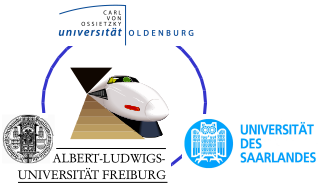

AVACS – Automatic Verification and Analysis of Complex Systems

REPORTS

of SFB/TR 14 AVACS

Editors: Board of SFB/TR 14 AVACS

PTIME parametric verification of safety properties for reasonable linear hybrid automata

by

Werner Damm Carsten Ihlemann Viorica Sofronie-Stokkermans

AVACS Technical Report No. 70
August 2011
ISSN: 1860-9821

Publisher: Sonderforschungsbereich/Transregio 14 AVACS
(Automatic Verification and Analysis of Complex Systems)
Editors: Bernd Becker, Werner Damm, Bernd Finkbeiner, Martin Fränzle,
Ernst-Rüdiger Olderog, Andreas Podelski
ATRs (AVACS Technical Reports) are freely downloadable from `www.avacs.org`

Author(s) contact: Werner Damm (`wd@avacs.org`).

PTIME parametric verification of safety properties for reasonable linear hybrid automata

Werner Damm[1], Carsten Ihlemann[2], and Viorica Sofronie-Stokkermans[3]

[1] Carl von Ossietzky University Oldenburg, Oldenburg, Germany damm@offis.de
[2] Max-Planck-Institut für Informatik, Saarbrücken, Germany ihlemann@mpi-inf.mpg.de
[3] Max-Planck-Institut für Informatik, Saarbrücken, Germany sofronie@mpi-inf.mpg.de

Abstract. This paper identifies an industrially relevant class of linear hybrid automata (LHA) called *reasonable LHA* for which parametric verification of convex safety properties with exhaustive entry states can be verified in polynomial time and time-bounded reachability can be decided in nondeterministic polynomial time for non-parametric verification and in exponential time for parametric verification. Properties with exhaustive entry states are restricted to runs originating in a (specified) inner envelope of some mode-invariant. Deciding whether an LHA is reasonable is shown to be decidable in polynomial time.

1 Introduction

Hybrid systems are at the heart of almost any safety critical systems built today. Whether it is in designing a new advanced driver assistance system, controlling production plants for chemical processes, or providing envelope protection in primary flight control: the capability to express the influence of discrete mode changes on continuous dynamics is a fundamental prerequisite for modeling such applications. In typical industrial design flows, Matlab-Simulink/Stateflow models are used for the specification of the underlying controllers, and safety and stability are verified informally, using simulation and testing. Often, the implementation of such controllers is based on code generated automatically from such specification models, employing tools such as the Embedded Encoder or Target Link. For use in safety critical applications, modeling guidelines are imposed to ease testability and verification, such as discussed below.

Giving such specification models a formal semantics in terms of hybrid automata, such as in [2, 34], makes them amenable to formal verification techniques. The challenge in making these applicable in practice is in identifying industrially relevant classes of hybrid automata, for which verification of safety properties is truly feasible, that is, where safety properties can be decided in polynomial time. This paper proposes *reasonable linear hybrid automata* as such a model: we show that for reasonable LHA

- safety properties can be decided in polynomial time
- time-bounded reachability properties can be checked in nondeterministic polynomial time

provided that the entry conditions cover the inner envelopes of the modes. For the parametric verification of such properties we identify conditions under which the complexity is in PTIME, NP, or EXPTIME. Finally, we show that the property of *being reasonable* can be decided in polynomial time.

Linear hybrid automata [3] allow for linear constraints on the values of the continuous variables (resp. on their rate of growth) of the form $\sum_{i=1}^{n} a_i x_i \leq a$ (resp. $\sum_{i=1}^{n} c_i \dot{x}_i \leq c$). As in [3], we assume that guards and invariants are given as conjunctions of linear constraints, and that updates are expressed in linear arithmetic. This class of hybrid automata has been shown to be expressive enough to give sufficiently concise conservative abstractions for non-linear hybrid systems (see e.g. [14]) as naturally resulting from Matlab-Simulink models, and comes with symbolic semi-decision procedures, such as originally proposed in [13] and subsequently often refined, e.g. in [35, 16, 11, 8]. We propose a novel subclass of LHA which we call *reasonable* for which a particular class of safety properties can be decided, in fact in polynomial time. First, we require each mode additionally to

be equipped with what is often called an inner-envelope of the mode invariant. Intuitively, these serve to assure chatter-freedom, that is to assure each state is visted with a minimal dwelling time. We then restrict ourselves to what we call safety properties with exhaustive entry conditions of the form:

$$\Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Box\phi_{\mathsf{safe}})$$

and timed-reachability properties with exhaustive entry conditions:

$$\Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Diamond_{\leq t}\phi_{\mathsf{safe}})$$

where ϕ_{ExhEntry} is the disjunction of all inner envelopes of the modes, and both ϕ_{safe} and the inner envelopes are given by convex linear constraints. Our *reasonable LHA* exploit modeling guidelines used in safety critical system designs, enforcing the following restrictions on LHA:

Input determinism Guards of transitions originating from one and the same state are mutually exclusive, and initial states of modes are disjoint.

Invariant compatability Mode invariants are safe (i.e. are all subsets of ϕ_{safe}). Moreover, whenever the invariants becomes false, then there is at least one transition leaving the mode whose guard is enabled.

Chatter-freedom All transitions enter a mode within its inner envelope. Moreover, there is a minimal dwelling time s.t. when a mode is entered in its inner envelope, no guard of a transition leaving the mode will be enabled before the minimal dwelling time is expired.

Our approach is based on a reduction of such verification problems to checking satisfiability of formulae in fragments of linear arithmetic, which can be decided in polynomial time for verification of safety properties, and in NP for time-bounded reachability properties. This holds as well for parametric verification if the parameters occur only as bounds within linear constraints (such as in those for mode-invariants, for guards, jumps, in ϕ_{safe}). We also study more general parametric verification problems, where first these bounds can be expressed by concave resp. convex functions (for lower resp. upper bounds), and show again that even for this class parametric verification of safety properties with exhaustive entry states can be decided in polynomial time. When parameters appear in the terms of constraints, the resulting non-linearity yields an exponential complexity.

As a running example, we use a simple chemical process control, where substances to be mixed must maintain constant ratios within margins. Here ϕ_{safe} encodes that indeed the ratio of concentrations is maintained, and that the tank is never overflowing. As an example of a time bounded reachability property, we are interested in proving that the tank is drained within 200 ms if the critical ratio of concentrations is exceeded. We have used the SMT solver Z3 to verify the above safety and time-bounded reachability properties within fragments of a second.

Related work. A considerable amount of work was dedicated to identifying classes of hybrid automata for which checking safety is decidable. Reachability and safety in LHA are in general undecidable, while invariant checking and bounded reachability are decidable. One of the first decidability results for reachability is for initialized rectangular hybrid automata [13], a restricted class of rectangular hybrid automata (i.e. hybrid automata for which at each control location the flow is described by differential inclusions of the form $\dot{x} \in [a, b]$) which require resets for continous variables upon mode transitions, unless the newly entered and left mode share the same dynamics. These results have been extended in [22] which identifies classes of LHA for which reachability is decidable and in [21] which uses translations of verification problems into satisfiability problems for theories of real numbers (possibly with exponentiation). Decidability of *finite-precision* hybrid automata was studied in [1]. In [20, 4, 5], o-minimal hybrid systems are studied; it is shown that for such hybrid systems reachability can be reduced to reachability in finite models due to existence of finite bismulations – this is used to give decidability results. The restrictions imposed when defining o-minimal hybrid systems – in particular the requirement that only constant resets are allowed – are quite severe. In contrast, the restrictions we impose (mainly invariant compatibility and chatter freedom) are much more likely to be met in applications. Parametric verification of linear hybrid automata was first studied in [3]. In [26, 27] a dynamic hybrid logic is developed allowing non-linear differential equations within modalities, and possibilities of deriving constraints

on parameters guaranteeing satisfaction of system requirements are discussed. While [26, 27] support significantly richer classes of dynamics, the underlying logic is highly undecidable, and there is no investigation into decidable subclasses of the verification problem. We improve on previous work by Wang [35] in that our approach gives an effective procedure, while his back-reachability based symbolic execution approach is not guaranteed to terminate. [10] incrementally learns constraints on good parameters from counterexamples; this approach is only terminating if there are only finitely many counterexamples. In [10] parameters are not allowed as bounds for derivatives, and system variables and parameters cannot be combined multiplicatively. In our work, we allow this, at the price of passing from linear constraints to non-linear constraints. Some first steps towards using reasoning in theory extensions for the verification of parametric systems and also for inferring constraints on the parameters which ensure safety were taken in [30]. In this paper we extend these results; however, the results are here presented without referring to (local) theory extensions whenever this is possible (an exception is the verification of parametric systems with functions as parameters, where locality of theory extensions is explicitly mentioned).

This paper is a considerably extended version of [6]: on the one hand, it contains full proofs which could not be included in [6]. In addition, we here consider a general form of parametric LHA, in which instead of allowing parametric constants as bounds we also allow functional parametric bounds (which vary in time) in the constraints on the values of the continuous variables, i.e. dynamics of the form

$$\forall t \quad f_1(t) \leq \sum a_i x_i(t) \leq f_2(t) \quad \text{and} \quad \forall t \quad \sum b_i \dot{x}_i(t) \leq b,$$

where f_1 is a convex function and f_2 is a concave function (either concrete or parametric).

Although it is known that invariant checking and bounded model checking for linear hybrid automata are decidable, no precise complexity results exist in the literature. It is also very important to identify classes of linear hybrid automata for which safety checking and time-bounded reachability are decidable and have low complexity. To the best of our knowledge, no systematic study exists in which classes of properties which can be checked in PTIME are identified, and conditions on LHA which allow for efficient checking of safety properties and/or time-bounded reachability are given. This is the aim of our paper.

Structure of the paper. The paper is structured as follows. In Sect. 2 we present the main definitions; we introduce our running example; and then present the complexity results for fragments of linear real arithmetic which will be used in the paper. In Sect. 3 we show that invariant checking is decidable in polynomial time, and bounded model checking is in general decidable in nondeterministically polynomial time, also for parametric systems in which the parameters are described as variables, if the constraints on parameters are expressed by linear inequalities. We then consider more general parametric systems, in which the parameters are functional. We analyze the complexity of generating constraints on parameters. In Sect. 4 we introduce reasonable hybrid automata and show that this property can be checked in polynomial time. In Sect. 5 we consider more general safety properties and bounded time reachability and show that for reasonable (extended) linear hybrid automata checking safety properties can be reduced to invariant checking and checking time-bounded reachability properties can be reduced to problems very similar to bounded model checking. Sect. 6 contains a summary of the results contained in the paper, and Sect. 7 contains some experimental results.

2 Preliminaries

We start by defining the classes of hybrid automata which are studied in this paper (linear hybrid automata, parametric linear hybrid automata, and extended (parametric) linear hybrid automata), we present the main verification problems we will study and then summarize some complexity results for fragments of linear real arithmetic.

2.1 Hybrid automata

Definition 21 *A hybrid automaton (HA) is a tuple*

$$S = (X, Q, \mathsf{flow}, \mathsf{Inv}, \mathsf{Init}, E, \mathsf{jump})$$

consisting of:

(1) A finite set $X = \{x_1, \ldots, x_n\}$ of real valued variables and a finite set Q of control modes, that together define the state space of the system.
(2) A family $\{\mathsf{flow}_q \mid q \in Q\}$ specifying the continuous dynamics in each control mode, where flow_q is a predicate over the variables in $X \cup \dot{X}$, where $\dot{X} = \{\dot{x}_1, \ldots, \dot{x}_n\}$, where $\dot{x}_i$ is the derivative of x_i.
(3) A family $\{\mathsf{Inv}_q \mid q \in Q\}$ defining the invariant conditions for each control mode, where for every $q \in Q$, Inv_q is a predicate over the variables in X.
(4) A family $\{\mathsf{Init}_q \mid q \in Q\}$ defining the initial states for each control mode, where for every $q \in Q$, Init_q is a predicate over the variables in X.
(5) The control switches, modeled by a finite multiset E with elements in $Q \times Q$. Every $(q, q') \in E$ is a directed edge between q (source mode) and q' (target mode).
(6) A family of guards for every control switch $\{\mathsf{guard}_e \mid e \in E\}$, where guard_e is a predicate over X.
(7) A family of jump conditions $\{\mathsf{jump}_e \mid e \in E\}$, where jump_e is a predicate over $X \cup X'$, where $X' = \{x'_1, \ldots, x'_n\}$ is a copy of X consisting of "primed" variables.

Definition 22 *An HA is* input-deterministic *if:*

(i) $\mathsf{Init}_{q_1} \wedge \mathsf{Init}_{q_2} \models \bot$ *for all* $q_1 \neq q_2, q_1, q_2 \in Q$
(ii) $\mathsf{guard}_{(q,q_1)} \wedge \mathsf{guard}_{(q,q_2)} \models \bot$ *for all* $(q, q_1), (q, q_2) \in E, q_1 \neq q_2$.

At any time instant, the state of a hybrid automaton specifies a control location and values for all data variables.

Definition 23 *A* state *of the hybrid automaton S is a pair (q, a) consisting of a control mode $q \in Q$ and a vector $a = (a_1, \ldots, a_n)$ that represents a value $a_i \in \mathbb{R}$ for each variable $x_i \in X$. A state (q, a) is* admissible *if Inv_q is true when each x_i is replaced by a_i. A state (q, a) is* initial *if Init_q is true when each x_i is replaced by a_i.*

There are the following types of state change:

Jump The state can change by an instantaneous transition that changes the control location and the values of data variables according to the jump conditions, or

Flow The state can change due to the evolution in a given control mode over an interval of time: the values of the data variables change in a continuous manner according to the flow rules of the current control location.

Definition 24 *A* run *of the hybrid automaton S is a finite sequence $s_0 s_1 \ldots s_k$ of admissible states s_j such that*

- *the first state s_0 is an initial state of S,*
- *each pair (s_j, s_{j+1}) of consecutive states in the sequence is either a jump of S or a flow of S.*

Linear hybrid automata An atomic linear predicate is a linear inequality (strict or non-strict inequality between a rational constant and a linear combination of variables with rational coefficients, such as $3x_1 - x_2 + 7x_5 \leq 4$). A convex linear predicate is a finite conjunction of linear inequalities.

Definition 25 *[3] A hybrid automaton S is a linear hybrid automaton (LHA) if it satisfies the following two requirements:*

1. **Linearity** *For every control mode $q \in Q$, the flow condition flow_q, the invariant condition Inv_q, and the initial condition Init_q are convex linear predicates. For every control switch $e = (q, q') \in E$, the jump condition jump_e and the guard guard_e are convex linear predicates. In addition, we assume that the flow conditions flow_q are conjunctions of* non-strict *inequalities.*
2. **Flow independence** *For every control mode $q \in Q$, the flow condition flow_q is a predicate over the variables in $\dot{X}$ only (and does not contain any variables from X). This requirement ensures that the possible flows are independent from the values of the variables, and depend only on the control mode.*

2.2 Example

We illustrate these definitions by means of a simplified chemical plant example.

