strict need to distinguish between activities and transitions in $\mathrm{FOL}(\mathbb{R}, +, \cdot)$. However, we do not want to deviate too much from standard expositions of hybrid automata. Furthermore, this distinction turns out to be convenient in the further development, as activities and transitions are dissimilar in our model of noise.

We call a hybrid automaton *deadlock-free* iff any state that is reachable through a transition has an activity-successor and, vice versa, any state that is reachable through an activity has a transition-successor, i.e. iff

$$\bigwedge_{\sigma,\sigma' \in \Sigma} \forall \overleftarrow{\mathbf{x}}, \mathbf{x} \,.\, \left(\mathit{trans}_{\sigma \to \sigma'} \Rightarrow \exists \mathbf{x}' \,.\, \left(\mathit{act}_{\sigma'}[\mathbf{x}/\overleftarrow{\mathbf{x}}, \mathbf{x}'/\mathbf{x}] \right) \right)$$

and

$$\bigwedge_{\sigma \in \Sigma} \forall \overleftarrow{\mathbf{x}}, \mathbf{x} \,.\, \left(\mathit{act}_{\sigma} \Rightarrow \bigvee_{\sigma' \in \Sigma} \exists \mathbf{x}' \,.\, \left(\mathit{trans}_{\sigma \to \sigma'}[\mathbf{x}/\overleftarrow{\mathbf{x}}, \mathbf{x}'/\mathbf{x}] \right) \right) \quad ,$$

where $[y/x]$ denotes substitution of x by y, are tautologies.

Dynamic behaviour. During execution, hybrid automata engage in an alternating sequence of discrete transitions and evolution phases, where the continuous variables evolve according to an activity. Hence a (partial) execution containing $n \in \mathbb{N}$ transitions comprises n transitions interspersed between $n+1$ evolution phases, where the final states (wrt. both discrete and continuous state components) of the evolution phases meet the initial states of the following transition and vice versa the final states of the transitions meet the initial states of the following evolution phase. A (finite or infinite) *path* thus is a sequence $\langle (\sigma_0, \mathbf{y}_0), (\sigma_1, \mathbf{y}_1), \ldots \rangle \in (\Sigma \times \mathbb{R}^d)^\star$ or $\in (\Sigma \times \mathbb{R}^d)^\omega$, resp., such that $\sigma_i = \sigma_{i+1}$ for even i, and

$$I_{\mathbf{x}:=\mathbf{y}_0} \models \mathit{initial}_{\sigma_0} \quad ,$$

$$I_{\overleftarrow{\mathbf{x}}:=\mathbf{y}_i, \mathbf{x}:=\mathbf{y}_{i+1}} \models \begin{cases} \mathit{act}_{\sigma_i} \ , & \text{if } i \text{ is even,} \\ \mathit{trans}_{\sigma_i \to \sigma_{i+1}} \ , & \text{if } i \text{ is odd} \end{cases}$$

for each i less than the length of the path. Here, $I_{\mathbf{x}:=\mathbf{y}}$ denotes the valuation

$$I_{\mathbf{x}:=\mathbf{y}}(v) = \begin{cases} P_i(\mathbf{y}), & \text{if } v = x_i \text{ for some } i \in \{1, \ldots, d\}, \\ & \quad \text{where } P_i \text{ denotes projection to the } i\text{-th component,} \\ I(v), & \text{otherwise.} \end{cases}$$

Note that the choice of I does not matter in above applications, as only variables in $\mathbf{x}$ occur free in $\mathit{initial}_{\sigma_0}$ and only variables in $\overleftarrow{\mathbf{x}}$ or $\mathbf{x}$ occur free in act_{σ_i} and $\mathit{trans}_{\sigma_i \to \sigma_{i+1}}$.

In accordance to the definition of paths, reachability of a discrete state σ' and a continuous state $I_{\mathbf{x}}$ from an discrete start state σ and a continuous start state

$I_{\overleftarrow{\mathbf{x}}}$ through an execution (i.e., a path segment that need not start in an initial state) of length n can be formalised through the inductively defined predicate $\phi^n_{\sigma\to\sigma'}$, where

$$\phi^0_{\sigma\to\sigma'} = \begin{cases} \texttt{false}, & \text{if } \sigma \neq \sigma' \ , \\ act_\sigma & \text{, if } \sigma = \sigma' \ , \end{cases}$$

$$\phi^{n+1}_{\sigma\to\sigma'} = \begin{cases} \bigvee_{\tilde\sigma\in\Sigma} \exists\, \tilde{\mathbf{x}}\,.\, \begin{pmatrix} \phi^n_{\sigma\to\tilde\sigma}[\tilde{\mathbf{x}}/\mathbf{x}] \wedge \\ trans_{\tilde\sigma\to\sigma'}[\tilde{\mathbf{x}}/\overleftarrow{\mathbf{x}}] \end{pmatrix} & \text{, if } n \text{ is even,} \\ \exists\, \tilde{\mathbf{x}}\,.\, \begin{pmatrix} \phi^n_{\sigma\to\sigma'}[\tilde{\mathbf{x}}/\mathbf{x}] \wedge \\ act_{\sigma'}[\tilde{\mathbf{x}}/\overleftarrow{\mathbf{x}}] \end{pmatrix} \ , & \text{if } n \text{ is odd.} \end{cases}$$

Accordingly,

$$Reach^{\leq n}_{\sigma'} \stackrel{\text{def}}{=} \bigvee_{i\in\mathbb{N}_{\leq n}} \bigvee_{\sigma\in\Sigma} initial_\sigma[\overleftarrow{\mathbf{x}}/\mathbf{x}] \wedge \phi^i_{\sigma\to\sigma'} \tag{1}$$

characterizes the continuous states reachable on a path of length at most n that ends in discrete state σ'. Consequently, state σ' is reachable iff there is some $n \in \mathbb{N}$ for which $Reach^{\leq n}_{\sigma'}$ is satisfiable.

