Verification of Bounded Discrete Horizon Hybrid Automata

Vladimeros Vladimerou, Pavithra Prabhakar, Mahesh Viswanathan, and Geir Dullerud

***Abstract*—We consider the class of o-minimally definable hybrid automata with a bounded discrete-transition horizon. We show that for every hybrid automata is this class, there exists a bisimulation of finite index, and that the bisimulation quotient can be effectively constructed when the underlying o-minimal theory is decidable. More importantly, we give natural specifications for hybrid automata which ensure the boundedness of discrete-transition horizon. In addition, we show that these specifications are reasonably tight with respect to the decidability of the models and that they can model modern day real-time and embedded systems. As a result, the analysis of several problems for these systems admit effective algorithms. We provide a representative example of a hybrid automaton in this class.**

Unlike previously examined subclasses of o-minimally defined hybrid automata with decidable verification properties (such as o-minimal [1] and extended o-minimal hybrid automata [2]), we do not impose re-initialization of the continuous variables in a memoryless fashion when a discrete transition is taken. Our class of hybrid systems has both rich continuous dynamics and strong discrete-continuous coupling, showing that it is not necessary to either simplify the continuous dynamics or restrict the discrete dynamics to achieve decidability.

***Index Terms*—Verification, O-minimality, Hybrid Automata, Cyber Physical Systems.**

I. Introduction

A. Motivation, related work and contributions

With the wide use of embedded computing and control algorithms for complex systems it is important to have models for the interaction of computer software and the dynamic environment it deals with. A widely known model for such systems is that of *hybrid automata* (HA) [3]. Hybrid automata have both discretely and continuously varying states whose dynamics may be tightly coupled. An *analysis* or *verification* problem asks if a given property is satisfied by a given model of a system.

Due to the complexity allowed by the general class of hybrid automata models, even simple properties such as the reachability problem [4] are known to be undecidable. Previous subclasses of hybrid automata that *do* admit algorithms for temporal property verification require either very simple dynamics for their continuous components (timed [5] and rectangular [4] hybrid automata) or strong resets which decouple the discrete dynamics from the continuous dynamics (o-minimal hybrid automata [1], or extended o-minimal hybrid automata [2]). These observations reinforced the folklore impression that in order to achieve decidability a model has to have either restricted coupling between continuous and discrete dynamics or simple continuous dynamics. Some exceptions have appeared, such as piecewise constant derivatives (PCD) systems [6] or polygonal systems [7]. However, the decidability in these cases is restricted to very low dimensions.

In this article we show that we can attain decidability even when simultaneously allowing complex continuous dynamics and strong coupling between the discrete-continuous interactions. We make use of a notion of boundedness in the discrete dynamics. We consider hybrid systems whose flows, invariants, guards and resets are definable in an o-minimal theory and have bounded number of discrete transitions in any execution. We show that such systems have a finite bisimulation which can be effectively constructed when the underlying theory is decidable. The observation that bounded horizon guarantees finite bisimulation was independently observed in [8] and [9]. In [9], the observation was made in the context of CTL model-checking by successive abstractions. Systems with bounded time have been studied in the context of timed automata [10], [11]; these consider the execution over a bounded time allowing arbitrary number of discrete transitions within the specified time, where as we allow arbitrary time elapse between transitions (in fact unbounded time after the last discrete transition) but constrain the number of discrete transitions in the executions.

We then introduce a natural class of specifications which satisfy discrete boundedness property. STORMED systems were first introduced in [8] as a class of systems with natural conditions on flows, guards, invariants and resets, which are satisfied by various physical systems, and the conditions are such that they ensure discrete boundedness property. We present STORMED hybrid systems and show that they satisfy the bounded discrete horizon property. We also show that these models are relatively tight by showing that the relaxation of certain constraints on STORMED systems renders the reachability problem undecidable. We illustrate that our specifications can be used to model and verify real systems through a concrete example. The present article builds upon the results in [8] and [9]. In comparison to [8], here we present extensions to the STORMED model which are decidable; and present an example of a STORMED system.