Example 1 *We consider the following (very simplified) chemical plant example, modeling the situation in which we control the reaction of two substances, and the separation of the substance produced by the reaction: Let x_1, x_2 and x_3 be variables which describe the evolution of the volume of substances 1 and 2, and the substance 3 generated from their reaction, respectively. Assume that $\epsilon_a > 0, \delta_a > 0, L_{\mathsf{overflow}} > L_f > 0, \mathsf{max}, \mathsf{min} > 0, \mathsf{dmin} > 0$ are constants. (These are typical candidates for parameters in parametric verification.) The plant is described by a hybrid automaton with four modes:*

Mode 1: Fill. *In this mode the temperature is low, and hence the substances 1 and 2 do not react. In this mode the substances 1 and 2 (possibly mixed with a very small quantity of substance 3) are filled in the tank in equal quantities up to a certain margin of error. This is described by the following invariants and flow conditions:*

$\mathsf{Inv}_1 \quad x_1 + x_2 + x_3 \leq L_f \ \wedge \ \bigwedge_{i=1}^{3} x_i \geq 0 \ \wedge \ -\epsilon_a \leq x_1 - x_2 \leq \epsilon_a \ \wedge \ 0 \leq x_3 \leq \mathsf{min}$

$\mathsf{flow}_1 \ \dot{x_1} \geq \mathsf{dmin} \wedge \dot{x_2} \geq \mathsf{dmin} \wedge \dot{x_3} = 0 \wedge -\delta_a \leq \dot{x_1} - \dot{x_2} \leq \delta_a$

*If the proportion is not kept the system jumps into mode 4 (***Dump***); if the total quantity of substances exceeds level L_f (tank filled) the system jumps into mode 2 (***React***).*

Mode 2: React. *In this mode the temperature is high, and the substances 1 and 2 react. The reaction consumes equal quantities of substances* 1 *and* 2 *and produces substance* 3*.*

$\mathsf{Inv}_2 \quad L_f \leq x_1 + x_2 + x_3 \leq L_{\mathsf{overflow}} \ \wedge \ \bigwedge_{i=1}^{3} x_i \geq 0 \ \wedge \ -\epsilon_a \leq x_1 - x_2 \leq \epsilon_a \ \wedge \ 0 \leq x_3 \leq \mathsf{max}$

$\mathsf{flow}_2 \ \dot{x_1} \leq -\mathsf{dmin} \wedge \dot{x_2} \leq -\mathsf{dmin} \wedge \dot{x_3} \geq \mathsf{dmin} \wedge \dot{x_1} = \dot{x_2} \wedge \dot{x_3} + \dot{x_1} + \dot{x_2} = 0$

*If the proportion between substances 1 and 2 is not kept the system jumps into mode 4 (***Dump***); if the total quantity of substances 1 and 2 is below some minimal level* min *the system jumps into mode 3 (***Filter***).*

Mode 3: Filter. *In this mode the temperature is low again and substance 3 is filtered out.*

$\mathsf{Inv}_3 \quad x_1 + x_2 + x_3 \leq L_{\mathsf{overflow}} \ \wedge \ \bigwedge_{i=1}^{3} x_i \geq 0 \wedge -\epsilon_a \leq x_1 - x_2 \leq \epsilon_a \ \wedge \ x_3 \geq \mathsf{min}$

$\mathsf{flow}_3 \ \dot{x_1} = 0 \wedge \dot{x_2} = 0 \wedge \dot{x_3} \leq -\mathsf{dmin}$

*If the proportion between substances 1 and 2 is not kept the system jumps into mode 4 (***Dump***). Otherwise, if the concentration of substance 3 is below some minimal level* min *the system jumps into mode 1 (***Fill***).*

Mode 4: Dump. *In this mode the content of the tank is emptied. For simplicity we assume that this happens instantaneously, i.e.* $\mathsf{Inv}_4 : \bigwedge_{i=1}^{3} x_i = 0$ *and* $\mathsf{flow}_4 : \bigwedge_{i=1}^{3} \dot{x_i} = 0$.

Jumps. *The automaton has the following jumps:*

- $e = (1,2)$ *with* $\mathsf{guard}_e(x_1, x_2, x_3) = x_1 + x_2 + x_3 \geq L_f$ *which leaves the variables unchanged.*
- $e = (2,3)$ *with* $\mathsf{guard}_e(x_1, x_2, x_3) = x_1 + x_2 \leq \mathsf{min}$ *which leaves the variables unchanged.*
- $e = (3,1)$ *with* $\mathsf{guard}_e(x_1, x_2, x_3) = -\epsilon_a \leq x_1 - x_2 \leq \epsilon_a \wedge 0 \leq x_3 \leq \mathsf{min}$ *which leaves the variables unchanged.*
- *Two edges* e_1^1, e_2^1 *from* 1 *to* 4*, and two edges* e_1^2, e_2^2 *from* 2 *to* 4*, with*
  $\mathsf{guard}_{e_1^j}(x_1, x_2, x_3) = x_1 - x_2 \geq \epsilon_a$,
  $\mathsf{guard}_{e_2^j}(x_1, x_2, x_3) = x_1 - x_2 \leq -\epsilon_a$*; and*
  $\mathsf{jump}_{e_i^j}(x_1, x_2, x_3, x_1', x_2', x_3') = \bigwedge_{i=1}^{3} x_i' = 0, (j, i = 1, 2)$*;*
- *Two edges* e_1^3, e_2^3 *from* 3 *to* 4*, with*
  $\mathsf{guard}_{e_1^3}(x_1, x_2, x_3) = x_1 - x_2 \geq \epsilon_a$*;*
  $\mathsf{guard}_{e_2^3}(x_1, x_2, x_3) = x_1 - x_2 \leq -\epsilon_a$*, and*
  $\mathsf{jump}_{e_i^3}(x_1, x_2, x_3, x_1', x_2', x_3') = \bigwedge_{i=1}^{3} x_i' = 0$ *for* $i = 1, 2$.

2.3 Extended hybrid automata

In this paper we will also consider extended hybrid automata (EHA), which are hybrid automata in which in addition a set of "inner envelopes" of modes is specified.

Definition 26 *An extended hybrid automaton is a tuple of the form:*

$$S = (X, Q, \mathsf{flow}, \mathsf{Inv}, \mathsf{InEnv}, \mathsf{Init}, E, \mathsf{jump}),$$

with the property that $(X, Q, \mathsf{flow}, \mathsf{Inv}, \mathsf{Init}, E, \mathsf{jump})$ *is a hybrid automaton and for every mode* $q \in Q$ *also an inner envelope* InEnv_q *is specified such that:*

- *The initial states are contained in the inner envelope, i.e.*

$$\models \forall x_1, \ldots, x_n (\mathsf{Init}_q(x_1, \ldots, x_n) \rightarrow \mathsf{InEnv}_q(x_1, \ldots, x_n)).$$

- *The inner envelope is "strictly contained" in the mode invariant, i.e. it satisfies the following conditions:*

$$\models \forall x_1, \ldots, x_n (\mathsf{InEnv}_q(x_1, \ldots, x_n) \rightarrow \mathsf{Inv}_q(x_1, \ldots, x_n)).$$
$$\not\models \forall x_1, \ldots, x_n (\mathsf{InEnv}_q(x_1, \ldots, x_n) \leftrightarrow \mathsf{Inv}_q(x_1, \ldots, x_n)).$$

Definition 27 *An* extended linear hybrid automaton *(ELHA) is a LHA which is also an EHA in which each* InEnv_q *can be described as a conjunction of linear inequalities (as a convex predicate).*

Example 2 *An inner envelope for mode 1 in Sect. 2.2 could for instance be:*

$$\begin{aligned} \mathsf{InEnv}_1(x_1, x_2, x_3) = \; & 0 \leq x_1 + x_2 + x_3 \leq L_{\mathsf{safe}} \wedge \textstyle\bigwedge_{i=1}^{3} x_i \geq 0 \\ & \wedge -\epsilon_{\mathsf{safe}} \leq x_1 - x_2 \leq \epsilon_{\mathsf{safe}} \;\wedge\; 0 \leq x_3 \leq \mathsf{min} \end{aligned}$$

where $0 < L_{\mathsf{safe}} < L_f$ *and* $0 < \epsilon_{\mathsf{safe}} < \epsilon_a$.

The following property will play an important rôle:

Definition 28 *Let* (s_1, s_2) *be a jump in* S *under mode switch* (q, q')*, where* $s_1 = (q, x_1, \ldots, x_n), s_2 = (q', x_1', \ldots, x_n')$*. We say that the jump* (s_1, s_2) *is* mode reachable *(w.r.t.* q*) if there exists a state* $s_0 = (q, x_1^0, \ldots, x_n^0)$ *such that* $\mathsf{InEnv}_q(x_1^0, \ldots, x_n^0)$ *and there exists a flow in mode* q *from* s_0 *to* s_1.

Thus transitions which are mode reachable (w.r.t. q) can always also be reached by an evolution from some state in the inner envelope of q.

2.4 Parametric linear hybrid automata

Parametric linear hybrid automata (PLHA) are parametric descriptions of linear hybrid automata, in which some of the coefficients or bounds in the linear inequalities describing mode invariants, guards, jumps, or in the description of dynamics are parameters in a set P. We allow an even more general form of parametricity, in which the bounds in state invariants, guards and jump conditions can be expressed using functions with certain properties. Such parametric descriptions of bounds are useful for instance in situations in which we want to verify systems which have non-linear behavior.

- In a parametric linear hybrid automaton without functions as parameters, any convex constraint $\sum_{i=1}^{n} a_i x_i \leq a$ (resp. $\sum_{i=1}^{n} a_i \dot{x}_i \leq a$) in the flow condition flow_q, the invariant condition Inv_q, the initial condition Init_q and the guard guard_e, as well for any convex linear constraint $\sum_{i=1}^{n} b_i x_i + c_i x_i' \leq d$ in $\mathsf{jump}(t, t')$, some of the coefficients a_i, b_i, c_i or the bounds a, d are allowed to be parametric.
- In the more general parametric hybrid systems in which we allow functional parameters as bounds, the mode invariant conditions contain generalized boundedness properties of form:

$$\phi(t) \rightarrow f_1(t) \leq \sum a_i x_i(t) \leq f_2(t)$$

  where $\phi(t)$ is a formula (in the language of linear real arithmetic) with free variable t, f_1 is a convex function and f_2 is a concave function depending on t only. Here, f_1 and f_2 can be (possibly non-linear) term-defined functions, or parametric functions which need to satisfy the additional convexity/concavity conditions.

We use the following axiomatizations $\mathsf{Conv}^I(f)$ (resp $\mathsf{Conc}^I(f)$) of the convexity (resp. concavity) conditions (for an interval I of the form $(-\infty, a], [a, b]$ or $[b, \infty)$):

$$\mathsf{Conv}^I(f) \;\; \forall x, y, z \left(x{\in}I \wedge y \in I \wedge x < z \leq y \rightarrow \frac{f(z){-}f(x)}{z{-}x} \leq \frac{f(y){-}f(x)}{y{-}x} \right).$$

$$\mathsf{Conc}^I(f) \;\; \forall x, y, z \left(x{\in}I \wedge y \in I \wedge x < z \leq y \rightarrow \frac{f(z){-}f(x)}{z{-}x} \geq \frac{f(y){-}f(x)}{y{-}x} \right).$$

Example 3 *We consider the following situations:*

- Parametric functional bounds on the values of the variables:
  $\forall t(0 \leq t \rightarrow (f_1(t) \leq x(t) \leq f_2(t)))$, f_1, f_2 *uninterpreted but satisfying the convexity/concavity conditions mentioned above.*
- Concrete bounds on the values of the variables:
  $\forall(0 \leq t \leq 1 \rightarrow (t^2 \leq x(t) \leq 4 - t^3))$.

In order to encompass the situations above we consider the class of (parametric) extended linear hybrid automata.

Definition 29 *Parametric linear hybrid automata (PLHA) are defined as linear hybrid automata, for which a set $P = P_c \cup P_f$ of parameters is specified (consisting of parametric constants P_c and parametric functions P_f) with the difference that for every control mode $q \in Q$ and every mode switch e:*

- *the flow condition* $\mathsf{flow}(q)$*, the invariant condition* $\mathsf{Inv}(q)$*, the initial condition* $\mathsf{Init}(q)$ *and the guard condition* guard_e *are convex* extended *linear predicates of the form*

$$\sum_{i=1}^{n} a_i x_i \leq f \qquad \textit{resp.} \qquad \sum_{i=1}^{n} b_i \dot{x}_i \leq b$$

- *the convex linear constraint in* $\mathsf{jump}(t, t')$ *is of the form*

$$\sum_{i=1}^{n} b_i x_i + c_i x_i' \leq d$$

where the coefficients a_i, b_i, c_i and the bounds b, d are either numerical constants or parametric constants in P_c; and f is either a constant, or a parametric constant in P_c, or an uninterpreted function (parameter in P_f) satisfying the concavity condition, or a concrete concave function.

The flow independence conditions hold as in the case of linear hybrid automata.

Definition 210 *Parametric extended linear hybrid automata (PELHA) are defined analogously, by additionally imposing that the inner envelopes are described by convex extended linear predicates of the form*

$$\sum_{i=1}^{n} d_i x_i \leq g$$

where the coefficients d_i are either numerical constants or parametric constants in P_c; and g is either a constant, or a parametric constant in P_c, or an uninterpreted function (parameter in P_f) satisfying the convexity condition, or a concrete convex function.

2.5 Verification Problems

We consider the following verification problems:

Invariant checking is the problem of checking whether a quantifier-free formula Ψ in real arithmetic over the variables X is invariant (under jumps and flows) in a linear hybrid automaton S, i.e.:

(1) $\mathsf{Init}_q \models \Psi$ for all $q \in Q$;
(2) Ψ is invariant under jumps and flows:

Flow For every flow in a mode q, the continuous variables satisfy Ψ both during and at the end of the flow.

Jump For every jump according to a control switch e, if the values of the continuous variables satisfy Ψ before the jump, they satisfy Ψ after the jump.

Bounded model checking is the problem of checking whether a formula Ψ is preserved under runs of length bounded by k, i.e.:

(1) $\mathsf{Init}_q \models \Psi$ for every $q \in Q$;
(2) Ψ is preserved under runs of length j for all $1 \leq j \leq k$.

Safety properties. We consider properties of the form:

$$\phi = \Box(\phi_{\mathsf{entry}} \rightarrow \Box\phi_{\mathsf{safe}})$$

stating that for every run in an automaton S, if the predicate ϕ_{entry} holds at the beginning of the run, then ϕ_{safe} is always true during the run.

Checking time-bounded reachability problems, i.e. checking properties stating that for every run σ in the automaton S, if ϕ_{entry} holds at the beginning of the run, then ϕ_{safe} becomes true in run σ at latest at time t, i.e. of the form:

$$\phi = \Box(\phi_{\mathsf{entry}} \rightarrow \diamond_{\leq t}\phi_{\mathsf{safe}}).$$

Assumption. The safety and time-bounded reachability conditions we analyze in the paper are assumed to have *exhaustive entry conditions* (i.e. $\phi_{\mathsf{entry}} = \phi_{\mathsf{ExhEntry}} := \bigvee_{q \in Q} \mathsf{InEnv}_q$). Through this paper we will assume that the invariance resp. safety properties Ψ and ϕ_{safe} to be checked are convex linear predicates over X.

2.6 Complexity results for linear arithmetic

We now summarize some existing results on the complexity of satisfiability checking for fragments of linear arithmetic which we will use in the paper:

Theorem 211 (Linear arithmetic) *The following hold:*

(1) The satisfiability of any conjunction of (strict and non-strict) linear inequalities can be checked in PTIME [17, 33].
(2) The complexity of checking the satisfiability of sets of clauses in linear arithmetic is in NP [31].

We present some classes of sets of clauses in linear arithmetic for which satisfiability can be checked in PTIME.