Note that $Reach^{\leq n}_{\sigma'}$ is a formula of $\mathrm{FOL}(\mathbb{R}, +, \cdot)$ such that reachability of σ' within at most n steps is decidable due to the decidability of $\mathrm{FOL}(\mathbb{R}, +, \cdot)$ [Tar48]. By successively testing increasing n, this does immediately yield a (well-known) semi-decision procedure for state reachability:

Lemma 1 (Semi-decidability of reachability). *It is semi-decidable whether a given state in a polynomial hybrid automaton is reachable.* □

Now it is straightforward to derive from the definition of $Reach^{\leq n}_{\sigma'}$ a predicate

$$Recurr^{\leq n} \stackrel{\text{def}}{=} \bigvee_{i\in\mathbb{N}_{\leq n}} \bigvee_{\sigma\in\Sigma} \bigvee_{\sigma'\in\Sigma} \bigvee_{1\leq j\leq n-i} \begin{pmatrix} initial_\sigma[\overleftarrow{\mathbf{x}}/\mathbf{x}] \wedge \\ \phi^{2i+1}_{\sigma\to\sigma'} \wedge \\ \phi^{2j-1}_{\sigma'\to\sigma'}[\mathbf{x}/\overleftarrow{\mathbf{x}}] \end{pmatrix} \tag{2}$$

that formalizes whether a state recurrence is encountered within the first $n+1$ transitions, i.e. on a path of length at most $2n+1$. This predicate again being in $\mathrm{FOL}(\mathbb{R}, +, \cdot)$, we obtain a semi-decision procedure for state recurrence:

Lemma 2 (Semi-decidability of recurrence). *It is semi-decidable whether a given polynomial hybrid automaton has a recurrent state.* □

However, these semi-decision procedures do not generalize to decision procedures: state reachability is known to be undecidable even for hybrid automata featuring just two clocks and a single stop-watch and which are, furthermore, confined to a bounded state space [HKPV95]. Here, a clock is a continuous variable having constant slope 1 within any activity, and a stop-watch is a continuous variable alternating between slopes 0 and 1 only. Thus, undecidability applies already to hybrid automata with just three continuous variables x_1, x_2, x_3, and activity predicates of only the two forms $\exists\, \Delta t\,.\,(x_1 = \overleftarrow{x_1}) \wedge \bigwedge_{i=\{2,3\}}(x_i = \overleftarrow{x_i} + \Delta t)$ and $\exists\, \Delta t\,.\, \bigwedge_{i=\{1,2,3\}}(x_i = \overleftarrow{x_i} + \Delta t)$. In the sequel, we will pursue a construction that utilizes an even more restricted subclass of polynomial hybrid automata, namely that of parametric timed automata [AHV93], and that — beyond undecidability of state reachability — proves undecidability of state recurrence and of progress also. In parametric timed automata, the continuous variables are either clocks, i.e. have slope 1 in every state and may only be checked and/or reset to 0 in transitions, or parameters. Parameters have slope 0 throughout and may not be affected by transitions, thus remaining constant throughout.

Theorem 1 (Undecidability of dynamic properties). *It is in general undecidable whether a given polynomial hybrid automaton*

1. *can reach a given state (on* some *path),*
2. *has a recurrent state,*
3. *will on* all *its paths eventually (i.e., after a finite number of steps) reach a given state.*

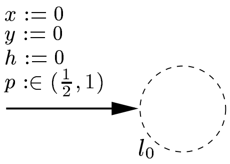
Property 3 is not even semi-decidable.
All these properties do still apply if we restrict attention to parametric timed automata with bounded range of the clock and parameter values instead of general polynomial hybrid automata.

Proof. We reduce the halting problem or — for property 3 of the theorem — boundedness problem of two-counter machines to the problem of testing whether some effectively constructed hybrid automaton (1) can reach a given state, (2) has a recurrent state, or (3) will on all its paths eventually reach a given state. The hybrid automaton A_M that we associate to a two-counter machine M uses four continuous variables x, y, h, and p which take values in the range $[0, 1]$, i.e. remain bounded along each run. x, y, and h are clocks in the sense of Alur and Dill [AD94], i.e. $\dot{x} = \dot{y} = \dot{h} = 1$ holds in all states. The remaining variable p is a parameter, i.e. is held constant. Hence, A_M is in fact a parametric timed automaton [AHV93] with three clocks x, y, h and one parameter p. Furthermore, A_M confines its variables to bounded range.

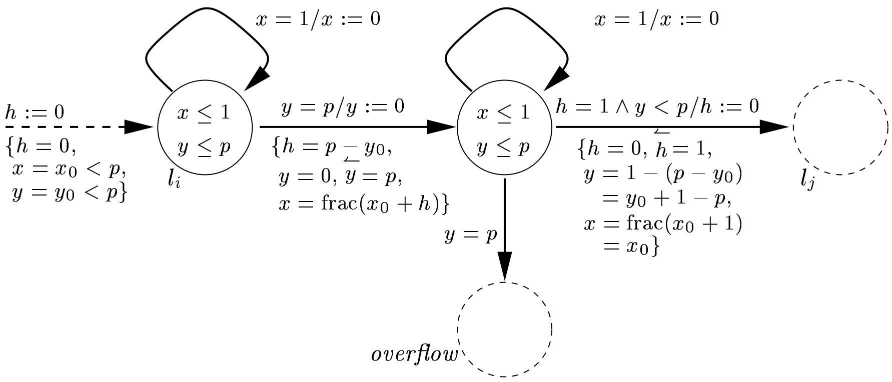
The encoding we use is that the two counter values n and m of a two-counter machine M are represented by the continuous variables x and y of A_M through the encoding $x = n(1 - p)$ and $y = m(1 - p)$. As the counter values n and m are in general unbounded, this would of course imply unbounded range of x and y. We confine x and y to bounded range by clipping their values to the interval $[0, p)$. Therefore, we will not perform increments if the resulting values of x or y would reach or exceed p, but send A_M to a special sink state *overflow* instead. During short transients, x and y may reach larger values, however, but even then their values will never exceed 1. Fig. 1 shows the details of the encoding.

Clipping the values of x and y to $[0, p)$ confines the representable integer range to $\left\{0, \ldots, \left\lceil \frac{p}{1-p} \right\rceil - 1\right\}$. But as parameter $p \in (\frac{1}{2}, 1)$ is chosen arbitrarily, every bounded run of M is accurately simulated by some run of A_M, namely one

Initialization: l_0 is the start location of M.