V. Vladimerou (vladimeros.vladimerou@tema.toyota.com, corresponding author) is with the Integrated Vehicle Systems Department at the Toyota Technical Center in Anna Arbor, MI. (Tel) 734-995-0062, (Fax) 734-778-8956.

P. Prabhakar (pavithra@caltech.edu) is with the IMDEA Software Institue and the Center for the Mathematics of Information at Caltech, MC 305-16, Pasadena, California 91125.

M. Viswanathan (vmahesh@illinois.edu) is with the Department of Computer Science at University of Illinois at Urbana-Champaign, 201 N Goodwin Avenue, Urbana, IL 61801. (Tel) 217-265-6298, (Fax) 217-265-6591.

G.E. Dullerud (dullerud@illinois.edu) is with the Department of Mechanical Science and Engineering at University of Illinois at Urbana-Champaign, 1206 West Green Street, Urbana, IL 61801. (Tel) 217-265-5078, (Fax) 217-244-6534. G.E. Dullerud is partially supported by AFOSR under grant FA9550-09-1-0221 and NSF under grant 0729500

B. Summary of article

As a first observation, we prove that systems with a bounded-discrete-horizon that are definable in an o-minimal structure admit a finite bisimulation. When the theory is decidable (e.g., semi-algebraic FOL) there are algorithms to construct such a bisimulation. We therefore give a simpler and less restrictive, in a way, specification for a subclass of hybrid automata which suffices to obtain decidability of expressive temporal logics such as CTL* or μ-calculus [12].

Subsequently we show how an extended class of models suitable for embedded and real-time systems satisfies these specifications and hence verification of a CTL* property in that class is equivalent to verifying a property on a finite automaton which is known to be decidable. This subclass comprises of so called STORMED hybrid systems[8] (with extensions) which satisfy the following constraints: They have the guards of two discrete transitions **s**eparable by some minimum positive distance. Next, they are definable in an *order-minimal* (**o**-minimal) theory. Furthermore, the flows (solutions of differential equations) of the continuous states have positive projections on some **m**onotonic direction ϕ on which their guards have **d**elimited-**e**nds.

We show the constraints of this subclass are reasonably tight, as relaxations of any of the them yield undecidable models. We also argue for the suitability of it for modeling embedded and real-time systems by giving an very general example of a system that can be modeled as such.

The article is sectioned as follows:

- We give a brief introduction to o-minimality and bisimulation.
- We define hybrid automata, adopted from [3] as well as o-minimally definable bounded-discrete-horizon hybrid automata.
- We present our main result, which is that the bounded discrete horizon hybrid automata, which are o-minimally definable admit a finite bisimulation which respects a given definable partition
- We revise STORMED hybrid systems [8] and show that they are bounded-discrete-horizon. We also show that the STORMED specifications are tight by proving that when any of them are removed, we have undecidability, by using reduction from two-counter machines.
- Before our concluding remarks we give an example of a real-time system, showing that it can be modeled as an extended STORMED hybrid system, therefore o-minimally definable bounded-discrete-horizon systems can describe such systems.

II. Preliminaries

In this section, we introduce definitions and notation we will use in the rest of the paper. In particular, we will define certain concepts related to logic, transition systems and relations.