Horn-disjunctive linear constraints. A Horn-disjunctive linear constraint (or HDL for short) is a disjunction $d_1 \vee \cdots \vee d_n$ where each $d_i, i = 1, \ldots, n$ is a linear inequality or a linear disequation, and the number of inequalities among $d_1, \ldots, d_n$ does not exceed one.

Theorem 212 ([18]) *The satisfiability of any conjunction C of Horn disjunctive linear constraints (over $\mathbb{R}$ or $\mathbb{Q}$) can be decided in PTIME. Moreover, we can eliminate n variables from a set C of Horn disjunctive linear constraints in time $O(|C|^n)$.* [4]

UTVPI$^{\neq}$ constraints. $UTVPI^{\neq}$ constraints (where $UTVPI^{\neq}$ is an abbreviation for "unit two variables per inequality with disequalities") are conjunctions of constraints of the form $ax + by \leq c$ or $ax + by \neq c$, with $a, b \in \{-1, 0, 1\}$. For this class satisfiability checking and variable elimination are much easier.

Theorem 213 ([19]) *The satisfiability of any set C of $UTVPI^{\neq}$ constraints (over $\mathbb{R}$ or $\mathbb{Q}$) can be decided in time $O(n^3 + d)$, where d is the number of disequations and n is the number of variables in C. Any number of variables can be eliminated in time $O(dn^4)$, where n and d are as before.*

Ord-Horn constraints. An Ord-Horn constraint is an implication of the form $\bigwedge_{i=1}^{n} x_i \leq y_i \rightarrow x \leq y$, where x_i, y_i, x, y are variables.

Theorem 214 ([24]) *The satisfiability of any conjunction of Ord-Horn constraints (over $\mathbb{R}$ or $\mathbb{Q}$) can be decided in PTIME.*

2.7 Notation

In what follows we use the following notation. If $x_1, \ldots, x_n$ are continuous variables we will sometimes denote the sequence $x_1, \ldots, x_n$ with $\overline{x}$ and the sequence $\dot{x}_1, \ldots, \dot{x}_n$ with $\overline{\dot{x}}$. We will denote the sequence of values $x_1(t), \ldots, x_n(t)$ of these variables at a time t with $\overline{x}(t)$. The same applies also for variables with superscripts: If $x_1^k, \ldots, x_n^k$ are continuous variables we will denote the sequence $x_1^k, \ldots, x_n^k$ with $\overline{x}^k$ and the sequence $\dot{x}_1^k, \ldots, \dot{x}_n^k$ with $\overline{\dot{x}}^k$.

3 Simple verification problems

We are interested in parameterized verification of safety properties of controllers specified using (extended) linear hybrid automata. The verification problems we consider in this section are invariant checking and bounded model checking. We show that invariant checking is decidable in polynomial time, and bounded model checking is in general decidable in nondeterministically polynomial time. We also analyze parametric systems and show that the complexity results above also

[4] As mentioned in [18], this is a PTIME result under the assumption that the number of variables n which are eliminated is fixed (then the degree of the polynomial is directly proportional to the number of variables to be eliminated).

hold for parametric systems if the constraints on parameters are expressed by linear inequalities. We also analyze the complexity of generating constraints on parameters.

In Section 5 we consider more general safety properties and bounded time reachability and show that for certain *reasonable* (extended) linear hybrid automata checking safety properties can be reduced to invariant checking and checking time-bounded reachability properties can be reduced to problems very similar to bounded model checking.

3.1 Complexity of invariant checking

We first consider the problem of checking whether a quantifier-free formula Ψ in real arithmetic over the variables $X = \{x_1, \ldots, x_n\}$ is invariant (under jumps and flows) in a linear hybrid automaton S, i.e.:

(a) $\models \forall \overline{x}(\mathsf{Init}_q(\overline{x}) \rightarrow \Psi(\overline{x}))$ for all $q \in Q$;
(b) Ψ is invariant under jumps and flows:
- **Flow** For every flow in a mode q, the continuous variables satisfy Ψ both during and at the end of the flow.
- **Jump** For every jump according to a control switch e, if the values of the continuous variables satisfy Ψ before the jump, they satisfy Ψ after the jump.

We can model these problems using hybrid transition constraints which specify the states, variables and function symbols whose values change over time; and where sets of states and transitions (jumps and flows) are specified using formulae in suitable signatures.

We first analyze this problem for non-parametric linear hybrid automata. We assume throughout the paper that:

$$\mathsf{Inv}_q = \bigwedge_{j=1}^{m_q}(\sum_{i=1}^{n} a_{ij}^q x_i \lhd_j a_j^q), \quad \mathsf{flow}_q = \bigwedge_{j=1}^{n_q}(\sum_{i=1}^{n} c_{ij}^q \dot{x}_i \leq c_j^q), \quad \mathsf{guard}_e = \bigwedge_{j=1}^{m_e}(\sum_{i=1}^{n} g_{ij}^e x_i \lhd_j g_j^e),$$
$$\lhd_j \in \{\leq, <\} \qquad \text{where } e = (q, q') \in E, \lhd_j \in \{\leq, <\}$$

(and similarly for Init_q). These constraints can also be expressed referring to the time moment t as:

$$\mathsf{Inv}_q(x_1(t), \ldots, x_n(t)) = \textstyle\bigwedge_{j=1}^{m_q}(\sum_{i=1}^{n} a_{ij}^q x_i(t) \lhd_j a_j^q)$$

$$\mathsf{flow}_q(t) = \textstyle\bigwedge_{j=1}^{n_q}(\sum_{i=1}^{n} c_{ij}^q \dot{x}_i(t) \leq_j c_j^q)$$

and similarly for guard_e and Init_q. We will now express the flow conditions $\mathsf{flow}_q(t)$ without referring to the values of the derivative, using the following formulae (where $0 \leq t \leq t'$):

$$\underline{\mathsf{flow}}_q^{\overline{x}}(t, t') = \bigwedge_{j=1}^{n_q}(\sum_{i=1}^{n} c_{ij}^q (x_i(t') - x_i(t)) \leq_j c_j^q (t' - t)).$$

Theorem 31 will show that for LHA no precision is lost with this alternative axiomatization.

Notation. In what follows, if the variables $\overline{x} = x_1, \ldots, x_n$ are clear from the context we omit the upper index and write simply $\underline{\mathsf{flow}}_q(t, t')$; however, the index will be used e.g. in Theorem 37.

We axiomatize a flow in control mode q in the time interval $[t_0, t_1]$ (where $0 \leq t_0 \leq t_1$) as follows:

$$\mathsf{Flow}_q(t_0, t_1) = \forall t(t_0 \leq t \leq t_1 \rightarrow \mathsf{Inv}_q(\overline{x}(t))) \wedge \forall t, t'(t_0 \leq t < t' \leq t_1 \rightarrow \underline{\mathsf{flow}}_q(t, t')).$$

Assume that the jump update jump_e and guard guard_e at moment t for the control switch $e = (q, q') \in E$ are expressed by the following convex linear predicates:

$$\mathsf{guard}_e(\overline{x}(t)) = \textstyle\bigwedge_{j=1}^{m_e}(\sum_{i=1}^{n} g_{ij}^e x_i(t) \leq g_j^e)$$
$$\mathsf{jump}_e(\overline{x}(t), \overline{x}'(0)) = \textstyle\bigwedge_{j=1}^{n_e}(\sum_{i=1}^{n} b_{ij}^e x_i(t) + c_{ij}^e x_i'(0) \leq d_j^e).$$

For $e = (q, q') \in E$, we axiomatize the jump condition by

$$\mathsf{Jump}_e(\overline{x}, \overline{x}') := \mathsf{guard}_e(\overline{x}) \wedge \mathsf{jump}_e(\overline{x}, \overline{x}').$$

We want to analyze whether certain properties are preserved during jumps and flows. Let Ψ be a convex formula over the variables X. Ψ is preserved under a flow from time t_0 to t (where $0 \leq t_0 < t$) in state q iff the following formula is unsatisfiable:

$$\mathsf{Inv}_q(\overline{x}(t_0)) \wedge \Psi(\overline{x}(t_0)) \wedge \mathsf{Flow}_q(t_0, t) \wedge \neg\Psi(\overline{x}(t)).$$

Note that this is a satisfiability problem for a formula with free variables t_0 and t (implicitly existentially quantified) with the subformula $\mathsf{Flow}_q(t_0, t)$ containing a universal quantifier, i.e. for a formula containing an alternation of quantifiers of the form $\exists\forall$. In what follows we will show that in certain cases a simpler encoding is possible.

An optimized translation. We now present a considerably simpler encoding of the runs and safe runs in linear hybrid automata, in which $\mathsf{Flow}_q(t_0, t_1)$ is replaced with:

$$\mathsf{Inv}_q(x(t_0)) \wedge \mathsf{Inv}_q(x(t_1)) \wedge \underline{\mathsf{flow}}_q(t_0, t_1).$$

In Theorem 31 we will show that for linear hybrid automata precision is not lost after this replacement.

Theorem 31 *The following are equivalent for any LHA:*

(1) Ψ is an invariant of the hybrid automaton;
(2) For every $q \in Q$ and $e = (q, q') \in E$, the following formulae are unsatisfiable:

$$F_{\mathsf{Flow}}(q) \qquad \Psi(\overline{x}(t_0)) \wedge \mathsf{Flow}_q(t_0, t) \wedge \neg\Psi(\overline{x}(t)) \wedge t \geq t_0$$
$$F_{\mathsf{jump}}(e) \qquad \Psi(\overline{x}(t)) \wedge \mathsf{Jump}_e(\overline{x}(t), \overline{x}'(0)) \wedge \mathsf{Inv}_{q'}(\overline{x}'(0)) \wedge \neg\Psi(\overline{x}'(0))$$

(3) For every $q \in Q$ and $e = (q, q') \in E$, the following formulae are unsatisfiable:

$$F_{\mathsf{flow}}(q) \qquad \Psi(\overline{x}(t_0)) \wedge \mathsf{Inv}_q(\overline{x}(t_0)) \wedge \underline{\mathsf{flow}}_q(t_0, t) \wedge \mathsf{Inv}_q(\overline{x}(t)) \wedge \neg\Psi(\overline{x}(t)) \wedge t \geq t_0$$
$$F_{\mathsf{jump}}(e) \qquad \Psi(\overline{x}(t)) \wedge \mathsf{Jump}_e(\overline{x}(t), \overline{x}'(0)) \wedge \mathsf{Inv}_{q'}(\overline{x}'(0)) \wedge \neg\Psi(\overline{x}'(0))$$

Proof: [5] By definition, Ψ is an invariant of the hybrid automaton S iff Ψ is preserved under jumps and flows. It is easy to see that Ψ is preserved under jumps iff for every $e \in E$ the formula $F_{\mathsf{jump}}(e)$ is unsatisfiable. Ψ is preserved under flows iff there exists no state q, no interval $[t_0, t]$ with $t_0 \leq t$, no functions $x_1, \ldots, x_n$ (modeling the evolution in time of the continuous variables in X), such that the automaton is in mode q during the flow and the change of the values of $x_1, \ldots, x_n$ satisfies the flow conditions at every point during the flow, i.e., either $t_0 = t_1$ or otherwise $t_0 < t$ and for every $t' \in [t_0, t]$, $\bigwedge_{j=1}^{n_q}(\sum_{i=1}^{n} c_{ij}^q \dot{x}_i(t') \leq c_j^q)$, and such that Ψ holds at moment t_0 and does not hold at moment t.

To prove (1) $\Rightarrow$ (2), assume that the formula $F_{\mathsf{Flow}}(q)$ is satisfiable. Then there exist functions $x_1, \ldots, x_n$, and a time interval $[t_0, t]$ with $t_0 \leq t$ such that

- at t_0 the automaton is in mode q and $\Psi(\overline{x}(t_0))$ holds,
- S remains in state q during the whole interval $[t_0, t]$,
- $\Psi(\overline{x}(t))$ does not hold, and
- either $t_0 = t$ or otherwise $t_0 < t$ and for every $t', t'' \in [t_0, t]$ with $t' < t''$ we have: $\bigwedge_{j=1}^{n_q}(\sum_{i=1}^{n} c_{ij}^q(x_i(t'') - x_i(t')) \leq c_j^q(t'' - t'))$.

[5] We here give a direct proof; an alternative proof which uses the locality (cf. [28]) of the axioms of the form $\mathsf{Flow}_q(t_0, t_1)$ can also be given.

Then $\lim_{\substack{t'\to t\\ t'>t}} \bigwedge_{j=1}^{n_q} \sum_{i=1}^{n} c_{ij}^q \frac{x_i(t')-x_i(t)}{t'-t} \leq c_j^q$, hence $\sum_{i=1}^{n} c_{ij}^q \lim_{\substack{t'\to t\\ t'>t}} \frac{x_i(t')-x_i(t)}{t'-t} = \sum_{i=1}^{n} c_{ij}^q \dot{x}_i(t) \leq c_j^q$.
Then $x_1, \ldots, x_n$ describe a flow within mode q from a state satisfying Ψ to a state which does not satisfy Ψ. This contradicts the fact that Ψ is an invariant.

To prove that (2) $\Rightarrow$ (3) assume now that the formula $F_{\mathsf{flow}}(q)$ is satisfiable. A model of $F_{\mathsf{flow}}(q)$ consists of time moments $t_0 \leq t$ and linear functions $x_1, \ldots, x_n$ such that the values of these functions at the extremities of the interval $[t_0, t]$ satisfy formula $F_{\mathsf{flow}}(q)$. By the convexity of the state invariants, such a model describes a flow in mode q from a state in which Ψ holds to a state in which Ψ does not hold, i.e. also a model of $F_{\mathsf{Flow}}(q)$.

We now prove that (3) $\Rightarrow$ (1): Assume that Ψ is not preserved under flows in some state q for some interval $[t_0, t]$. If $t_0 = t$ everything is clear. Assume now that $t_0 < t$. We show that the interval $[t_0, t]$ and the functions describing the evolution of $x_1, \ldots, x_n$ define a model of formula $F_{\mathsf{flow}}(q)$. For this, we have to check that $\underline{\mathsf{flow}_q}(t_0, t)$ holds. As the functions $f_j(t) = \sum_{i=1}^{n} c_{ij}^q x_i(t)$ are continuous and differentiable, by the mean value theorem, for every $t_0 < t_1$ there exists $c \in [t_0, t_1]$ such that
$\frac{f_j(t_1)-f_j(t_0)}{(t_1-t_0)} = \dot{f}_j(c) = \sum_{i=1}^{n} c_{ij}^q \dot{x}_i(c) \leq_j c_j^q$
($x_1, \ldots, x_n$ satisfy the flow condition, $\sum_{i=1}^{n} c_{ij}^q \dot{x}_i(c) \leq c_j^q$). This shows that $x_1, \ldots, x_n$ define a model of $\underline{\mathsf{flow}_q}(t_0, t)$. □

An important consequence of Theorem 31 is the fact that checking invariant properties for linear hybrid automata for safety properties expressible as convex linear predicates (i.e. conjunctions of strict and non-strict inequalities) over X can be done in polynomial time.