Increment: l_i : `increment`(y); `goto` l_j



Test and decrement: l_i : `if` $y = 0$ `then goto` l_j `else decrement`(y); `goto` l_k

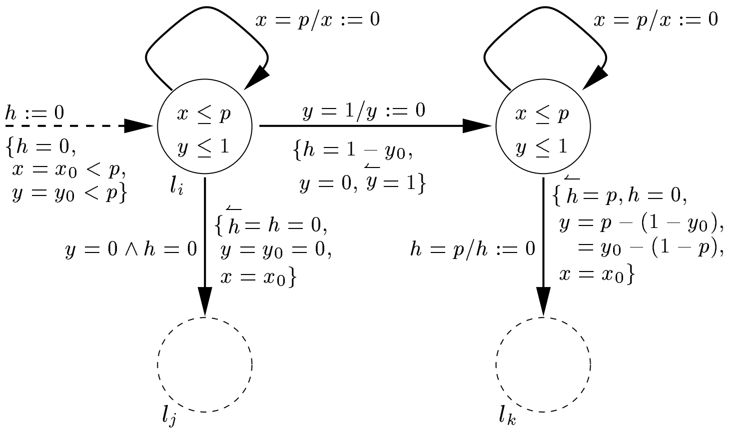


Fig. 1. Encoding the instructions of two-counter machines by parameterized timed automata. All variables except p are clocks, i.e. $\dot{x} = \dot{y} = \dot{h} = 1$ holds in all states (and has thus been omitted from the graphics). p is a parameter that is initially chosen and henceforth kept constant, i.e. $\dot{p} = 0$ holds everywhere. Annotations in curly brackets state key invariants that hold whenever the corresponding transitions are taken.

with p close enough to 1. On the other hand, runs that start with inadequate choice of p will inevitably end in the *overflow* state.

Using this encoding, location l_i of A_M is reachable iff M's computation reaches l_i. In particular, A_M's encoding of M's final location l_{stop} will be reachable in A_M iff M halts. As the halting problem for two-counter machines is undecidable, this implies undecidability of state reachability for bounded parametric timed automata. This proves undecidabililty of property (1) of the theorem.

Using a straightforward reduction of halting to state recurrence, this undecidability result can be generalized to recurrence properties, thus showing undecidability of (2). The essential steps are to make the terminal state l_{stop} of M recurrent and to ensure that all other states are non-recurrent. The latter can be done through the standard trick of adding a step counter to the two-counter machine that is simulated, thus — by counting up until the terminal state is reached — obtaining non-recurrence in the case of non-termination.

To deal with property (3) we observe that A_M will eventually reach state *overflow* on all its paths — i.e., independently of the particular choice of p — iff M's computation is unbounded. Hence, due to undecidability of boundedness of two-counter machines, it is in general undecidable whether some A_M will eventually reach state *overflow*. Consequently, it is undecidable whether a bounded parametric timed automaton will eventually reach a given state. I.e. property (3) is undecidable. Furthermore, this construction would yield a semi-decision method for unboundedness of two-counter machines if there were a semi-decision procedure for property (3). Thus, property (3) is not even semi-decidable. □

Note that Theorem 1 generalizes the undecidability result of Alur, Henzinger, and Vardi for parametric timed automata [AHV93] in two directions: first, it shows that undecidability of state reachability applies even when parameters and clocks are confined to bounded range, a case left open in [AHV93] where only the unbounded case was investigated. Second, we show that undecidability also holds for state recurrence and for progress of parametric timed automata.

3 Progress properties and state recurrence

According to Theorem 1, all interesting dynamic properties of hybrid automata are undecidable. Nevertheless, some prospects for mechanized analysis of hybrid automata remain, as state reachability and state recurrence are semi-decidable (Lemmata 1 and 2). However, at least to the practitioner, existence of recurrent states is hardly an interesting property in itself. It merely is a workhorse in the verification of finite-state systems, as there it is closely related to satisfaction of progress properties: for such systems, it can be used for disproving that on all possible computation paths, eventually something good will happen. For infinite-state systems, e.g. hybrid automata, there is no immediate correspondence between existence of infinite paths with (un-)desirable properties and existence of recurrent states with (un-)desirable properties, as with an infinity of states, infinite paths need not have recurrent states. Hence, the connection between state recurrence and progress properties for hybrid automata needs extra investigation, which is the theme of the current section. We start by formally defining satisfaction of a progress property by a hybrid automaton.

Definition 1 (Satisfaction of $\Diamond\phi$). *Let A be a deadlock-free polynomial hybrid automaton with $A = (\Sigma, \mathbf{x}, (act_\sigma)_{\sigma\in\Sigma}, (trans_{\sigma\to\sigma'})_{\sigma,\sigma'\in\Sigma}, (initial_\sigma)_{\sigma\in\Sigma})$, and let ϕ be an* $\mathrm{FOL}(\mathbb{R},+,\cdot)$*-predicate with* $\mathrm{free}(\phi) \subseteq \mathbf{x}$. *We say that A* satisfies "eventually ϕ" (denoted $A \models \Diamond\phi$) *iff $A_{\neg\phi}$ has no infinite path, where $A_{\neg\phi}$ denotes A confined to the portion of the state space satisfying $\neg\phi$.*
Formally, given an arbitrary $\mathrm{FOL}(\mathbb{R},+,\cdot)$*-predicate ψ with* $\mathrm{free}(\psi) \subseteq \mathbf{x}$, *we define $A_\psi \overset{\mathrm{def}}{=} (\Sigma, \mathbf{x}, (\widetilde{act}_\sigma)_{\sigma\in\Sigma}, (\widetilde{trans}_{\sigma\to\sigma'})_{\sigma,\sigma'\in\Sigma}, (\widetilde{initial}_\sigma)_{\sigma\in\Sigma})$, where $\widetilde{act}_\sigma \overset{\mathrm{def}}{=} act_\sigma \wedge \psi$, $\widetilde{trans}_{\sigma\to\sigma'} \overset{\mathrm{def}}{=} trans_{\sigma\to\sigma'} \wedge \psi$, and $\widetilde{initial}_\sigma \overset{\mathrm{def}}{=} initial_\sigma \wedge \psi$ for each $\sigma, \sigma' \in \Sigma$.*

Remark 1. Note that the restriction to deadlock-free automata is essential: a non-deadlock-free automaton A with a single state and no transitions would, along the lines of Definition 1, satisfy any progress property $\Diamond\phi$, because A and thus also $A_{\neg\phi}$ do not have paths longer than 2.

Yet, while the restriction to deadlock-free automata circumvents such pathologies, there remain subtle effects that are debatable. These have to do with the pragmatics of using hybrid automata as a model of real-time systems. Models of real-time systems are subject to peculiar effects like Zeno behaviour or indefinite idling in a state. Individual expectations on the impact of these peculiarities on satisfaction of progress properties may vary, and thus may or may not coincide with our definition.

While polynomial hybrid automata have no built-in model of time, there is a standard way of representing time, namely through representing the timed behaviour of a differential inequation $\dot{x} = 3$ by an activity $\exists\, \Delta t\,.\ \overleftarrow{x} + 3\cdot\Delta t = x\,.$ Consequently, the entity Δt occurring in activity predicates corresponds to the time spent in that activity. Given such an encoding, we will observe that Zeno behaviour can block progress according to Definition 1, as an automaton like that of Fig. 2 will violate a progress property even though it could spend only finite time in a Zeno behaviour before progress occurs.