A. Order-minimality (o-minimality)

We will assume that the reader is familiar with first-order logic. We use the the standard symbols $\wedge$ and $\vee$ to represent the logical connectives conjunction and disjunction; and $\models$ to represent satisfaction. A *vocabulary* or *language* consists of a finite set of relation symbols and a finite set of function symbols[1]. In this article we will consider first order vocabularies consisting of only relation[2] and constant symbols. A *structure* over a language consists of a non-empty set called *domain*, together with an assignment of a relation on the domain for each relation symbol in the language and an element of the domain for each constant symbol. Structures are used to define the semantics of First Order Logic (outside the scope of this introduction). We will call $\mathcal{A}$ to be a τ-structure if it is a structure over the signature τ. Let A be the domain of a τ-structure $\mathcal{A}$. We say that a k-ary relation $S \subseteq A^k$ is *definable* in a τ-structure $\mathcal{A}$ if there is a formula $\varphi(\chi_1, \chi_2, \ldots \chi_k)$ over τ with free variables $\chi_1, \ldots \chi_k$, such that $S = \{(a_1, \ldots, a_k) | \mathcal{A} \models \varphi[\chi_i \mapsto a_i]_{i=1}^k\}$. A k-ary function f is definable if its graph (i.e. the set $\{(\chi_1, \ldots, \chi_k, f(\chi_1, \ldots \chi_k))\}$) is definable. The set of all sentences that hold in a structure $\mathcal{A}$ is called the *theory* of $\mathcal{A}$, denoted by $T(\mathcal{A})$. We say that $T(\mathcal{A})$ or simply $\mathcal{A}$ is *decidable* if membership in the set $T(\mathcal{A})$ can be decided by some algorithm. For example, in a decidable structure one can check the emptiness of a definable relation, as well as whether two definable relations are equal.

A binary relation $\leq$ on a set A that is reflexive, transitive, antisymmetric ($\forall a, b\ ((a \leq b \ \wedge \ b \leq a) \ \Rightarrow a = b)$), and total ($\forall a, b\ (a \leq b \vee b \leq a)$) is said to be a *total ordering*. A *totally ordered set* is a set on which we have a total ordering. An *interval* is a subset of a totally order set, which can be defined using one or two bounds as follows: $\{x : a \leq x \leq b\}$, $\{x : x \leq a\}$, and $\{x : a \leq x\}$. The interval $\{x : a \leq x \leq b\}$ with $a = b$, is a single point. We write $\mathcal{A} = (A, \leq, \ldots)$ to imply that the τ-structure $\mathcal{A}$ has a total ordering relation $\leq$ and other elements in its structure.

Definition 2.1 (o-minimal): We say that a totally ordered structure $\mathcal{A} = (A, \leq, \ldots)$ is *o-minimal* (order-minimal) if every definable subset[3] of A can be expressed as a *finite* union of intervals [13].

Some examples of o-minimal structures are

- $(\mathbb{R}, <, +, \times, \mathtt{exp})$, and
- $(\mathbb{R}, <, +, \times)$,

where $+, \times, \mathtt{exp}$ are the addition, multiplication and exponentiation operations on reals, respectively. Additional examples can be found in [14], [13]. The theory of $(\mathbb{R}, <, +, \times)$ is known to be decidable [15]. We call anything defined in this structure *semi-algebraically definable*.

B. Transition System, Simulation and Bisimulation

A binary relation R on a set A is a subset of $A \times A$. We denote $(a, b) \in R$ by aRb . A binary relation R on a set A that is reflexive (aRa), symmetric ($aRb \Leftrightarrow bRa$) and

[1]Finitary functions include nullary functions which are constants.

[2]A (partial) function can be represented as a relation corresponding to its graph, for example, the function, which maps a real number x to its positive square root, has as its graph the relation $(x, \sqrt{x}) := \{(x, a) : a^2 = x \wedge a \geq 0\}$.

[3]Note that a definable set here refers to subset of A, which is essentially a unary relation on A.

transitive ($aRb \wedge bRc \Rightarrow aRc$) is called an *equivalence relation*. An equivalence relation partitions the set A into *equivalence classes*: $[a]_R = \{b \in A \mid aRb\}$. A partition Π of the set A defines a natural equivalence relation $\equiv_\Pi$, where $a \equiv_\Pi b$ iff a and b belong to the same partition in Π. In this article, when we refer to a partition Π of a set S, we essentially mean an equivalence relation on S. Hence we will use the terms equivalence relation and partition interchangeably. Finally, we will say an equivalence relation R_1 *refines* another equivalence relation R_2 iff $R_1 \subseteq R_2$.