Corollary 32 *Let S be an LHA and Ψ be a convex linear predicate over X. The satisfiability of any of the formulae $F_{\mathsf{flow}}(q)$ and $F_{\mathsf{jump}}(e)$ in Theorem 31 can be checked in polynomial time.*

Proof: We analyze the form of the constraints in the formulae above. All formulae of type $\mathsf{Inv}_q(\overline{x}(t))$ are conjunctions of linear inequalities in $x_1(t), \ldots, x_n(t)$. All formulae of the form $\mathsf{Jump}_e(\overline{x}(t), \overline{x}'(0))$ are conjunctions of linear inequalities in $x_1(t), \ldots, x_n(t), x_1'(0), \ldots, x_n'(0)$. The formula $\underline{\mathsf{flow}_q}(t_0, t_1)$ is a conjunction of strict and non-strict linear inequalities in $x_1(t_0), x_1(t_1), \ldots, x_n(t_0), x_n(t_1), t_0, t_1$; $\Psi(\overline{x}(t))$ is a conjunction of strict and non-strict linear inequalities in $x_1(t), \ldots, x_n(t)$, and $\neg\Psi(\overline{x}(t))$ is a disjunction of strict and non-strict linear inequalities in $x_1(t), \ldots, x_n(t)$. By distributivity, we need to check the satisfiability of $\mathsf{length}(\neg\Psi(\overline{x}(t)))$ conjunctions of linear inequalities. The PTIME complexity is based on the fact that any conjunction of linear inequalities and strict inequalities can be checked in polynomial time [17] (cf. Theorem 211). □

3.2 Invariant checking and parametric linear hybrid automata

For *parametric linear hybrid automata* we can consider two types of problems:

(1) parametric invariant checking (assuming that constraints on the parameters are given), and
(2) synthesis of constraints on parameters which guarantee that a formula is invariant under jumps and flows.

We first analyze the complexities of these problems in the case when no functional parameters occur, and then discuss the situation in which the dynamics is described using functional parameters.

Case 1: No functional parameters allowed We first consider the situation when no functional parameters occur. We allow constant parameters as bounds in the linear inequalities over the values of the continuous variables in Init_q, Inv_q, guard_e, jump_e and Ψ. The following results are consequences of Theorems 31, 211 and 212:

Theorem 33 (Parametric invariant checking) *Let S be a parametric hybrid automata with $P = P_c$ (i.e. all parameters are constants). Let Γ be a conjunction Γ of strict and non-strict linear inequalities over $P = P_c$ expressing the relationships between the parameters. Assume S satisfies a weak notion of parametricity, in which we allow the bounds in the linear inequalities over the values of the continuous variables in Init_q, Inv_q, guard_e and jump_e (where $q \in Q$ and $e \in E$) to be parameters. Let Ψ be a property expressed as convex linear predicate over X (possibly containing parameters as bounds in the linear inequalities). Then checking whether Ψ is an invariant is decidable in PTIME.*

Proof: By Theorem 31, Ψ is an invariant under the assumption that Γ holds iff for every $q \in Q$ and every $e = (q, q') \in E$, $\Gamma \wedge F_{\mathsf{flow}}(q)$ and $\Gamma \wedge F_{\mathsf{jump}}(e)$ are unsatisfiable. The assumptions ensure that these are sets of linear inequalities, thus satisfiability is decidable in PTIME. □

Theorem 34 (Constraint synthesis) *Let S be a parametric hybrid automata with $P = P_c$ (i.e. all parameters are constants). Let Γ be a conjunction Γ of strict and non-strict linear inequalities over $P = P_c$ expressing the relationships between the parameters. Assume S satisfies a weak notion of parametricity, in which we allow the bounds in the linear inequalities over the values of the continuous variables in $\mathsf{Init}_q, \mathsf{Inv}_q, \mathsf{guard}_e$, and jump_e to be parameters. Let Ψ be a property expressed as convex linear predicate over X (possibly containing parameters as bounds in the linear inequalities).*

(1) The problem of deriving constraints on parameters which guarantee that Ψ, is an invariant has polynomial complexity, with the degree of the polynomial equal to the number of continuous variables of the system.
(2) If all constraints are in $UTVPI^{\neq}$ then quantifier elimination can be done in time $O(dn^4)$, where d is the number of disequations and n is the number of variables in the set of constraints in Theorem 31.

Proof: (1) is an immediate consequence of Theorem 212. (2) is a consequence of Theorem 213. □

Remark. If the description of the automaton also contains parameters as coefficients and/or bounds for the linear inequalities describing the flows then Theorem 31 provides a reduction to checking the satisfiability of a family of non-linear constraints and parametric verification and the complexity is in general exponential.

Example 4 *We illustrate the ideas on Example 2.2. The invariance property we study is:*

$$\Psi(x_1, x_2, x_3) = x_1 + x_2 + x_3 \leq L_{\mathsf{overflow}} \wedge -\epsilon \leq x_1 - x_2 \leq \epsilon.$$

We assume that $L_f < L_{\mathsf{overflow}}$ and $\epsilon_a < \epsilon$. By Theorem 31, Ψ is an invariant iff for every mode $q \in \{1, 2, 3, 4\}$ the following formula $F_{\mathsf{flow}}(q)$ is unsatisfiable:

$$\Psi(\overline{x}(0)) \wedge \neg\Psi(\overline{x}(t)) \wedge \mathsf{Inv}_q(\overline{x}(0)) \wedge \mathsf{Inv}_q(\overline{x}(t)) \wedge \underline{\mathsf{flow}}_q(\overline{x}, t)$$

and $F_{\mathsf{Jump}}(e)$ is unsatisfiable for all $e \in E$. We ignore the redundant formulae corresponding to jumps which do not change the values of the variables and only analyze the jumps with reset, namely $e_{i_j} = (i_j, 4), i = 1, 2, 3, j = 1, 2$.[6] As an illustration we present the formula $F_{\mathsf{flow}}(q)$ for $q = 2$ (invariance under the flow in reaction mode):

$$\begin{aligned}
&(x_1(0)+x_2(0)+x_3(0) \leq L_{\mathsf{overflow}} \wedge -\epsilon \leq x_1(0)-x_2(0) \leq \epsilon)\wedge\\
&\neg(x_1(t)+x_2(t)+x_3(t) \leq L_{\mathsf{overflow}} \wedge -\epsilon \leq x_1(t)-x_2(t) \leq \epsilon)\wedge\\
&L_f \leq x_1(0) + x_2(0) + x_3(0) \leq L_{\mathsf{overflow}} \wedge x_3(0) \leq \mathsf{max} \wedge\\
&L_f \leq x_1(t) + x_2(t) + x_3(t) \leq L_{\mathsf{overflow}} \wedge x_3(t) \leq \mathsf{max} \wedge\\
&x_1(t) - x_1(0) \leq -\mathsf{dmin} \cdot t \wedge x_2(t) - x_2(0) \leq -\mathsf{dmin} \cdot t \wedge x_3(t) - x_3(0) \geq \mathsf{dmin} \cdot t\wedge\\
&(x_1(t) - x_1(0)) - (x_2(t) - x_2(0)) = 0\wedge\\
&(x_1(t) - x_1(0)) + (x_2(t) - x_2(0)) + (x_3(t) - x_3(0)) = 0.
\end{aligned}$$

[6] For these, the verification tasks are trivial.

This is the disjunction of two conjunctions of strict and non-strict linear inequalities, hence its satisfiability can be checked in PTIME. We used H-PILoT [15] and also, separately, Z3 [23] for checking the satisfiability of $F_{\mathsf{flow}}(q)$ (assuming $0 < L_f < L_{\mathsf{overflow}} \wedge 0 < \epsilon_a < \epsilon \wedge 0 < \mathsf{dmin}$) or to generate constraints on the parameters which guarantee (un)satisfiability of $F_{\mathsf{flow}}(q)$. We also used H-PILoT directly on $F_{\mathsf{Flow}}(q)$ (the complete instantiation facilities of H-PILoT allowed to construct $F_{\mathsf{flow}}(q)$ from $F_{\mathsf{Flow}}(q)$ automatically).

Case 2: Functional parameters are allowed We first show that the results of Theorem 31 also hold for PLHA with dynamics described by convex *extended* linear predicates of the form

$$\sum_{i=1}^{n} a_i x_i \leq f \qquad \text{resp.} \qquad \sum_{i=1}^{n} b_i \dot{x}_i \leq b$$

where f is either an uninterpreted function satisfying the concavity condition or a concrete concave function (possibly constant) depending only on time (t), and b is either a parameter or a constant, and for properties Ψ which are expressed by convex *extended* linear predicates of the form

$$\sum_{i=1}^{n} d_i x_i \leq g$$

where g is either an uninterpreted function satisfying the concavity condition or a concrete concave function (possibly constant) depending only on time (t).

We use the following axiomatization of the concavity conditions (for an interval I of the form $(-\infty, a], [a, b]$ or $[b, \infty)$).

$$\mathsf{Conc}^I(f) \;\; \forall x, y, z \left(x, y \in I \wedge x < z \leq y \rightarrow \frac{f(z) - f(x)}{z - x} \geq \frac{f(y) - f(x)}{y - x} \right).$$

Let Conc^I be the conjunction of all the concavity conditions (relative to an interval I) for the bounding parametric (non-interpreted) functions which appear in the specification of the parametric hybrid automaton under consideration.

Theorem 35 *The following are equivalent for any LHA:*

(1) Ψ is an invariant of the automaton;

(2) For every $q \in Q$ and $e = (q, q') \in E$, the following formulae are unsatisfiable:

$$F_{\mathsf{Flow}}(q) \qquad \mathsf{Conc}^{[t_0,t]} \wedge \Psi(\overline{x}(t_0)) \wedge \mathsf{Flow}_q(t_0, t) \wedge \neg\Psi(\overline{x}(t)) \wedge t \geq t_0$$

$$F_{\mathsf{jump}}(e) \qquad \Psi(\overline{x}(t)) \wedge \mathsf{Jump}_e(\overline{x}(t), \overline{x}'(0)) \wedge \mathsf{Inv}_{q'}(\overline{x}'(0)) \wedge \neg\Psi(\overline{x}'(0))$$

(3) For every $q \in Q$ and $e = (q, q') \in E$, the following formulae are unsatisfiable:

$$F_{\mathsf{flow}}(q) \qquad \mathsf{Conc}^{[t_0,t]} \wedge \Psi(\overline{x}(t_0)) \wedge \mathsf{Inv}_q(\overline{x}(t_0)) \wedge \underline{\mathsf{flow}}_q(t_0, t) \wedge \mathsf{Inv}_q(\overline{x}(t)) \wedge \neg\Psi(\overline{x}(t)) \wedge t \geq t_0$$

$$F_{\mathsf{jump}}(e) \qquad \Psi(\overline{x}(t)) \wedge \mathsf{Jump}_e(\overline{x}(t), \overline{x}'(0)) \wedge \mathsf{Inv}_{q'}(\overline{x}'(0)) \wedge \neg\Psi(\overline{x}'(0))$$

(4) For every $q \in Q$ and $e = (q, q') \in E$, the formulae $F'_{\mathsf{flow}}(q)$ and $F_{\mathsf{jump}}(e)$ are unsatisfiable, where $F_{\mathsf{jump}}(e)$ is as above and:

$$F'_{\mathsf{flow}}(q) \; \mathsf{Inv}_q(\overline{x}(t_0)) \wedge \Psi(\overline{x}(t_0)) \wedge \underline{\mathsf{flow}}_q(t_0, t) \wedge \mathsf{Inv}_q(\overline{x}(t)) \wedge \neg\Psi(\overline{x}(t)) \wedge t \geq t_0$$

Proof: [7] The proof closely follows the structure of the proof of Theorem 31. (1) $\Rightarrow$ (2) and (3) $\Rightarrow$ (1) can be proved as in Theorem 31. To prove that (2) $\Rightarrow$ (3) assume now that the formula $F_{\mathsf{flow}}(q)$ is satisfiable. A model of $F_{\mathsf{flow}}(q)$ consists of time moments $t_0 \leq t$ and linear functions $x_1, \ldots, x_n$ (and concave functions modeling the upper bounds for their linear combinations) such that the values of these functions at the extremities of the interval $[t_0, t]$ satisfy

[7] We here give a direct proof; an alternative proof can also be given, which uses the locality (cf. [28]) of the axioms of the form $\mathsf{Flow}_q(t_0, t_1)$ and of the convexity axioms.

formula $F_{\mathsf{flow}}(q)$. The convexity of the state invariants is now guaranteed by the concavity of the functions modeling the upper bounds. This is trivial if $t_0 = t$. Assume now that $t_0 < t$ and the dynamics in mode q is described by $\sum a_i x_i \leq f$ and $\sum b_i \dot{x}_i \leq b$. The function x_i describing the line passing through points $x_i(t_0)$ and $x_i(t)$ has the property that for every $t' \in (t_0, t)$, $x_i(t') = \frac{(t-t')x_i(t_0)+(t'-t_0)x_i(t)}{t-t_0}$. Therefore, $\sum a_i^q x_i(t') = \frac{(t-t')\sum a_i x_i(t_0)+(t'-t_0)\sum a_i x_i(t)}{t-t_0} \leq \frac{(t-t')\sum a_i x_i(t_0)+(t'-t_0)\sum a_i x_i(t)}{t-t_0} \leq \frac{(t-t')f(t_0)+(t'-t_0)f(t)}{t-t_0} \leq f(t')$; the last inequality is a consequence of the concavity of f. The fact that the constraint on slopes is fulfilled can be shown as in the proof of Theorem 31.

Then such a model describes a flow in mode q from a state in which Ψ holds to a state in which Ψ does not hold, i.e. also a model of $F_{\mathsf{Flow}}(q)$.

It is obvious that (4) $\Rightarrow$ (3). We now show that (3) $\Rightarrow$ (4). Assume that the formula $F'_{\mathsf{flow}}(q)$ is satisfiable. A model of $F_{\mathsf{flow}}(q)$ consists of time moments $t_0 \leq t$ and linear functions $x_1, \ldots, x_n$ (and concave functions modeling the bounds) such that the values of these functions at the extremities of the interval $[t_0, t]$ satisfy formula $F'_{\mathsf{flow}}(q)$. As before, the case when $t_0 = t$ is simple. We assume that $t_0 < t$. It is easy to show that also the linear functions passing through the values of these functions at the extremities of the interval determine a model of $F'_{\mathsf{flow}}(q)$. Moreover, since every linear function is concave, these new function symbols satisfy, in particular, the conditions in $\mathsf{Conc}^{[t_0,t_1]}$. □

Theorem 35 adresses the question whether there exist values of the parameters (be they functional or non-functional) for which Ψ is not an invariant. If (4) holds, then Ψ is an invariant under jumps and flows for all possible interpretations of the parameters.

We would like to consider additional constraints on the parameters, and check invariance of formulae under these additional assumptions. Of special interest are constraints on the functional parameters with the following locality property (which we studied in [28]):

(Loc) For every set $G(\overline{y})$ of quantifier-free clauses with variables $\overline{y}$, possibly containing occurrences of the functional parameters say in a set Σ, the following are equivalent:
(1) $\Gamma_f \wedge G(\overline{y})$ is unsatisfiable in the extension $\mathbb{R}^\Sigma$ of $\mathbb{R}$ with uninterpreted functions in Σ
(2) $\Gamma_f[G] \wedge G(\overline{y})$ is unsatisfiable in $\mathbb{R}^\Sigma$

where $\Gamma_f[G]$ consists of all instances of the clauses in Γ_f in which the universally quantified variables are replaced with terms containing only variables among $\overline{y}$, with the property that every term starting with a function symbol in Σ is, after this instantiation, a term occurring in $G(\overline{y})$.