A dual situation occurs when a discrete state of the hybrid automaton may be held indefinitely long without a transition being enforced within finite time, as in automaton B of Fig. 3. Deadlock-freedom ensures that there always remains some transition that will eventually be enabled, and one may adopt or refute the fairness assumption that under these circumstances, some transition will indeed eventually be taken. If the fairness assumption is adopted, B satisfies $\Diamond x \neq 0$, which is the view taken by Definition 1. The opposite view would be that B may stay in state σ forever, thus violating $\Diamond x \neq 0$. The later view differs from Definition 1 and would call for either a change of the definition or a transformation on the hybrid automaton under investigation to accommodate that view. The latter is, however, simple as any automaton that can stay indefinitely long in some state may be augmented with extra transitions that may be taken during such idling phases. The standard way is to add transitions that become ripe periodically, thus not introducing Zenoness (which might prohibit other transitions

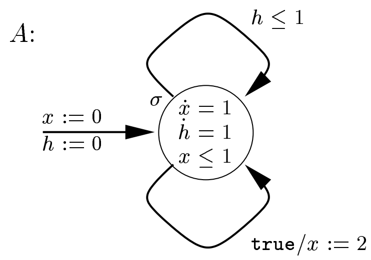


Fig. 2. Zeno behaviour may block progress: A, if formalized by the predicates $initial_\sigma = (x = 0 \wedge h = 0)$, $act_\sigma = (\exists \Delta t\,.\ \overleftarrow{x} + 1 \cdot \Delta t = x \wedge \overleftarrow{h} + 1 \cdot \Delta t = h \ \wedge\ x \leq 1)$, and $trans_{\sigma \to \sigma} = (\overleftarrow{h} \leq 1 \wedge \overleftarrow{h} = h \wedge \overleftarrow{x} = x \ \vee\ x = 2 \wedge \overleftarrow{h} = h)$, does not satisfy $\Diamond x = 2$.

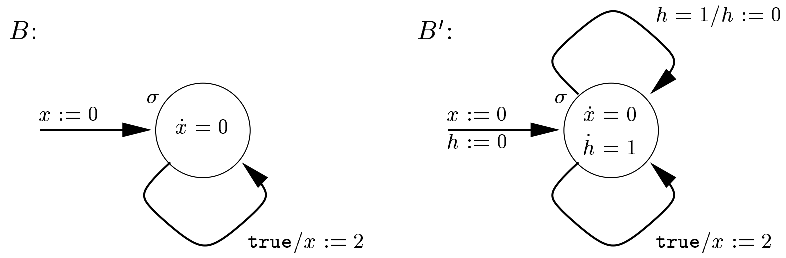


Fig. 3. Weak fairness implied by Definition 1: Automaton B — formally denoting $act_\sigma = (\exists \Delta t\,.\ \overleftarrow{x} + 0 \cdot \Delta t = x)$, $trans_{\sigma \to \sigma} = (x = 2)$, and $initial_\sigma = (x = 0)$ — satisfies $\Diamond x \neq 0$. I.e., it is assumed to eventually take a transition. If such fairness is not desired then its implications can be masked by adding clocks and transitions which become ripe periodically, as in automaton B'. B' violates $\Diamond x \neq 0$.

from contributing to progress), yet satisfying the implied fairness. An example of this transformation can be found in automaton B' of Fig. 3. •

According to Definition 1, verifying a progress property of the form $\Diamond \phi$ amounts to proving non-existence of an infinite path in $A_{\neg\phi}$, i.e. non-existence of an infinite path invariantly satisfying $\neg\phi$. In finite-state systems, this reduces to showing that there is no recurrent state on a $\neg\phi$-path. However, in infinite-state systems, infinite paths need not be recurrent. Thus, the semi-decision method of Lemma 2 does not even yield a semi-decision method for falsifying $\Diamond \phi$.

Nevertheless, there is a close relationship between state recurrence and progress if we deal with hybrid automata that are subject to disturbances due to noise. To demonstrate this, we will adopt the model of noise introduced in [Frä99]. However, this very simple notion — to be repeated in the next section — merely serves as a vehicle for illustration. We conjecture that the remainder of the theory holds true for a broader range of reasonable models of noise.

3.1 A simple model of noise

We begin by repeating the formalization of a simple model of noise in hybrid automata that was first put forward in [Frä99], where it was analyzed for its impact on decidability of reachability (i.e., safety) properties. Within this model we assume that noise will leave the discrete state set and transitions unaffected, but will make the evolution phases more nondeterministic. I.e., given a hybrid automaton

$$A = (\Sigma, \mathbf{x}, (act_\sigma)_{\sigma\in\Sigma}, (trans_{\sigma\to\sigma'})_{\sigma,\sigma'\in\Sigma}, (initial_\sigma)_{\sigma\in\Sigma}) \ ,$$

a *disturbed variant* of A is any hybrid automaton

$$\tilde{A} = (\Sigma, \mathbf{x}, (\widetilde{act}_\sigma)_{\sigma\in\Sigma}, (trans_{\sigma\to\sigma'})_{\sigma,\sigma'\in\Sigma}, (initial_\sigma)_{\sigma\in\Sigma})$$

with $act_\sigma \Rightarrow \widetilde{act}_\sigma$ for each $\sigma \in \Sigma$.

Now assume that the continuous state space comes equipped with a first-order definable metrics *dist*, with $dist(\mathbf{x},\mathbf{y})$ being its definition in FOL$(\mathbb{R}, +, \cdot)$. Furthermore, let ε be a first-order definable constant in $\mathbb{R}_{>0}$. We say that a disturbance $\tilde{A}$ of A is a *disturbance of noise level* ε *or more* wrt. *dist* iff

$$\exists \mathbf{y} \,.\, \big(act_\sigma[\mathbf{y}/\mathbf{x}] \wedge dist(\mathbf{x},\mathbf{y}) < \varepsilon\big) \Rightarrow \widetilde{act}_\sigma \tag{3}$$

for each $\sigma \in \Sigma$. Thus, evolution phases can drift away within at least an open ball of radius ε under such disturbances, as depicted in Fig. 4. It seems reasonable to assume that within any realistic noise field such disturbances exist for some sufficiently small ε.[5]

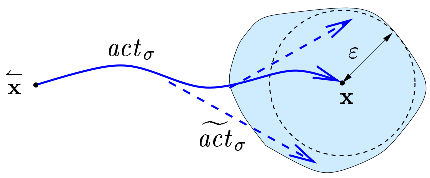


Fig. 4. Noise of level ε or more: if $\mathbf{x}$ is reachable from $\overleftarrow{\mathbf{x}}$ under some activity act_σ, then the corresponding disturbed activity $\widetilde{act}_\sigma$ can reach a superset of the open ball of radius ε around x.

[5] Concerning the range of applicability of the model, it is worth noting that for the theory exposed in the remainder, the fact that *every* activity is subject to drift of at least ε for some $\varepsilon > 0$ is not essential. While we have chosen such a model in order to simplify exposition, the theory itself can be easily extended to the more general situation that only every cycle in the discrete state space of the hybrid automaton (i.e. every alternating sequence of activities and transitions going from some discrete state σ via other discrete states back to σ) contains at least one activity with drift of at least ε.

Obviously, a disturbance will yield additional possible behaviours, possibly leading to otherwise unreachable states. It turns out that in case of existence of infinite paths, some of these noisily reachable states are always recurrent.

3.2 Infinite paths and noisy state recurrence

We will in the remainder concentrate on progress properties of the form $\Diamond\neg\phi$ with ϕ being a bounded property, where an FOL($\mathbb{R}, +, \cdot$)-predicate ϕ with free(ϕ) $\subseteq \mathbf{x}$ is called *bounded* iff $\{\mathbf{y} \in \mathbb{R}^d \mid I_{\mathbf{x}:=\mathbf{y}} \models \phi\}$ is bounded.[6] I.e., the requirements formulae we want to verify demand that all possible trajectories of the hybrid automaton will eventually leave some bounded region. Note that the restriction to bounded properties actually poses no restriction in the quite common case that the reachable state space of the hybrid automaton is itself bounded.

Lemma 3. *Let ϕ be a bounded property and A a polynomial hybrid automaton. If A does not satisfy $\Diamond\neg\phi$ — i.e., if A_ϕ has an infinite path — then for any disturbance $\tilde{A}$ of positive noise level, $\tilde{A}_\phi$ has a recurrent state.*

Proof. Assume that A violates $\Diamond\neg\phi$ and that $\pi = \langle(\sigma_0, \mathbf{y}_0), (\sigma_1, \mathbf{y}_1), \ldots\rangle$ is a witness path for violation of $\Diamond\neg\phi$, i.e. an infinite path of A_ϕ. Let $\sigma \in \Sigma$ be a discrete state that is visited infinitely often in π and let $\pi_\sigma \stackrel{\text{def}}{=} \{\mathbf{y}_i \mid \sigma_i = \sigma,\ i \text{ is odd}\}$.

If π_σ is finite then π itself revisits some hybrid state infinitely often, which then also is a recurrent state of any disturbance $\tilde{A}$. Otherwise, as ϕ is bounded, the Bolzano-Weierstrass theorem, generalized to $\mathbb{R}^d$, implies that π_σ has an accumulation point $\mathbf{y} \in \mathbb{R}^d$. Consequently, for arbitrary $\varepsilon > 0$ there are infinitely many points $(\sigma_j, \mathbf{y}_j)$ in the witness path π such that j is odd and $\sigma_j = \sigma$ and $\mathit{dist}(\mathbf{y}_j, \mathbf{y}) < \frac{\varepsilon}{2}$. Any such point is recurrent in any disturbance of noise level ε or more, as for each such j there is an odd $k > j$ with $\sigma_k = \sigma$ and $\mathit{dist}(\mathbf{y}_k, \mathbf{y}) < \frac{\varepsilon}{2}$, which implies that $(\sigma_j, \mathbf{y}_j)$ is a direct successor state of $(\sigma_{k-1}, \mathbf{y}_{k-1})$ in a disturbance $\tilde{A}_\phi$ of noise level ε or more, because $\mathit{dist}(\mathbf{y}_j, \mathbf{y}_k) \leq \mathit{dist}(\mathbf{y}_j, \mathbf{y}) + \mathit{dist}(\mathbf{y}_k, \mathbf{y}) < \varepsilon$. $\square$

Lemma 3 shows that existence of an infinite path in A_ϕ implies existence of a recurrent state in any disturbance of positive noise level. The converse implication holds if we add some constraint on the viable paths in a disturbed automaton. Up to now, we have defined disturbance in terms of hybrid automata, i.e. infinite-state transition systems without acceptance conditions. We may, however, add certain acceptance conditions in order to rule out certain undesirable limit behaviours. E.g., for a given polynomial hybrid automaton $A = (\Sigma, \mathbf{x}, (\mathit{act}_\sigma)_{\sigma\in\Sigma}, (\mathit{trans}_{\sigma\to\sigma'})_{\sigma,\sigma'\in\Sigma}), (\mathit{initial}_\sigma)_{\sigma\in\Sigma})$ and given $\varepsilon > 0$ consider its ε-disturbance $A^\varepsilon \stackrel{\text{def}}{=} (\Sigma, \mathbf{x}, (\widetilde{\mathit{act}}_\sigma)_{\sigma\in\Sigma}, (\mathit{trans}_{\sigma\to\sigma'})_{\sigma,\sigma'\in\Sigma}), (\mathit{initial}_\sigma)_{\sigma\in\Sigma})$ with $\widetilde{\mathit{act}}_\sigma \stackrel{\text{def}}{=} \exists\, \mathbf{y}\, .\, \mathit{act}_\sigma[\mathbf{y}/\mathbf{x}] \wedge \mathit{dist}(\mathbf{x}, \mathbf{y}) < \varepsilon$: here, all disturbances occurring along a path are *strictly less* than ε; yet, the supremum of disturbances may well equal

[6] Note that the choice of I does not matter in $\{\mathbf{y} \in \mathbb{R}^d \mid I_{\mathbf{x}:=\mathbf{y}} \models \phi\}$, as free($\phi$) $\subseteq \mathbf{x}$.

ε. We may avoid this by adding a further constraint on the viable paths, stating that a path is considered non-viable iff the supremum of the disturbances along the path equals ε. I.e., an infinite path $\langle(\sigma_0, \mathbf{y}_0), (\sigma_1, \mathbf{y}_1), \ldots\rangle$ of A^ε is *viable* iff there is $\delta < \varepsilon$ such that

$$I_{\overleftarrow{\mathbf{x}}:=\mathbf{y}_i, \mathbf{x}:=\mathbf{y}_{i+1}} \models \exists\, \mathbf{y}\, .\, act_{\sigma_i}[\mathbf{y}/\mathbf{x}] \wedge dist(\mathbf{x}, \mathbf{y}) \leq \delta$$

for each even i. In the remainder, we will denote A^ε restricted to its viable paths, i.e. augmented with above path-acceptance condition, by $A^{\sup<\varepsilon}$. From the definition it is obvious that $Paths(A^{\sup<\varepsilon}) = \bigcup_{\delta<\varepsilon} Paths(A^\delta)$, where $Paths(\cdot)$ denotes the set of possible paths of the respective automaton.