Example 5 *In previous work we identified a wide class of properties which have such a locality condition. Besides convexity and concavity, other properties are monotonicity properties or boundedness properties. For details we refer to [29]. The equivalence of (2) and (3) in Theorem 31 is a locality result for the set of axioms for flows we consider: It states that the universally quantified clauses in the formula* $\mathsf{Flow}_q(t_0, t_1)$*, namely:*

$$\forall t(t_0 \leq x \leq t_1 \rightarrow \mathsf{Inv}_q(\overline{x}(t))) \wedge \forall t, t'(t_0 \leq t < t' \leq t_1 \rightarrow \underline{\mathsf{flow}}_q(t, t'))$$

can be replaced with their instances in which the variables t, t' *are instantiated with* t_0 *and* t_1*.*

Theorem 36 (Parametric invariant checking) *Consider a weak notion of parametricity, in which we allow the bounds in the linear inequalities over the values of the continuous variables in* Init_q, Inv_q, guard_e, *and* jump_e *to be parameters (either constant or functions). Let* Ψ *be a property expressed as convex linear predicate over* X*, possibly containing parameters (either constant or functions) as bounds in the linear inequalities. Assume that:*

(i) The properties of the parameters are expressed as a conjunction $\Gamma_0 \wedge \Gamma_f$*, where* Γ_0 *is a conjunction of strict and non-strict linear inequalities representing the relationships between non-functional parameters (in a set* Σ*) and* Γ_f *is a set of (universally quantified) clauses expressing the properties of the functional parameters – containing the concavity conditions for the bounding functional parameters; and*

(ii) The set of axioms Γ_f is a set of clauses which satisfies condition Loc.

Then checking whether Ψ is an invariant is decidable. If after instantiation, the set of instances in $\Gamma_f[F_{\mathsf{flow}}(q)]$ and $\Gamma_f[F_{\mathsf{jump}}(e)]$ consists of conjunctions of linear inequalities for every q, e then, under the assumptions in Thm. 31 invariant checking is decidable in PTIME also in this case.

Proof: By Theorem 31, Ψ is an invariant under the assumptions $\Gamma_0 \wedge \Gamma_f$ iff for every $q \in Q$ and every $e = (q, q') \in E$, $\Gamma_0 \wedge \Gamma_f \wedge F_{\mathsf{flow}}(q)$ and $\Gamma_0 \wedge \Gamma_f \wedge F_{\mathsf{jump}}(e)$ are unsatisfiable.

- The formulae Γ_0 and $F_{\mathsf{flow}}(q)$ are ground (the variables $\overline{x}$ are implicitly existentially quantified when checking satisfiability); they are in fact a set of ground unit clauses consisting of linear inequalities. If $G(\overline{x}) = \Gamma_0 \wedge F_{\mathsf{flow}}(q)$, then by the locality assumption for Γ_f, $\mathbb{R} \wedge \Gamma_f \wedge G$ is unsatisfiable iff $\Gamma_f[G] \wedge G(\overline{x})$ is unsatisfiable in $\mathbb{R}^{\Sigma}$, where $\Gamma_f[G]$ consists of all instances of the clauses in Γ_f in which the universally quantified variables are replaced with terms containing only variables among $\overline{x}$, with the property that every term starting with a function symbol in Σ is, after this instantiation, a term occurring in $G(\overline{x})$. If after instantiation, the set of instances in $\Gamma_f[G]$ consist of conjunctions of linear inequalities for every q, e then, under the assumptions in Thm. 31 invariant checking is decidable in PTIME.
- Since the formulae $F_{\mathsf{jump}}(e)$ do not contain any occurrence of functional parameters, $\Gamma_0 \wedge \Gamma_f \wedge F_{\mathsf{jump}}(e)$ is unsatisfiable iff $\Gamma_0 \wedge F_{\mathsf{jump}}(q)$ is unsatisfiable. This is a set of linear inequalities, so satisfiability can be checked in PTIME. □

Of course, if the bounds are given by concrete, non-linear functions, the problem remains decidable, but the complexity will be higher (typically in EXPTIME).

For the weak notion of parametricity, in which we allow the bounds in the linear inequalities over the values of the continuous variables to be parameters (either constant or functions), constraints on the parameters (including universally quantified constraints on the functional parameters) can be obtained using quantifier elimination and the method described in [30].

3.3 Complexity of bounded model checking

We consider the problem of checking whether a formula safe is preserved under runs of length bounded by k.

Theorem 37 *Assume that* Safe *is a convex linear predicate and that the LHA S is input-deterministic. The following are equivalent:*

(1) There exists no run σ of S with length at most k starting with a state satisfying Init *such that condition* Safe *does not hold at the end of σ.*

(2) The formulae F_i are unsatisfiable for all $1 \leq i \leq k$, where F_i states that there exists a run with end point satisfying ¬Safe *passing through exactly i states, i.e. is the following formula (where* InState *is a predicate which indicates the state the system is in):*

$$
\begin{aligned}
&\bigvee_{q\in Q} \quad [\mathsf{Init}_q(\overline{x}^1(0)) \wedge \mathsf{InState}^1{=}q \wedge \mathsf{Inv}_q(\overline{x}^1(0)) \wedge \underline{\mathsf{flow}}_q^{\overline{x}^1}(0,t_1) \wedge \mathsf{Inv}_q(\overline{x}^1(t_1)) \wedge t_1{\geq}0 \wedge \\
&\qquad \bigvee_{(q,q')\in E} \mathsf{guard}_{(q,q')}(\overline{x}^1(t_1))] \wedge \\
&\bigwedge_{\substack{e_1=\\(q,q')\in E}} [\mathsf{InState}^1{=}q \wedge \mathsf{guard}_{e_1}(\overline{x}^1(t_1)) \rightarrow (\mathsf{jump}_{e_1}(\overline{x}^1(t_1),\overline{x}^2(0)) \wedge \\
&\qquad \wedge \mathsf{InState}^2{=}q' \wedge \mathsf{Inv}_{q'}(\overline{x}^2(0)) \wedge \underline{\mathsf{flow}}_{q'}^{\overline{x}^2}(0,t_2) \wedge \mathsf{Inv}_{q'}(\overline{x}^2(t_2)) \wedge t_2{\geq}0 \wedge \\
&\qquad \bigvee_{(q',q")\in E} \mathsf{guard}_{(q',q")}(\overline{x}^2(t_2)))] \wedge \\
&\cdots \\
&\bigwedge_{\substack{e_{i-1}=\\(q,q')\in E}} [\mathsf{InState}^{i-1}{=}q \wedge \mathsf{guard}_{e_{i-1}}(\overline{x}^{i-1}(t_{i-1})) \rightarrow \mathsf{jump}_{e_{i-1}}(\overline{x}^{i-1}(t_{i-1}),\overline{x}^i(0)) \wedge \\
&\qquad \wedge \mathsf{InState}^i{=}q' \wedge \mathsf{Inv}_{q'}(\overline{x}^i(0)) \wedge \underline{\mathsf{flow}}_{q'}^{\overline{x}^i}(0,t_i) \wedge \mathsf{Inv}_{q'}(\overline{x}^i(t_i)) \wedge t_i \geq 0] \wedge \\
&\qquad \neg\mathsf{Safe}(\overline{x}^i(t_i))
\end{aligned}
$$

Hence the bounded model checking problem is in NP.

Proof: Assume that the formula in (2) is satisfiable. Then we can use a model of this formula to construct a run of S with length at most k which starts with a state satisfying Init and ends with an unsafe state, so (1) does not hold. In the model we construct, $x_1, \ldots, x_n$ are linear functions (lines passing through $x_i(t_i)$ and $x_i(t_{i+1})$). The fact that such linear functions satisfy the flow conditions can be shown as in the proof of Theorem 31, (2) $\Rightarrow$ (3).

Conversely, assume that there exists a run of S with length at most k which starts with a state satisfying Init and ends with an unsafe state. The functions $x_1, \ldots, x_n$ can be used to construct a model for the formula in (2); the fact that the model also satisfies the formulae $\underline{\mathsf{flow}}_q(t_i, t_i')$ can be shown by using the same argument as in Theorem 31, (3) $\Rightarrow$ (1).

We now analyze the form of the formulae in F_i. Using distributivity, the conjunct for mode q in the first disjunction can be written as a disjunction of $\mathsf{bd}_E(q)$ conjunctions of linear inequalities, where $\mathsf{bd}_E(q)$ is the number of all mode switches starting in mode q. Thus the disjunction in the first part of the formula can be written as a disjunction of $\sum_{q\in Q} \mathsf{bd}_E(q) = |E|$ conjunctions of linear inequalities, each of size linear in the description of S.

Every implication in the part of the formula describing jump j in the run is of the form

$\mathsf{InState}^j = q \wedge \psi(x) \rightarrow \bigwedge_{i=1}^n p_i(x,x') \wedge (\bigvee_{e=(q,q')} \mathsf{guard}_e(x'))$.

This is the conjunction between n implications of size linear in the initial size of the problem and one conjunction of $\prod_{e=(q,q')\in E} \mathsf{length}(\mathsf{guard}_e)$ implications, resulting from the CNF translation of $\mathsf{InState}^j = q \wedge \psi(x) \rightarrow (\bigvee_{e=(q,q')} \mathsf{guard}_e(x'))$.

Hence, the total number of clauses in the part of F_i describing the jumps is at most $i * K_S$ where:

$K_S = \sum_{q\in Q}(p(|S|) + \prod_{e=(q,q')\in E} \mathsf{length}(\mathsf{guard}_e))$,

where $p(|S|)$ is a linear factor in the size of the description of the LHA S. K_S can be considered to be a constant depending only on the description of the automaton S.

Since the first and last formula of F_i are disjunctions, F_i can be written as the disjunction of at most $|E| * \mathsf{length}(\mathsf{Inv})$ conjunctions of at most $i * K_S$ clauses consisting of disjunctions of strict and non-strict linear inequalities. (Note that for the reduction to the problem of checking the satisfiability of a family of disjunctions of strict and non-strict linear inequalities we can replace all values $x_k^i(0), x_k^i(t_i)$ with variables, but we also need to add the $k * i$ congruence axioms corresponding to $0 = t_i \rightarrow x_k^i(0) = x_k^i(t_i)$; this does not change the general form of the problem.)

Since all these formulae have size polynomial in terms of i and of the size of the safety property to be checked (up to a constant depending on the description of the automaton S), the complexity of checking satisfiability of such formulae is in NP [31] (cf. Theorem 211). $\square$

We identify situations in which checking unsatisfiability of the formulae in (2) above can be performed in polynomial time. For this, we use results on disjunctive constraints in linear arithmetic for which satisfiability is decidable in PTIME mentioned in Theorem 212.

Corollary 38 *Assume that for every $e \in E$, guard_e contains only equalities, and from each mode there is at most one switch to another mode. Then checking the satisfiability of all formulae F_i can be done in PTIME.*

Proof: The premise ensures that each F_i is equivalent to a disjunction of $|Q| * \mathsf{length}(\mathsf{Safe})$ conjunctions of Horn disjunctive constraints. The result follows from Theorem 212. □

We study some alternative situations in which PTIME satisfiability can be guaranteed.

Theorem 39 *Assume that Init_q, Inv_q, Safe, Jump, guard_e can be expressed as conjunctions of inequalities between variables, flow can be expressed as a conjunction of non-strict inequalities between variables, and that from each mode there is at most one switch to another mode. We can adapt the encoding of the formulae F_i by accumulating the overall time (such that the last constraint is $t_i \leq t$). Hence, the satisfiability of all formulae F_i can be checked in PTIME.*

Proof: The premise ensures that F_i is a conjunction of Ord-Horn constraints for all i, so we can use Theorem 214. □

Remark: The restriction that the constraint in flow can be expressed as a strict or non-strict inequality between variables holds iff the only constraints allowed in flow are monotonicity or antitonicity conditions of the form:

$$\text{if } t_1 \leq t_2 \text{ then } x(t_1) \leq x(t_2) \qquad (\text{or if } t_1 \leq t_2 \text{ then } x(t_1) \geq x(t_2)).$$

Parametric hybrid automata. We first consider the situation in which all parameters are not functions. The results in Theorem 33 and Theorem 34(1) lift in a natural way to yield complexity results for bounded model checking of parametric hybrid automata (in this case EXPTIME, also for the case of convex parametric bounds).

Theorem 310 (Parametric BMC) *Consider a weak notion of parametricity, in which we allow the bounds in the linear inequalities over the values of the continuous variables in $\mathsf{Init}_q, \mathsf{Inv}_q, \mathsf{guard}_e$, and jump_e to be parameters. Let Safe be a property expressed as convex linear predicate over X (possibly containing parameters as bounds in the linear inequalities). Assume that the relationships between the parameters – expressed as a conjunction Γ of strict and non-strict linear inequalities – are given. Then the problem of checking whether Safe is preserved on all paths on length $\leq k$ starting with an initial state is in NP.*

Proof: Analogous to the proof of Theorem 33, taking into account that in this case we have a reduction to checking the satisfiability of a set of clauses in linear arithmetic. □

Theorem 311 (Constraint synthesis for BMC) *Consider a weak notion of parametricity, in which we allow the bounds in the linear inequalities over the values of the continuous variables in $\mathsf{Init}_q, \mathsf{Inv}_q, \mathsf{guard}_e$, and jump_e to be parameters. Let Safe be a property expressed as convex linear predicate over X (possibly containing parameters as bounds in the linear inequalities).*

(1) The problem of deriving constraints on parameters which guarantee that Safe is preserved on all paths on length $\leq k$ starting with an initial state is decidable, and has in general exponential complexity.
(2) If for every $e \in E$, guard_e contains only equalities, and from each mode there is at most one switch to another mode, then the problem has polynomial complexity, with the degree of the polynomial equal to the number of continuous variables of the system.

Proof: Analogous to the proof of Theorem 34(1). The assumption in (2) ensures that each F_i is equivalent to a disjunction of conjunctions of Horn disjunctive constraints, and for such constraints

quantifier elimination has polymonial complexity, with the degree of the polynomial equal to the number of continuous variables of the system. □

Remark. Theorem 34(2) will not hold in this case because the complexity results for quantifier elimination for $UTVPI^{\neq}$ refer only to unit clauses.

Functional parameters. These results also extend in a natural way to the situation in which functional parameters are allowed in the bounds of the linear inequalities describing the constraints on the values of the continuous variables.

Theorem 312 (Parametric BMC) *Consider a notion of parametricity, in which we allow the bounds in the linear inequalities over the values of the continuous variables in* Init_q, Inv_q, guard_e, *and* jump_e *to be parameters (either constant or functions). Let* Safe *be a property expressed as convex linear predicate over* X*, possibly containing parameters (either constant or functions) as bounds in the linear inequalities. Assume that:*

(i) the properties of the parameters are expressed as a conjunction $\Gamma_0 \wedge \Gamma_f$*, where* Γ_0 *is a conjunction of strict and non-strict linear inequalities representing the relationships between non-functional parameters (in a set* Σ*) and* Γ_f *is a set of (universally quantified) clauses expressing the properties of the functional parameters – containing the concavity conditions for the bounding functional parameters; and*

(ii) The axioms in Γ_f *form a set of (universally quantified) clauses which satisfy condition* Loc.

Then bounded model checking for property Safe *is decidable. If after instantiation, for every* i *the set of instances in* $\Gamma_f[F_i]$ *consists of conjunctions of implications of linear inequalities, then bounded model checking is in NP also in this case.*

Proof: Analogous to the proof of Theorem 36. □

If we consider a notion of parametricity, in which we allow the bounds in the linear inequalities over the values of the continuous variables to be parameters (either constant or functions), then additional constraints on the parameters (including universally quantified constraints on the functional parameters) can be obtained using quantifier elimination and the results in [30].

Example 6 *Consider again the LHA in Example 1, and the same safety property as in Example 4. We want to check whether an unsafe state can be reached from an initial state in at most two steps (we assume that* $\mathsf{Init}_q = \mathsf{Inv}_q$ *for every* $q \in Q$*). This can be reduced to checking whether the following formula is unsatisfiable (where the disjunction in the first part of the formula is taken over all modes, and the conjunction in the middle part of the formula is taken over all mode switches):*

$$
\begin{aligned}
&[(\mathsf{InState}^1{=}1 \wedge \mathsf{Inv}_1(x_1^1(0),x_2^1(0),x_3^1(0)) \wedge \underline{\mathsf{flow}}_1^{\overline{x}^1}(0,t_1) \wedge \mathsf{Inv}_1(x_1^1(t_1),x_2^1(t_1),x_3^1(t_1)) \wedge t_1{\geq}0 \wedge \\
&\quad (\mathsf{guard}_{(1,2)}(x_1^1(t_1),x_2^1(t_1),x_3^1(t_1)) \vee \mathsf{guard}_{(1,4)}(x_1^1(t_1),x_2^1(t_1),x_3^1(t_1)) \vee \mathsf{guard}_{(1,4)'}(x_1^1(t_1),x_2^1(t_1),x_3^1(t_1))) \\
&\vee \ldots \\
&\quad (\mathsf{InState}^1{=}4 \wedge \mathsf{Inv}_4(x_1^1(0),x_2^1(0),x_3^1(0)) \wedge \underline{\mathsf{flow}}_4^{\overline{x}^1}(0,t_1) \wedge \mathsf{Inv}_4(x_1^1(t_1),x_2^1(t_1),x_3^1(t_1)) \wedge t_1{\geq}0))] \\
\wedge\ &[(\mathsf{InState}^1{=}1 \wedge \sum_{i=1}^{3} x_i^1(t_1){\geq}L_f \rightarrow \bigwedge_{i=1}^{3} x_i^2(0){=}x_i^1(t_1) \wedge \mathsf{InState}^2{=}2 \wedge \mathsf{Inv}_2(\overline{x}^2(0)) \wedge \underline{\mathsf{flow}}_2^{\overline{x}^2}(0,t_2) \wedge \mathsf{Inv}_2(\overline{x}^2(t_2)) \wedge t_2{\geq}0) \\
&\wedge(\mathsf{InState}^1{=}1 \wedge x_1^1(t_1) - x_2^1(t_1) \geq \epsilon_a \rightarrow \bigwedge_{i=1}^{3} x_i^2(0) = 0 \wedge \mathsf{InState}^2{=}4 \wedge \bigwedge_{i=1}^{3} x_i^2(0) = 0 \wedge \bigwedge_{i=1}^{3} x_i^2(t_2) = 0 \wedge t_2{\geq}0) \\
&\wedge(\mathsf{InState}^1{=}1 \wedge x_1^1(t_1) - x_2^1(t_1) \leq -\epsilon_a \rightarrow \bigwedge_{i=1}^{3} x_i^2(0) = 0 \wedge \mathsf{InState}^2{=}4 \wedge \bigwedge_{i=1}^{3} x_i^2(0) = 0 \wedge \bigwedge_{i=1}^{3} x_i^2(t_2) = 0 \wedge t_2{\geq}0)] \\
\wedge\ &\ldots \\
&[(\mathsf{InState}^1{=}3 \wedge -\epsilon_a \leq x_1^1(t_1) - x_2^1(t_1) \leq \epsilon_a \wedge 0 \leq x_3^1(t_1) \leq \mathsf{min} \rightarrow \\
&\quad \bigwedge_{i=1}^{3} x_i^2(0) = x_i^1(t_1) \wedge \mathsf{InState}^2{=}1 \wedge \mathsf{Inv}_1(x_1^2(0),x_2^2(0),x_3^2(0)) \wedge \underline{\mathsf{flow}}_1(\overline{x}^2,t_2) \wedge \mathsf{Inv}_1(x_1^2(t_2),x_2^2(t_2),x_3^2(t_2)) \wedge t_2{\geq}0) \\
&\wedge(\mathsf{InState}^1{=}3 \wedge x_1^1(t_1) - x_2^1(t_1) \geq \epsilon_a \rightarrow \bigwedge_{i=1}^{3} x_i^2(0) = 0 \wedge \mathsf{InState}^2{=}4 \wedge \bigwedge_{i=1}^{3} x_i^2(0) = 0 \wedge \bigwedge_{i=1}^{3} x_i^2(t_2) = 0 \wedge t_2{\geq}0) \\
&\wedge(\mathsf{InState}^1{=}3 \wedge x_1^1(t_1) - x_2^1(t_1) \leq -\epsilon_a \rightarrow \bigwedge_{i=1}^{3} x_i^2(0) = 0 \wedge \mathsf{InState}^2{=}4 \wedge \bigwedge_{i=1}^{3} x_i^2(0) = 0 \wedge \bigwedge_{i=1}^{3} x_i^2(t_2) = 0 \wedge t_2{\geq}0)] \\
\wedge\ &\quad (x_1^2(t_2) + x_2^2(t_2) + x_3^2(t_2) > L_{\mathsf{overflow}} \vee x_1^2(t_2) - x_2^2(t_2) > \epsilon \vee x_1^2(t_2) - x_2^2(t_2) < -\epsilon).
\end{aligned}
$$

To obtain the translation to a satisfiability problem in $LI(\mathbb{R})$ we also need to take into account the instances of the congruence axioms: $0 = t_1 \rightarrow x_k^1(0) = x_k^1(t_1)$ and $0 = t_2 \rightarrow x_k^2(0) = x_k^2(t_2)$, for $k = 1, 2, 3$.

It can be seen that every component in the disjunction in the first part of the formula is a conjunction of linear inequalities. The disjunction in the last part of the formula is a disjunction of strict linear inequalities. Writing $a < b$ as $a \leq b \wedge a \neq b$ it is easy to see that this last formula can be seen as a disjunction of Horn disjunctive linear constraints.

The second part of the formula can be written as a conjunction of implications of the form $\bigwedge_{i=1}^{n} p_i \rightarrow p$, where $p_1, \ldots, p_n$ and p are linear inequalities. In conclusion, using distributivity, the formula above can be written as the disjunction of $4 * 3$ conjunctions of implications of the form $\bigwedge_{i=1}^{n} p_i \rightarrow p$, where $p_1, \ldots, p_n$ and p are linear inequalities.

If all the guards in the automaton are expressed using equalities (for our example, only using the limit conditions instead of the full inequalities, e.g. $x_1 + x_2 + x_3 = L_f$ instead of $x_1 + x_2 + x_3 \geq L_f$ for the transition from mode 1 to mode 2) then the formula is the disjunction of 12 conjunctions of Horn disjunctive linear constraints, hence its satisfiability is decidable in PTIME.

4 Reasonable hybrid automata

We are interested in a class of controllers for which mode entry conditions are chosen with sufficient safety margin such that it is guaranteed that the system remains in a node at least a given time, and for which mode invariants are compatible with the safety condition(s).

Invariant compatibility. We consider hybrid automata in which the mode invariants are compatible with the safety condition Safe and with the guards of the transitions. In order to define compatibility with the guards of the transitions we introduce the following notation. If C is a conjunction of linear inequalities let $C^b = \{\sum_{j=1}^{n} a_{ij}x_j{=}a_i \mid \sum_{j=1}^{n_i} a_{ij}x_j \lhd a_i \in C, \text{ where } \lhd \in \{\leq, <, >, \geq\}\}$ be the boundary of C^b. We say that a set of values $(v_1, \ldots, v_n)$ satisfies C^b if it satisfies at least one equality in C^b. (Thus, C^b is regarded as the disjunction of the constraints it contains.) For every convex predicate Ψ we denote by Ψ^c the closure of Ψ, obtained by replacing all strict inequalities with non-strict inequalities.

Definition 41 *The mode invariants of a hybrid automaton S are compatible with the safety condition* Safe *and with the guards of the transitions if:*

(1a) Compatibility with mode invariants: *All closures of mode invariants are safe*[8] *, i.e. for every mode $q \in Q$, $\models \forall \overline{x}(\mathsf{Inv}_q^c(\overline{x}) \rightarrow \mathsf{Safe}(\overline{x}))$;*

(1b) Compatibility with the guards: *If the values of the control variables are on the boundary of the mode invariant of mode q and those values are reachable by a flow starting in the inner envelope of the mode then the values of the control variables must satisfy the guard of some mode switch.*

$$\forall \overline{y} \forall \overline{z} [(\mathsf{Inv}_q^b(\overline{z}) \wedge \mathsf{Inv}_q^c(\overline{z}) \wedge \exists t(0 < t \wedge \bigwedge_{i=1}^{n} x_i(0) = y_i \wedge \mathsf{InEnv}_q(\overline{x}(0)) \wedge \underline{\mathsf{flow}}(0, t) \wedge \bigwedge_{i=1}^{n} x_i(t) = z_i)) \rightarrow \bigvee_{(q,q') \in E} \mathsf{guard}_{(q,q')}(\overline{z})].$$

Note: Since, as mentioned above, Inv_q^b is regarded as the disjunction of the equalities it containts, the formula $\mathsf{Inv}_q^b \wedge \mathsf{Inv}_q^c(\overline{z})$ describes the boundary of the polyhedron defined by $\mathsf{Inv}_q^c(\overline{z})$, i.e. the boundary of the invariant of mode q.

Chatter-freedom. A hybrid automaton S is *chatter-free* if there exists $\epsilon_t > 0$ such that mode entry conditions are chosen with sufficient safety margin such that it is guaranteed that the system remains in a node at least a given time ϵ_t. Formally:

[8] This condition can be replaced with the condition that all invariants are safe if we ensure that the safety properties we consider are defined only using non-strict inequalities, i.e. are closed sets.

Definition 42 *A hybrid automaton S is* chatter-free with minimal dwelling time ϵ_t *(where $\epsilon_t > 0$) if:*

(2a) all transitions lead to an inner envelope, i.e. for all $q \in Q, (q, q') \in E$ the following formula is valid:
$\forall \overline{x}(\mathsf{Inv}_q(\overline{x}) \wedge \mathsf{guard}_{(q,q')}(\overline{x}) \wedge \mathsf{jump}_{(q,q')}(\overline{x}, \overline{x}') \rightarrow \mathsf{InEnv}_{q'}(\overline{x}'))$;

(2b) for any flow starting in the inner envelope of a mode q, no guard of a mode switch (q, q') will become true in a time interval smaller than ϵ_t.

A hybrid automaton S is chatter-free *iff S is chatter-free with minimal dwelling time ϵ_t for some $\epsilon_t > 0$.*

Reasonable hybrid automata. We say that an extended hybrid automaton S is *reasonable* w.r.t. a safety property Safe, if (1) S is input deterministic; (2) the mode invariants are compatible with Safe and with the guards of the transitions; (3) S is chatter-free.

4.1 Checking the property of being reasonable

We show that checking whether an extended LHA is reasonable can be done in PTIME.

Checking input determinism Checking input determinism for LHA reduces to checking satisfiability of linear constraints, so can be done in PTIME.

Checking invariant compatibility We will give a PTIME test for checking invariant compatibility. We first show that compatibility of the mode invariants with safety conditions expressed as convex linear predicates can be checked in PTIME:

Lemma 43 *Let S be an LHA and let Safe be a convex linear predicate. The problem of checking whether for all $q \in Q$, $\models \forall \overline{x}(\mathsf{Inv}^c_q(\overline{x}) \rightarrow \mathsf{Safe}(\overline{x}))$ is decidable in PTIME.*

Proof: Let $q \in Q$ be a mode. Then $\forall x(\mathsf{Inv}^c_q(x) \rightarrow \mathsf{Safe}(x))$ is valid iff $\mathsf{Inv}^c_q(x) \wedge \neg\mathsf{Safe}(x)$ is unsatisfiable. The last problem is decidable in PTIME since $\mathsf{Inv}^c_q(x)$ and $\mathsf{Safe}(x)$ are sets of linear constraints. □

We now show that compatibility with the guards can be checked in PTIME.

Lemma 44 *Checking the condition (1b) in Definition 41 can be reduced to the problem of checking the satisfiability of $|Q| \prod_{e \in E} \mathsf{length}(\mathsf{guard}_e)$ systems of linear constraints (each decidable in PTIME).*

Example 7 *Consider a refinement of our example in which e.g. the flow in mode 1 is described by $\dot{x}_1 = \dot{x}_2 = k > 0$, and $\dot{x}_3 = 0$. We check whether condition (1b) in Definition 41 holds as follows:*

$$\mathsf{Inv}^b_1 = \{x_1 = 0, x_2 = 0, x_3 = 0, x_1 + x_2 + x_3 = 0, \\ x_1 + x_2 + x_3 = L_f, x_1 - x_2 = \epsilon_a, x_1 - x_2 = -\epsilon_a\}.$$

Let (x_1, x_2, x_3) be a state satisfying some equality in Inv^b_1 and such that there exists a flow (of positive length) leading from a state (x'_1, x'_2, x'_3) satisfying Inv_1 to (x_1, x_2, x_3). Because of the flow rules in mode 1, it can be proved that this state cannot satisfy any of $\{x_1 = 0, x_2 = 0, x_1 + x_2 + x_3 = 0\}$.

If $x_1 + x_2 + x_3 = L_f$ then the guard of mode switch (1,2) is true. If $x_1 - x_2 = \epsilon_a$ or $x_1 - x_2 = -\epsilon_a$ then the guard of one of the mode switches from 1 to 4 is true.

Checking chatter-freedom Chatter-freedom can also be checked in PTIME.

Theorem 45 *Let S be an LHA. Checking whether S is chatter-free can be done in PTIME.*

Proof: Condition (2a) can be expressed as a set of satisfiability problems for conjunctions of linear (strict) inequalities, so can be checked in PTIME.

The fact that condition (2b) can be checked in PTIME can be proved as follows. Note that (2b) holds iff $\bigwedge_{(q,q')\in E} F_{q,q'}$ is unsatisfiable, where:

$F_{q,q'}: \quad \mathsf{InEnv}(x_1(0),\ldots,x_n(0)) \wedge \mathsf{Inv}_q(\overline{x}(0)) \wedge \underline{\mathsf{flow}}_q(0,t) \wedge \mathsf{guard}_{(q,q')}(x_1(t),\ldots,x_n(t)) \wedge t \leq \epsilon_t.$

The formulae $\mathsf{Inv}_q(\overline{x}(t))$, $\mathsf{guard}(q,q',\overline{x},t)$ are conjunctions of linear inequalities in $x_1(t),\ldots,x_n(t)$. By assumption, $\mathsf{InEnv}(\overline{x}(0))$ is a disjunction of convex linear inequalities (here we can use distributivity which does not influence complexity). Note that $\underline{\mathsf{flow}}_q(0,t) = \bigwedge_{j=1}^{n_q^f}(\sum_{i=1}^{n} c_{ij}^q(x_i(t)-x_i(0)) \leq c_j^q * t)$ i.e. a conjunction of linear inequalities in $x_1(0)$, $x_1(t)$, ..., $x_n(0)$, $x_n(t)$ and t. Such sets of constraints can be checked in PTIME [17]. □

Example 8 *Consider a further refinement of Example 7 in which besides the minimal bound* dmin *also a maximal bound* dmax *is imposed on the rate of change of the substances x_i, $i = 1,2,3$ (reflected in the flow conditions). Assume that the inner envelope of mode 1 is:*

$\mathsf{InEnv}_1(x_1,x_2,x_3) = 0 \leq x_1+x_2+x_3 \leq L_{\mathsf{safe}} \wedge \bigwedge_{i=1}^{3} x_i \geq 0 \wedge -\epsilon_{\mathsf{safe}} \leq x_1 - x_2 \leq \epsilon_{\mathsf{safe}} \wedge 0 \leq x_3 \leq \mathsf{min}$

where $0 < L_{\mathsf{safe}} < L_f$ and $0 < \epsilon_{\mathsf{safe}} < \epsilon_a$.