Lemma 4 (Progress vs. recurrence). *Let A be a polynomial hybrid automaton, ϕ bounded, and $\varepsilon > 0$. Then $A_\phi^{\sup<\varepsilon}$ has an infinite path iff $A_\phi^{\sup<\varepsilon}$ has a recurrent state, i.e. recurrence and existence of infinite paths are related in exactly the same way as for finite automata.*

Proof. Existence of a recurrent state trivially implies existence of an infinite path. For the converse inclusion note that $Paths(A_\phi^{\sup<\varepsilon}) = \bigcup_{\delta<\varepsilon} Paths(A_\phi^\delta)$. Hence, $A_\phi^{\sup<\varepsilon}$ has an infinite path iff some A_ϕ^δ for some $\delta < \varepsilon$ has. But then, according to Lemma 3, $A_\phi^{\frac{\delta+\varepsilon}{2}}$ has a recurrent path. Consequently, as $\frac{\delta+\varepsilon}{2} < \varepsilon$, $A_\phi^{\sup<\varepsilon}$ has a recurrent path. □

4 An automatic verification procedure for progress

While Lemmata 3 and 4 provide a connection between state recurrence and progress, they are not of much help for effectively falsifying $\Diamond\neg\phi$, as they relate progress to existence of recurrent states within *arbitrary* disturbances of strictly positive noise level — an infinite set of disturbances that could not be probed in finite time. Fortunately, Lemma 3 can be strengthened:

Lemma 5 (Recurrence vs. path boundedness). *Let ϕ be a bounded property and A a polynomial hybrid automaton. If A_ϕ has paths of arbitrary length — i.e., if there is no* finite *bound on path length in A_ϕ — then for any disturbance $\tilde{A}$ of positive noise level, $\tilde{A}_\phi$ has a recurrent state.*

Proof. The proof closely resembles that of Lemma 3. Instead of the infinite witness path π we just have to take, for given $\varepsilon > 0$, a sufficiently long path π of A_ϕ such that π contains two distinct odd indices k and j that satisfy $\sigma_k = \sigma_j$ and $dist(y_j, y_k) < \varepsilon$. Such a path exists due to unboundedness of path length in A_ϕ and boundedness of ϕ. □

Note that Lemma 5 is stronger than Lemma 3, as non-existence of a finite bound on path length does *not* imply existence of an infinite path in an infinite-state system.

Lemma 5 is a valuable stepping stone towards a semi-decision procedure due to the fact that its premise (that A_ϕ has paths of arbitrary finite length) is co-semidecidable:

Lemma 6 (Semi-decidability of path-boundedness). *It is semi-decidable whether a polynomial hybrid automaton has a finite bound on its path length.*

Proof. For arbitrary $n \in \mathbb{N}$, the $\mathrm{FOL}(\mathbb{R}, +, \cdot)$ formula

$$Path^n \stackrel{\text{def}}{=} \bigvee_{\sigma \in \Sigma} \bigvee_{\sigma' \in \Sigma} initial_\sigma[\overleftarrow{\mathbf{x}} / \mathbf{x}] \wedge \phi^n_{\sigma \to \sigma'}$$

is satisfiable iff the automaton under investigation has a path of length n. Using a decision method for $\mathrm{FOL}(\mathbb{R}, +, \cdot)$ we can check satisfiability of $Path^n$ for successively larger n until $Path^n$ is found to be unsatisfiable. If this ever happens then we can terminate the semi-decision method, having found a witness for A having finitely bounded path length. $\square$

Now, Lemma 5 shows that Lemmata 2 and 6 are "almost complementary" in the sense that cases where none of the two semi-decision procedures matches are isolated: they are "borderline" in the sense that they do not feature recurrent ϕ-paths, yet any disturbance of *arbitrarily small* positive noise level has recurrent ϕ-paths, as the next theorem shows. Above notion of "borderline" is captured by the following definition:

Definition 2 (Fragility and robustness). *We say that A is* fragile wrt. recurrence of ϕ-states *iff A_ϕ has no recurrent state, yet for any disturbance $\tilde{A}$ of positive noise level, $\tilde{A}_\phi$ has a recurrent state.*

If A is not fragile wrt. recurrence of ϕ-states, i.e. if it either has recurrent ϕ-states itself or has disturbances without recurrent ϕ-states, then A is called robust wrt. recurrence of ϕ-states.

Using this definition and the semi-decision procedures of Lemmata 2 and 6, it is straightforward to show our main theorem:

Theorem 2 (Robustness implies decidability). *Let A be a polynomial hybrid automaton and ϕ a bounded property. If A is robust wrt. recurrence of ϕ-states then it is decidable whether $A \models \Diamond\neg\phi$.*

Proof. As A is robust wrt. recurrence of ϕ-states, there are two cases: either A_ϕ has a recurrent state — which is semi-decidable according to Lemma 2 — or there is $\delta > 0$ such that $(A^\varepsilon)_\phi$ has no recurrent state for each $\varepsilon < \delta$. In the latter case, A_ϕ has finitely bounded path length according to Lemma 5, which is semi-decidable due to Lemma 6. Consequently, we may start the semi-decision procedures of Lemmata 2 and 6 in parallel and can be sure that at least one of the two semi-decision procedures terminates on A_ϕ.