The minimal dwelling time in mode 1 is ϵ_t iff the following formulae are unsatisfiable:

$\mathsf{InEnv}_1(x_1(0),x_2(0),x_2(0)) \wedge \underline{\mathsf{flow}}_1(x_1,x_2,x_2,t) \wedge \mathsf{guard}_e(x_1(t),x_2(t),x_2(t)) \wedge t \leq \epsilon_t$

where e is one of the transitions from mode 1 into mode 2 or 4.

We first analyze the switch change $(1,2)$. The corresponding formula is F_e:

$0 \leq x_1(0)+x_2(0)+x_3(0) \leq L_{\mathsf{safe}} \wedge \bigwedge_{i=1}^{3} x_i(0) \geq 0 \wedge -\epsilon_{\mathsf{safe}} \leq x_1(0)-x_2(0) \leq \epsilon_{\mathsf{safe}} \wedge x_3(0) \leq \mathsf{min} \wedge$
$x_3(t) = x_3(0) \wedge \mathsf{dmin} \cdot t \leq x_1(t) - x_1(0) \leq \mathsf{dmax} \cdot t \wedge \mathsf{dmin} \cdot t \leq x_2(t) - x_2(0) \leq \mathsf{dmax} \cdot t \wedge$
$L_f \leq x_1(t) + x_2(t) + x_2(t) \wedge t \leq \epsilon_t.$

It can be seen (e.g. by direct proof or using QE) that F_e is unsatisfiable iff $L_f > L_{\mathsf{safe}} + 2 \cdot \mathsf{dmax} \cdot \epsilon_t$.

We now consider the two mode switches e_1^1, e_2^1 from 1 to 4. The corresponding formulae are $F_{e_1^1}$:

$0 \leq x_1(0)+x_2(0)+x_3(0) \leq L_{\mathsf{safe}} \wedge \bigwedge_{i=1}^{3} x_i(0) \geq 0 \wedge -\epsilon_{\mathsf{safe}} \leq x_1(0)-x_2(0) \leq \epsilon_{\mathsf{safe}} \wedge x_3(0) \leq \mathsf{min} \wedge$
$x_3(t) = x_3(0) \wedge \mathsf{dmin} \cdot t \leq x_1(t) - x_1(0) \leq \mathsf{dmax} \cdot t \wedge \mathsf{dmin} \cdot t \leq x_2(t) - x_2(0) \leq \mathsf{dmax} \cdot t \wedge$
$x_1(t) - x_2(t) \geq \epsilon_a \wedge t \leq \epsilon_t$

and $F_{e_2^1}$:

$0 \leq x_1(0)+x_2(0)+x_3(0) \leq L_{\mathsf{safe}} \wedge \bigwedge_{i=1}^{3} x_i(0) \geq 0 \wedge -\epsilon_{\mathsf{safe}} \leq x_1(0)-x_2(0) \leq \epsilon_{\mathsf{safe}} \wedge x_3(0) \leq \mathsf{min} \wedge$
$x_3(t) = x_3(0) \wedge \mathsf{dmin} \cdot t \leq x_1(t) - x_1(0) \leq \mathsf{dmax} \cdot t \wedge \mathsf{dmin} \cdot t \leq x_2(t) - x_2(0) \leq \mathsf{dmax} \cdot t \wedge$
$x_1(t) - x_2(t) \leq -\epsilon_a \wedge t \leq \epsilon_t.$

It can be seen that $F_{e_1^1}$ is unsatisfiable iff $\epsilon_{\mathsf{safe}} < \epsilon_a + (\mathsf{dmax} - \mathsf{dmin}) \cdot \epsilon_t$ and similar for $F_{e_2^1}$.
In our tests we considered:

(i) a non-parametric version of the running example in which $L_f, L_{\mathsf{safe}}, \mathsf{dmin}, \epsilon_a, \epsilon_{\mathsf{safe}}, \epsilon_t, \ldots$ are instantiated such that the constraints above (and similar constraints) are satisfied;
(ii) a parametric version of the example in which such constraints are additionally specified;
(iii) a parametric version of the example in which we use the equivalence above (and analogous for other modes/switches) to derive constraints on the parameters.

5 Complexity of verification problems for reasonable LHA

We now consider safety properties and time-bounded reachability and show that for reasonable LHA checking time-bounded reachability properties is decidable in nondeterministically polynomial time, and that for reasonable LHA where all transitions are universally mode reachable, checking safety properties is decidable in PTIME.

5.1 Safety properties with exhaustive entry conditions

We show that for reasonable LHA convex safety properties with exhaustive entry conditions can be checked in PTIME. Before this, we show that the invariant compatibility condition ensures that in all modes, if the variables change according to the flow rules for the mode it cannot happen that the mode invariant becomes false without a guard of a mode switch becoming true before this.

Lemma 51 *S be an ELHA satisfying the invariant compatibility condition (1b). Let q be a mode of S. Consider an evolution of the continuous variables of S according to the flow rules in q, which starts in the inner envelope of mode q. If no guard of a mode switch becomes true during this flow, then everywhere during the flow Inv_q holds. If the guard of a mode switch becomes true at the end of the flow (but no mode switch becomes true before the end) then before the end of the flow Inv_q holds, and at the end of the flow the values of the variables $x_1, \ldots, x_n$ satisfy Inv_q^c (the closure of Inv_q obtained by replacing all strict inequalities with non-strict inequalities).*

Proof: Assume that there exists an evolution of the continuous variables of S according to the flow rules in q (from time 0 to time t) from state s_1 to state s_2 such that Inv_q holds at s_1 and Inv_q^c does not hold at s_2. Since $x_1, \ldots, x_n$ are all continuous, any linear combinations of these functions has the intermediate value property, so (since Inv_q is convex) there exists t' with $0 \leq t' \leq t$ such that $\overline{x}(t')$ satisfies at least one of the equalities in $\mathsf{Inv}_q^b \wedge \mathsf{Inv}_q^c$. We consider the smallest t' in $[0, t]$ with this property. Since in this case there is a flow in mode q from $x(0)$ to $x(t')$ starting in the inner envelope of mode q, by the invariant compatibility condition (1b) it follows that $\mathsf{guard}_{(q,q')}(\overline{x}(t'))$ for some $(q, q') \in E$.

We identify two situations:

- $\mathsf{Inv}_q(\overline{x}(t'))$ holds. Then Inv_q holds at every point during this flow from 0 to t'.
- Inv_q does not contain its border (because it is defined using strict inequalities) and $\mathsf{Inv}_q(\overline{x}(t'))$ does not hold. However, $\mathsf{Inv}_q^c(\overline{x}(t'))$ holds.

Thus, if during the flow no guard of a mode switch (q, q') becomes true, Inv_q holds at every point during this flow; if a mode switch becomes true only in the end, Inv_q holds at every point before the end of the flow, and the last value of the variables satisfies Inv_q^c. □

Theorem 52 *Let $\phi = \Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Box\phi_{\mathsf{safe}})$, and let S be an LHA satisfying the invariant compatibility condition w.r.t. ϕ_{safe} and the chatter-freedom condition (2a). The following are equivalent:*

(1) $S \models \Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Box\phi_{\mathsf{safe}})$.
(2) The following invariance conditions hold:
  (a) $\phi_{\mathsf{ExhEntry}} \models \phi_{\mathsf{safe}}$[9].
  (b) ϕ_{safe} is invariant under all mode-reachable jumps of the hybrid automaton.

Proof: Assume first that (2) holds. Let σ be a run of the hybrid automaton S which starts in a state satisfying ϕ_{ExhEntry} (hence, by (2)(a), also ϕ_{safe}). We prove that ϕ_{safe} holds at the end of the run by induction on the length of the run. If the run has length 1, it consists of only one state (so no state change takes place). Hence, by (2)(a) ϕ_{safe} holds. Assume now that the property holds for all runs of length i. We prove it for runs of length $i + 1$.

- Assume $\sigma = (\sigma_1 s_1)s_2$, where the state change (s_1, s_2) corresponds to a flow within a mode q. Then by the induction hypothesis, Inv_q and ϕ_{safe} are satisfied at state s_1. Assume that ϕ_{safe} does not hold in state s_2. Then Inv_q does not hold at s_2. [10] By the chatter-freeness condition

[9] This is always guaranteed because $\phi_{\mathsf{ExhEntry}} \models \mathsf{Inv}_q$ for some q and invariant compatibility (1a).

[10] If we restrict all runs to sequences of admissible states, then both s_1 and s_2 are admissible, so satisfy Inv_q, hence, by invariant compatibility (1a) ϕ_{safe} holds at s_2. In that case we can safely replace invariant compatibility condition (1a) with $\models \forall\overline{x}(\mathsf{Inv}_q(\overline{x}) \rightarrow \phi_{\mathsf{safe}}(\overline{x}))$. In the text of the proof a stronger invariance property was proved.

(2a) we know that s_2 is reached during a flow starting in the inner envelope of mode q. By Lemma 51, it follows that if no guard of a mode switch holds during this flow, then at every moment during the flow in q (including state s_2), Inv_q^c holds, so by invariant compatibility (1a) ϕ_{safe} holds at s_2.
- Assume now that the last state change of $\sigma = (\sigma_1 s_1)s_2$, say $s_1 s_2$, happens during a jump. The last state change in $(\sigma_1 s_1)$ was caused by a flow (of length ≥ 0) in a state q, and the last state s_1 satisfies the guard of some mode switch $e = (q, q')$. By the chatter-freedom condition (2a), the system entered this state in the inner envelope of mode q, so the concrete jump $s_1 s_2$ is mode reachable. By (b), ϕ_{safe} is preserved under mode reachable jumps. Thus, ϕ_{safe} is true also in state s_2.

We now prove that (1) implies (2).

- Consider first a 1-state run, consisting of a state s which satisfies ϕ_{ExhEntry}. Then ϕ_{safe} must be true at s.
- Now consider a state change $s_1 s_2$ caused by a mode reachable jump e from a mode q to a mode q', starting in a state s_1 which satisfies ϕ_{safe}. As state s_1 satisfies the guard of e and e is mode reachable there exists a state s' satisfying InEnv_q (hence also ϕ_{ExhEntry} and ϕ_{safe}) and there exists a flow within q from s' to s_1. Thus, $s' s_1 s_2$ is a run from a state satisfying InEnv_q to s_2. Since $S \models \Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Box\phi_{\mathsf{safe}})$, ϕ_{safe} is true in state s_2. □

Lemma 53 *A jump (s_1, s_2) where s_1, s_2 are states with $s_1 = (q, x_1, \ldots, x_n)$ and $s_2 = (q', x_1', \ldots, x_n')$ is mode reachable iff the following formula holds:*

$$\mathsf{Inv}_q(\overline{x}) \wedge \mathsf{guard}_e(\overline{x}) \wedge \mathsf{jump}_e(\overline{x}, \overline{x}') \wedge \exists x_1^0, \ldots, x_n^0, t(\mathsf{InEnv}_q(\overline{x}^0) \wedge \bigwedge_{j=1}^{n_q} \sum_{i=1}^{n} c_{ij}^q (x_i - x_i^0) \leq c_j^q \cdot t)$$

where the coefficients c_{ij}^q and c_j^q are those in the formulae defining the flow rule in more q, $\underline{\mathsf{flow}}_q(0,t) = \bigwedge_{j=1}^{n_q} \sum_{i=1}^{n} c_{ij}^q (x_i(t) - x_i(0)) \leq c_j^q \cdot t$.

Proof: The formula in the statement of the theorem is a reformulation of the reachability condition. □

Theorem 54 *Let $\phi = \Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Box\phi_{\mathsf{safe}})$ where ϕ_{safe} is a convex linear predicate and S be an LHA satisfying the invariant compatibility condition w.r.t. ϕ_{safe}. Then (1) and (2) in Theorem 52 are equivalent to:*

(3) All formulae F_e for every $e{=}(q, q'){\in}E$, are unsatisfiable, where:

$$\begin{aligned} F_e = \phi_{\mathsf{safe}}(\overline{x}) \wedge (&\mathsf{Inv}_q(\overline{x}) \wedge \mathsf{guard}_e(\overline{x}) \wedge \mathsf{jump}(\overline{x}, \overline{x}') \wedge \\ &(\mathsf{InEnv}_q(\overline{x}^0) \wedge \textstyle\bigwedge_{j=1}^{n_q} \sum_{i=1}^{n} c_{ij}^q (x_i - x_i^0) \leq c_j^q \cdot t)) \wedge \\ &\wedge \neg\phi_{\mathsf{safe}}(\overline{x}') \end{aligned}$$

where $\underline{\mathsf{flow}}_q = \bigwedge_{j=1}^{n_q} \sum_{i=1}^{n} c_{ij}^q (x_i(t) - x_i(0)) \leq c_j^q \cdot t$.

Proof: The equivalence of (2) and (3) follows as in the proof of Theorem 31, taking into account Lemma 53 and the fact that under condition (1a), (2)(a) is always true. □

Corollary 55 *Let $\phi = \Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Box\phi_{\mathsf{safe}})$ where ϕ_{safe} is a convex linear predicate and S be a reasonable ELHA w.r.t. ϕ_{safe}. The problem of checking whether $S \models \phi$ is decidable in PTIME.*

Note. In Corollary 55 only the invariant compatibility condition and chatter-freeness condition (2a) in the definition of reasonable ELHA are needed.

5.2 Time-bounded reachability properties

We consider time-bounded reachability properties of the form:

$$\phi = \Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Diamond_{\leq t}\phi_{\mathsf{safe}}).$$

Theorem 56 *Let S be an ELHA satisfying the condition that there is a lower bound ϵ_t for the minimal dwell time for each mode. The following are equivalent:*

(1) $S \models \Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Diamond_{\leq t}\phi_{\mathsf{safe}})$.
(2) There exists a run σ of S with total time length at most t starting with a state satisfying ϕ_{ExhEntry}, s.t. the safety condition ϕ_{safe} holds at the end of σ.
(3) There exists a constant $k(t, \epsilon_t)$ and a run σ of S with at most $k(t, \epsilon_t)$ states (and total time length at most t), starting from a state satisfying ϕ_{ExhEntry} such that ϕ_{safe} holds at the end of σ.

For all input-deterministic ELHA, if ϕ_{safe} is a convex linear predicate then (3) is equivalent to (4):

(4) There exists a constant $k(t, \epsilon_t)$ such that at least one of the formulae $F'_i \wedge t_i \geq \epsilon_t \wedge \sum_{j=1}^{i} t_j \leq t$, $1 \leq i \leq k(t, \epsilon_t)$ is satisfiable (where the F'_i are like the formulae F_i in item (2) of Theorem 37, except for the fact that Init *is replaced with ϕ_{ExhEntry} and $\neg$*Safe *with ϕ_{safe}).*

Proof: (1) and (2) are equivalent by definition. (3) obviously implies (2). Assume that (2) holds. Let $k_1(t, \epsilon_t) = \frac{t}{\epsilon_t}$. Since the automaton must remain at least time ϵ_t in every continuous state, any run of cumulated time at most t consisting of an alternation of flows (each with duration at least ϵ_t) and jumps can have at most $k_1(t, \epsilon_t)$ flows, hence it can have length at most $k(t, \epsilon_t) = 2 * k_1(t, \epsilon_t)$. Hence (3) holds. If the assumptions in Theorem 37 hold, the equivalence of (3) and (4) can be proved as for Theorem 37. □

Corollary 57 *Let $\Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Diamond_{\leq t}\phi_{\mathsf{safe}})$ be a bounded-time reachability condition with exhaustive entry conditions, where ϕ_{safe} is a convex linear predicate. Let S be a reasonable ELHA w.r.t. ϕ_{safe}. Then the problem of checking whether $S \models \Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Diamond_{\leq t}\phi_{\mathsf{safe}})$ is in NP.*

The problem is decidable in PTIME if (i) all guards of mode switches can be expressed as equalities and from each mode there is at most one control switch to another mode, or (ii) all constraints in F'_i are in the OrdHorn class.

Note. In Corollary 57 only the input determinism and chatter-freeness conditions of S are needed. The results also hold for time-bounded reachability properties for which $\phi_{\mathsf{entry}} \models \phi_{\mathsf{ExhEntry}}$.

Example 9 *Consider the variant of our running example*[11] *in Example 4, in which bounds on the rate of change of the substances x_1, x_2, x_3 in modes 2 (*React*) and 4 (*Dump*) are additionally specified. Consider the following properties:*

(1) $\phi_1 = \Box(\phi_{\mathsf{entry}} \rightarrow \Diamond_{\leq t} x_1 = x_2 = x_3 = 0)$, *where* $\phi_{\mathsf{entry}} = 0 \leq x_1 + x_2 + x_3 \leq L_{\mathsf{overflow}} \wedge \bigwedge x_i \geq 0 \wedge \neg(-\epsilon_a \leq x_1 - x_2 \leq \epsilon_a)$.
(2) $\phi_2 = \Box(\mathsf{InEnv}_1 \rightarrow \Diamond_{\leq t}(x_3 \geq \mathsf{max} \vee x_1 = x_2 = x_3 = 0))$

(1) ϕ_1 states that if x_1, x_2 are mixed in the wrong proportion then there exists an evolution such that the tank is emptied in at most t time units. Since there is no mode switch from state 4 into another state, we only need to check runs of length at most 2, i.e. $\sigma_1 = 14$, $\sigma_2 = 24$ and $\sigma_3 = 34$ or prefixes thereof.

*(2) ϕ_2 states that if we start in an entry state in the inner envelope of mode 1 (***Fill***), then there exists a run of the system such that, in at most t units of time, either a sufficient amount of reaction product is produced or the substances are dumped. Assume that $t = 200s$, and the minimal dwelling time in each mode is $100s$. Then we only need to check runs with two states and one jump, namely $\sigma_1 = 14$ and $\sigma_2 = 12$.*

[11] We also considered a refinement of the example in Sect. 2.2 in which mode 4 (**Dump**) is replaced with three modes: **Add** x_1 (in which the inflow of substance x_2 is stopped and only x_1 is added), **Add** x_2 (in which the inflow of x_1 is stopped and only x_2 is added) and **Dump'** described as Dump. Due to space limitations we do not present this refinement here.

6 Summary of results

We can summarize our results as follows. Assume that S is an LHA and the safe states are conjunctions of linear inequalities.

Invariant checking and BMC (non-parametric, or with non-functional parameters). The following complexity results hold for non-parametric verification, and also for parametric verification – if the constraints Γ_0 on the parameters[12] are linear inequalities (invariant checking) resp. sets of clauses in linear arithmetic in the classes mentioned in the header (BMC), and parameters are allowed only as bounds for linear inequalities over the values of the continuous variables X [13]:

	Inv. checking	BMC		
		general	equality guards + determinism	ord Horn + determinism
LHA	PTIME	NP	PTIME	PTIME

For non-linear constraints, parametric coefficients or parametric bounds in flows the complexity is exponential.

The complexity of the problem of generating constraints on parameters (Inv(1): assumption that only bounds on linear combinations of control variables are parametric; Inv(2): assume that also coefficients and/or bounds on rates of growth are parametric) is presented below:

	Inv(1)	Inv(2)	BMC
LHA	$O(\|C\|^n)$	EXPTIME	EXPTIME
LHA ($UTVPI^{\neq}$)	PTIME	EXPTIME	EXPTIME

Invariant checking and BMC: PLHA with functional parameters. Assume that we only allow the bounds in the linear inequalities over the values of the continuous variables to be parametric (either constant or functions) and that the following hold:

(i) The properties of the parameters are expressed as a conjunction[14] $\Gamma = \Gamma_0 \wedge \Gamma_f$, where Γ_0 is a conjunction of strict and non-strict linear inequalities representing the relationships between non-functional parameters, and Γ_f is a set of clauses expressing the properties of the functional parameters (which is considered to be part of the description of the system);
(ii) The set of axioms Γ_f is a set of clauses which has a certain locality property; and
(iii) After instantiation[15], the set of instances in $\Gamma_f[F_{\mathsf{flow}}(q)]$ and $\Gamma_f[F_{\mathsf{jump}}(e)]$ consists of:
- conjunctions of linear inequalities for every q, e (for invariant checking);
- conjunctions of implications of linear inequalities (for general BMC);
- clauses in the classes mentioned in the header (for the special PTIME decidable classes of BMC in the last two columns).

[12] In this case Γ_0 is regarded as part of the description of the system, so the size of Γ_0 is considered to be a constant.
[13] "determinism" is condition: "$|\{(q,q')|(q,q')\in E\}|\leq 1$ for all $q\in Q$".
[14] Also in this case Γ is regarded as part of the description of the system, so the sizes of Γ_0 and Γ_f are supposed to be constant.
[15] For every set G of ground clauses, the number of clauses in the set of instances $\Gamma_f[G]$ is polynomial in the number of ground subterms occurring in G, but the degree of the polynomial depends on the form of Γ_f (which is here considered to be part of the description of the automaton, hence constant).

Then the following complexity results hold:

	Inv. checking	BMC		
		general	equality guards + determinism	ord Horn + determinism
PLHA	PTIME	NP	PTIME	PTIME

Testing the property of being reasonable. The complexity of the verification of reasonability conditions is described below:

Input determinism	PTIME
Invariant compatibility:	PTIME
– Compatibility with safety conditions (1a)	$\lvert Q\rvert$ PTIME tests
– Compatibility with the guards (in mode q)	$\prod_{(q,q')\in E} \mathsf{length}(\mathsf{guard}_{(q,q')})$ PTIME problems
Chatter-free (2a), (2b)	PTIME

Safety properties and time-bounded reachability. The complexity of these verification problems for reasonable ELHA is given below:

$\Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Box\phi_{\mathsf{safe}})$	$\Box(\phi_{\mathsf{ExhEntry}} \rightarrow \Diamond_{<t}\, \phi_{\mathsf{safe}})$	
Invariant Compatibility + Chatter-free (2a)	Input determ. + Chatter-free	Input determ. + Chatter-free + determinism + (equality guards or OrdHorn)
PTIME	NP	PTIME

7 Experimental results

We checked Example 2.2 with the SMT-solver Z3 [23]; for generating constraints between parameters we used Redlog [9]. For the parametric examples presented in this paper, some ability to handle non-linear arithmetical constraints was essential; Z3 is one of the few SMT checkers which supports such constraints. In our tests, Z3 proved to be very efficient: For bounded model checking it verified a run of 100 steps in slightly more than 1 second (cf. Fig. 1).

During the tests we noticed an interesting fact: If we consider BMC for usual LHA, the possibility exists that immediately after entering into a continuous mode, the guards of a jump become true and a jump takes place after a zero time delay. When encoding such problems as satisfiability problems, we therefore have to take into account the congruence axioms for the functions which model the evolution of the continuous variables over time. On the contrary, if chatter-freeness is guaranteed then we know that there is a minimal dwelling time inside any mode, so after each mode change, the limits of the interval during which the system is in a certain mode are different. Therefore, in this case we do not need to consider the congruence axioms for the functions which model the evolution of the continuous variables over time in each mode. This is reflected in the time difference in the table below, where the tests in the first column (BMC-LHA) refer to BMC for usual LHA, in which the time t spent inside of a mode satisfies the condition $t \geq 0$, whereas the tests in the second column (BMC-CF) refer to BMC for chatter-free LHA, in which the time t spent inside of a mode satisfies the condition $t \geq \delta$, where δ is a parameter for which we only specified that $\delta > 0$.

8 Conclusion

In this paper we proved that verification of safety properties for reasonable ELHA can be reduced to invariant checking resp. to problems very similar to bounded model checking and, ultimately, to checking the validity of certain formulae (obtained using a polynomial reduction). We showed that

Steps	**Z3** (BMC-LHA)	**Z3** (BMC-CF)
2	0.02s	0.01s
3	0.03s	0.02s
4	0.03s	0.02s
5	0.04s	0.03s
6	0.05s	0.04s

Steps	**Z3** (BMC-LHA)	**Z3** (BMC-CF)
7	0.06s	0.04s
10	0.07s	0.05s
20	0.15s	0.10s
50	0.43s	0.40s
100	1.07s	1.00s

The running times are given in User + sys times (in s). The experiments were carried out on an Intel Xeon 3 GHz, 512 kB cache. Z3 version 2.13 was used.

Fig. 1. Experimental results

the problem of checking the validity of such formulae is typically in NP, and identified verification tasks which can be performed in PTIME. For the parametric verification of such properties we identified conditions when the complexity is in PTIME, NP, or EXPTIME. We also analyzed the complexity of checking the property of being reasonable for ELHA.

Acknowledgments. This work was partly supported by the German Research Council (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS). See `www.avacs.org` for more information.

References

1. M. Agrawal, P. S. Thiagarajan. The Discrete Time Behavior of Lazy Linear Hybrid Automata. *Proc. HSCC 2005, LNCS 3414*, 55-69, Springer 2005.
2. A. Agrawal, G. Simon, G. Karsai. Semantic Translation of Simulink/Stateflow Models to Hybrid Automata Using Graph Transformations. *Electronic Notes in Theoretical Computer Science* 109: 43-56, 2004.
3. R. Alur, T.A. Henzinger, P.H. Ho. Automatic symbolic verification of embedded systems. *IEEE Trans. Software Eng.* 22(3): 181-201, 1996.
4. T. Brihaye, Ch. Michaux, C. Rivière, Ch. Troestler. On O-Minimal Hybrid Systems. *Proc. HSCC 2004, LNCS 2993*, 219-233, Springer 2004.
5. T. Brihaye, Ch. Michaux. On the expressiveness and decidability of o-minimal hybrid systems. *Journal of Complexity* 21(4): 447-478, 2005.
6. W. Damm, C. Ihlemann, V. Sofronie-Stokkermans. Decidability and complexity for the verification of reasonable linear hybrid automata. *Proc. HSCC 2011.* To appear, 2011.
7. W. Damm, G. Pinto, S. Ratschan. Guaranteed termination in the verification of LTL properties of non-linear robust discrete time hybrid systems. *Int. J. Found. Comput. Sci.* 18(1): 63-86, 2007.
8. W. Damm, H. Dierks, S. Disch, W. Hagemann, F. Pigorsch, C. Scholl, U. Waldmann, B. Wirtz. Exact and Fully Symbolic Verification of Linear Hybrid Automata with Large Discrete State Spaces. *Science of Computer Programming. Special Issue on Automated Verification of Critical Systems*, Ed. M. Roggenbach, Accepted for publication, 2011.
9. A. Dolzmann and T. Sturm. Redlog: Computer algebra meets computer logic. *ACM SIGSAM Bulletin* 31(2):2–9, 1997.
10. G. Frehse, S.K. Jha, B.H. Krogh. A counterexample guided approach to parameter synthesis for linear hybrid automata. *Proc. HSCC 2008*, LNCS 4981, pp. 187-200, Springer, 2008.
11. G. Frehse. Tools for the verification of linear hybrid automata models. *Handbook of Hybrid Systems Control, Theory – Tools – Applications*. Cambridge University Press, Cambridge, 2009.
12. S. Gulwani and A. Tiwari. Constraint-based approach for analysis of hybrid systems. In *Proc. CAV 2008*, LNCS 5123, pp. 190-203, Springer, 2008.
13. T.A. Henzinger, P.W. Kopke, A. Puri, P. Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences* 57(1): 94-124, 1998.
14. T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. Algorithmic Analysis of Nonlinear Hybrid Systems. *IEEE Transactions on Automatic Control* 43:540-554, 1998.
15. C. Ihlemann and V. Sofronie-Stokkermans. System description: H-PILoT. In *Proc. CADE 2009*, LNAI 5663, pp. 131-139, Springer 2009.

16. S. Jha, B.A. Brady, and S.A. Seshia Symbolic Reachability Analysis of Lazy Linear Hybrid Automata *Proceedings of FORMATS 2007*, 2007.
17. L. Khachian. A polynomial time algorithm for linear programming. *Soviet Math. Dokl.* 20:191-194, 1979.
18. M. Koubarakis. Tractable disjunctions of linear constraints: basic results and applications to temporal reasoning. *Theor. Comput. Sci.* 266: 311-339, 2001.
19. M. Koubarakis and S. Skiadopoulos. Querying temporal and spatial constraint networks in PTIME. *Artificial intelligence* 123: 223-263, 2000.
20. G. Lafferriere, G.J. Pappas, S. Sastry. O-Minimal hybrid systems. Mathematics of Control, Signals, and Systems, 13(1):1-21, 2000.
21. G. Lafferriere, G.J. Pappas, S. Yovine. A new class of decidable hybrid systems. *Proc. HSCC 1999*, LNCS 1569, pp.137-151, Springer, 1999.
22. J.S. Miller. Decidability and complexity results for timed automata and semi-linear hybrid automata. *Proc. HSCC 2000*, LNCS 1790, pp. 296-309, Springer, 2000.
23. L.M. de Moura and N. Bjørner. Z3: An Efficient SMT Solver. *Proc. TACAS 2008*, LNCS 4963, pp. 337-340, 2008.
24. B. Nebel and H.-J. Bürckert. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *Journal of the ACM* 42(1): 43–66, 1995.
25. G.E. Fainekos, G.J. Pappas. Robustness of temporal logic specifications. *Proc. FATES/RV 2006*, LNCS 4262, pp. 178-192, Springer, 2006.
26. A. Platzer and J.-D. Quesel. Logical verification and systematic parametric analysis in train control. *Proc. HSCC 2008*, LNCS 4981, pp. 646-649, Springer, 2008.
27. A. Platzer and J.-D. Quesel. European train control system: A case study in formal verification. *Proc. ICFEM 2009*, LNCS 5885, pp. 246-265, Springer, 2009.
28. V. Sofronie-Stokkermans. Hierarchic reasoning in local theory extensions. *Proc. CADE-20*, LNAI 3632, pp. 219-234, Springer, 2005.
29. V. Sofronie-Stokkermans. Efficient hierarchical reasoning about functions over numerical domains. In *Proc. KI 2008*, LNAI 5243, pp.135-143, Springer, 2008.
30. V. Sofronie-Stokkermans. Hierarchical reasoning for the verification of parametric systems. *Proc. IJCAR 2010*, LNAI 6173, pp. 171-187, Springer, 2010.
31. E.D. Sontag. Real addition and the polynomial hierarchy. *Inf. Proc. Letters* 20(3):115-120, 1985.
32. M. Swaminathan, M. Fränzle. A symbolic decision procedure for robust safety of timed systems. *Proc. TIME 2007*, p. 192, IEEE Computer Society, 2007.
33. G. J. Tee. Khachian's efficient algorithm for linear inequalities and linear programming. *ACM SIGNUM Newsletter archive* 15(1):13-15, 1980.
34. A. Tiwari. Formal Semantics and Analysis Methods for Simulink Stateflow Models. Unpublished report available from http://www.csl.sri.com/users/tiwari/, 2007.
35. F. Wang. Symbolic Parametric Safety Analysis of Linear Hybrid Systems with BDD-Like Data-Structures. *IEEE Trans. Software Eng.* 31(1): 38-51, 2005.