If the procedure of Lemma 2 terminates then A_ϕ has a recurrent state, which implies $A \not\models \Diamond\neg\phi$. Vice versa, if the procedure of Lemma 6 terminates then A_ϕ has finitely bounded path length, which is equivalent to $A \models \Diamond\neg\phi$ according to Definition 1. Hence, the parallel execution of the semi-decision procedures of Lemmata 2 and 6 decides satisfaction of progress properties for robust automata. $\square$

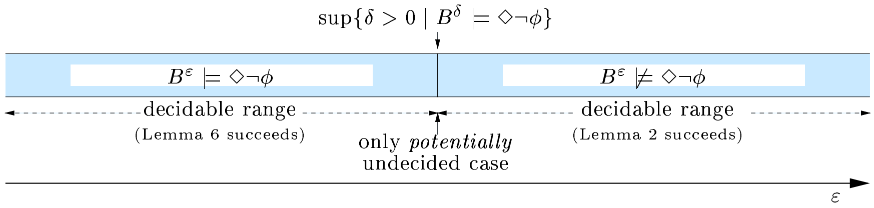


Fig. 5. The semi-decision procedure of Cor. 1 may only fail on a single noise level

Now, even though it is intuitively clear that fragility with respect to recurrence of ϕ-states is a rare borderline case, as for a fragile automaton even the slightest disturbance causes a drastic change in the recurrence pattern, it might still seem a dirty trick to have a verification method for progress with a premise concerning recurrence. However, we can replace this premise by a premise mentioning not being borderline with respect to progress instead, provided that the automaton A is itself a disturbed variant B^ε of another polynomial hybrid automaton B:

Corollary 1. *If $A = B^\varepsilon$ for some polynomial hybrid automaton B and some noise level $\varepsilon > 0$, and if ϕ is bounded then $A \models \Diamond\neg\phi$ is decidable unless $\varepsilon = \sup\{\delta > 0 \mid B^\delta \models \Diamond\neg\phi\}$, i.e. unless the noise level of A exactly matches the borderline between satisfaction and violation of $\Diamond\neg\phi$. This fact is illustrated in Figure 5.*

Proof. If $\varepsilon > \sup\{\delta > 0 \mid B^\delta \models \Diamond\neg\phi\}$ then $A_\phi = (B^\varepsilon)_\phi$ has a recurrent state due to Lemma 3 and thus A is robust wrt. recurrence of ϕ-states. If, on the other hand, $\varepsilon < \sup\{\delta > 0 \mid B^\delta \models \Diamond\neg\phi\}$ then — by Definition of $B^\delta \models \Diamond\neg\phi$ — all $(A^\gamma)_\phi$ in the range $\varepsilon \leq \gamma < \sup\{\delta > 0 \mid B^\delta \models \Diamond\neg\phi\}$ do not have recurrent states, and consequently, A is robust wrt. recurrence of ϕ-states. Hence, A is robust wrt. recurrence of ϕ-states whenever $\varepsilon \neq \sup\{\delta > 0 \mid B^\delta \models \Diamond\neg\phi\}$. Thus, the result follows from Theorem 2. □

Corollary 1 provides an almost complete automatic verification method for progress properties of noisy hybrid automata (i.e., automata of the form $A = B^\varepsilon$ for some polynomial hybrid automaton B and some $\varepsilon > 0$) which may only fail if we happen to test *exactly* the borderline case (i.e., $\varepsilon = \sup\{\delta > 0 \mid B^\delta \models \Diamond\neg\phi\}$).

By Lemma 4 and semi-decidability of state recurrence (Lemma 2), the result becomes even a little bit stronger if we restrict interest to viable paths of B^ε only, i.e. if we take $A = B^{\sup<\varepsilon}$:

Corollary 2. *If $A = B^{\sup<\varepsilon}$ for some polynomial hybrid automaton B and some $\varepsilon > 0$, and if ϕ is bounded then*

1. *$A \not\models \Diamond\neg\phi$ is semi-decidable, and*
2. *$A \models \Diamond\neg\phi$ is semi-decidable unless $\varepsilon = \sup\{\delta > 0 \mid B^\delta \models \Diamond\neg\phi\}$.*

Proof. If $A \not\models \Diamond\neg\phi$ then $A_\phi = B_\phi^{\sup<\varepsilon}$ has a recurrent state according to Lemma 4. Thus, $A \not\models \Diamond\neg\phi$ is semi-decidable by the procedure of Lemma 2.

$A \models \Diamond\neg\phi$, on the other hand, can be checked by the same technique as in the previous corollary. Thus, $A \models \Diamond\neg\phi$ is semi-decidable unless $\varepsilon = \sup\{\delta > 0 \mid B^\delta \models \Diamond\neg\phi\}$. □

I.e., if $A = B^{\sup<\varepsilon}$ for some noise level $\varepsilon > 0$ then falsification of $\Diamond\neg\phi$ always succeeds (if applicable) and it is only verification that may fail if we happen to test *exactly* the borderline case.

5 Decidability of robustness

Given the fact that *robust* systems can be automatically verified, it remains the question whether robustness can be automatically detected.

Corollary 3 (Undecidability of robustness). *It is in general undecidable whether a hybrid automaton is robust or fragile wrt. recurrence of ϕ-states for a given bounded predicate ϕ. Robustness is, however, semi-decidable.*

Proof. All fragile systems do not have recurrent ϕ-states due to Definition 2. Hence, existence of recurrent ϕ-states of a hybrid automaton is trivially decidable once it is known to be fragile. For robust automata, on the other hand, existence of recurrent ϕ-states is decidable, as they either have recurrent ϕ-states (which is semi-decidable due to Lemma 2) or they have finitely bounded path length according to Lemma 5 (which is semi-decidable due to Lemma 6). Consequently, decidability of being robust would imply decidability of state recurrence for polynomial hybrid automata with bounded reachable state space (to see this, just take as ϕ an arbitrary bounded over-approximation of the reachable state space). However, state recurrence is undecidable even for parametric timed automata with finitely bounded state space (cf. Theorem 1), and therefore robustness wrt. recurrence of ϕ-states is undecidable.

However, robustness of A wrt. recurrence of ϕ-states is semi-decidable, as A is robust iff either A_ϕ has a recurrent state, which can be detected by the semi-decision procedure of Lemma 2, or has disturbances of strictly positive noise level without recurrent ϕ-states. According to Lemma 5, the latter implies existence of a disturbance $\tilde{A}$ of strictly positive noise level with finitely bounded path length. But then, for some sufficiently large $n \in \mathbb{N}$, $A^{\frac{1}{n}}$ will also have finitely bounded path length. We can search for such n using the semi-decision method for finitely bounded path length of Lemma 6. □

The undecidability result, while disappointing, should nevertheless be no serious obstacle in practice. It is good engineering practice to make systems tolerant against noise. Thus, any well-engineered system should be robust s.t. the decision procedure of Theorem 2 comes to an answer. In particular, no need for checking robustness should arise in practice.

It should furthermore be noted that the *known* undecidable cases are "ideal" hybrid automata A in the sense that A itself is not a disturbance of positive noise level of any other hybrid automaton, but follows ideal trajectories instead. It is currently unknown whether the verification method of Corollaries 1 and 2 may indeed fail on some automata $A = B^{\varepsilon}$ (or $A = B^{\sup<\varepsilon}$, resp.) with $\varepsilon > 0$, or whether it will in fact always succeed on such "noisy" automata.

6 Discussion

We have been able to show that progress properties $\Diamond\neg\phi$ of linear and polynomial hybrid automata can be decided algorithmically in most practically interesting cases. The remaining cases are such that non-existence of recurrent states satisfying ϕ depends on the *complete* absence of noise.

We have called such systems, where the recurrence pattern of ϕ-states does not change abruptly when an infinitesimally small amount of noise is added, *robust with respect to recurrence of ϕ-states*. It should be noted that our notion of robustness mentions ϕ and thus is with respect to the particular progress property under investigation. This distinguishes our approach from the closely related approaches to robust satisfaction of properties by Puri [Pur98] and Asarin & Bouajjani [AB01], which furthermore are currently confined to state reachability (i.e., safety), while our approach is available for both safety properties [Frä99] and progress properties (this paper).

The advantage of defining robustness relative to the property under investigation is that this yields a notion of robustness which in contrast to the other notion applies to some hybrid automata with non-recursive reachability relation, hence broadening the range of applicability of the method, which is guaranteed to succeed on *every* robust automaton (see Theorem 2 and, for safety properties, [Frä99, Theorem 1]). In detail, the different notions of robustness in the approaches [Frä99] and [AB01] dealing with safety of hybrid automata are as follows: Asarin and Bouajjani [AB01] define that a system is robust iff its undisturbed reachability relation equals the intersection of all perturbed reachability relations arising from disturbance levels $\varepsilon > 0$. I.e., they consider an automaton A to be robust iff the reachable state space of A's ε-disturbances in the limit for ε tending to 0 exactly equals A's reachable state space. This tight approximation of A's reach set by its disturbances implies that A's reach set is recursive according to [AB01, Corollary 8], i.e. recursiveness of the reach set is a necessary condition of robustness. In contrast, we define robustness in [Frä99] with respect to the safety property $\Box\phi$ to be verified: there, an automaton A is considered to be robust iff either it is unsafe (i.e., $A \not\models \Box\phi$) or it has disturbances of positive noise level which are safe (i.e., $A \models \Box\phi$ for some $\varepsilon > 0$). Clearly, this notion does not have recursiveness of the reach set as a necessary condition. It should, however, be noted that the notion of robustness of [Frä99] is not strictly weaker than that of [AB01]; the two notions are in general incomparable.

As noise is a general phenomenon in continuous systems, the effect of noise on the computational power of different kinds of analog computation has been extensively studied, e.g. by Maass and Sontag in [MS97] and by Moore in [Moo98].

Their findings are not directly comparable to ours because of the quite simplistic, qualitative model of noise used in our approach and the probabilistic models used there. We do agree that a realistic model of noise should better be quantitative, representing noise as a probabilistic process. Within this article we have deliberately refrained from this, as such a model would hardly yield any practical verification procedure due to the large computational overhead caused by calculating the density function when iterating the probabilistic transition and evolution steps. Yet, the results obtained in [MS97] with respect to noise of a wide support, meaning that the system may be driven to arbitrary continuous state with some low probability, indicate that the question posed in the end of Section 5, namely whether undecidability may still prevail if the automaton A is of the form $A = B^{\varepsilon}$ or $A = B^{\sup<\varepsilon}$, is a viable one.

Finally, it should be noted that a fundamentally different approach to modeling noisy behaviour of timed or hybrid automata has been proposed by Gupta, Henzinger, and Jagadeesan [GHJ97]. In that line of work, the idealized behavioural model of timed automata is essentially kept and only filtered a posteriori by removing accepted and adding rejected trajectories that are isolated with respect to some topology on trajectories. However, it seems that this model has less favorable consequences with respect to decidability issues than the model of Puri [Pur98], which is more akin to our approach in that it models the impact of noise (i.e. clock drift, as only timed automata are dealt with) as a widening of the transition relation and defines the reachable states as those that are reachable under arbitrarily small positive noise.

Acknowledgements

Eugene Asarin and Ahmed Bouajjanni deserve special thanks for their kind hospitality which I enjoyed during a visit to Grenoble this spring. The many fruitful discussions we had helped in clarifying the relation between our different notions of robustness. Furthermore, I would like to thank the anonymous referees for their exceptionally detailed comments.

References

[AB01] Eugene Asarin and Ahmed Bouajjani. Perturbed turing machines and hybrid systems. In *Proceedings of the Sixteenth Annual IEEE Symposium on Logic in Computer Science (LICS 2001)*. IEEE, 2001.

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[AHS96] Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors. *Hybrid Systems III — Verification and Control*, volume 1066 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. Parametric real-time reasoning. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 592–601, 1993.

[AKNS95] P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors. *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995.

[Frä99] Martin Fränzle. Analysis of hybrid systems: An ounce of realism can save an infinity of states. In Jörg Flum and Mario Rodríguez-Artalejo, editors, *Computer Science Logic (CSL'99)*, volume 1683 of *Lecture Notes in Computer Science*, pages 126–140. Springer-Verlag, 1999.

[GHJ97] Vineet Gupta, Thomas A. Henzinger, and Radha Jagadeesan. Robust timed automata. In Oded Maler, editor, *Proceedings of the First International Workshop on Hybrid and Real-Time Systems (HART 97)*, volume 1201 of *Lecture Notes in Computer Science*, pages 331–345. Springer-Verlag, 1997.

[GNRR93] Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors. *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.

[HHWT95] Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. HyTech: The next generation. In *16th Annual IEEE Real-time Systems Symposium (RTSS 1995)*, pages 56–65. IEEE Computer Society Press, 1995.

[HKPV95] Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, pages 373–382. ACM, 1995.

[Moo98] Chris Moore. Finite-dimensional analog computers: Flows, maps, and recurrent neural networks. In C.S. Calude, J. Casti, and M.J. Dinneen, editors, *1st International Conference on Unconventional Models of Computation*. Springer-Verlag, 1998.

[MS97] Wolfgang Maass and Eduardo D. Sontag. Analog neural nets with Gaussian or other common noise distributions cannot recognize arbitrary regular languages. In *Electronic Colloquium on Computational Complexity, technical reports*. 1997.

[Pur98] Anuj Puri. Dynamical properties of timed automata. In A. P. Ravn and H. Rischel, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'98)*, volume 1486 of *Lecture Notes in Computer Science*, pages 210–227. Springer-Verlag, 1998.

[Tar48] Alfred Tarski. A decision method for elementary algebra and geometry. RAND Corporation, Santa Monica, Calif., 1948.