A *transition system* is given by a tuple $\mathcal{S} = (Q, Q^0, \rightarrow)$, where Q is a set of states, $Q^0 \subseteq Q$ is the set of initial states, and $\rightarrow \subseteq Q \times Q$ is the transition relation. Given a transition system $\mathcal{S} = (Q, Q^0, \rightarrow)$, a *simulation relation* is a binary relation $R \subseteq Q \times Q$ so that, if $(q_1, q_1') \in R$ and $q_1 \rightarrow q_2$, there is q_2' such that $q_1' \rightarrow q_2'$ and $(q_2, q_2') \in R$.

Definition 2.2 (bisimulation): A relation R is said to be a *bisimulation* iff both R and R^{-1} are simulation relations. A state q_1 is said to be *bisimilar* to q_2 when there is a bisimulation relation R such that $(q_1, q_2) \in R$. This is denoted by $q_1 \cong q_2$.

A bisimulation R is said to *respect* a partition $\mathcal{P}$ iff R refines the equivalence relation defined by $\mathcal{P}$. By convention, *bisimilarity* with respect to a partition $\mathcal{P}$, is an equivalence relation on Q which is also the largest (or coarsest) bisimulation relation on the system which respects the partition $\mathcal{P}$ [16]. It is said to be of *finite* index, or simply finite, if it has finitely many equivalence classes.

III. Bounded Discrete Horizon Hybrid Automata

Hybrid automata is a popular formalism to model systems with mixed-discrete continuous behaviors, namely, hybrid systems. We define *hybrid automata* [4] (HA) below.

Definition 3.1: A *hybrid automaton* $\mathcal{H}$ is a tuple (Loc, $Edge$, $Cont$, $Cont_0$, Loc_0, Inv, $Flow$, $Guard$, $Reset$) where

- Loc is a finite set of *control states*, also called *discrete states*.
- $Edge \subseteq Loc \times Loc$ is the set of edges between control states.
- $Cont = \mathbb{R}^n$ is the domain of the *continuous (part of the) state*. (Here n is called the dimension of the hybrid automaton).
- $Cont_0 \subseteq Cont$ is the set of *initial continuous states*.
- $Loc_0 \in Loc$ is the *initial control state*.
- $Inv : Loc \rightarrow 2^{Cont}$ is the function that associates with every control state an *invariant*.
- $Flow : Loc \times Cont \rightarrow (\mathbb{R}_{\geq 0} \rightarrow Cont)$ associates with each $(q, x) \in Loc \times Cont$ a *flow function* that describes how the continuous state changes with time.
- $Guard : Edge \rightarrow 2^{Cont}$ assigns to each edge a *guard*, which is a continuous state constraint that must hold in order to take the discrete transition.
- $Reset : Edge \rightarrow 2^{Cont \times Cont}$ associates with each edge a *reset*, a binary relation between the current and new continuous state at the point where a discrete transition is taken.

A. Conventions

The pair *Loc* and *Edge* form what is commonly called the *control graph*. The elements of $Loc \times Cont$ are called *(hybrid) states*. In our notation, for readability, we will often put the argument of a function as a subscript. Specifically, $\mathcal{I}_q := \mathcal{I}(q)$, $\mathcal{G}_{(p,q)} := \mathcal{G}(p, q)$ and also $\mathcal{R}_{(p,q)} := \mathcal{R}(p, q)$.

We also assume that the flow functions satisfy the *semi-group* property:

1) $Flow_{(q,x)}$ is continuous and $Flow_{(q,x)}(0) = x$.
2) for every $t \geq 0$ and $x' \in Cont$, if $Flow_{(q,x)}(t) = x'$ then for every $t' \geq 0$, we have that $Flow_{(q,x)}(t + t') = Flow_{(q,x')}(t')$.

We say that such flows have *time-independent semi-group* (TISG) property. This property is (roughly) ensured by specifying the continuous dynamics by a time-independent differential equation. A TISG flow would then be a solution to such a differential equation. Although it is often common to use differential equations to define the continuous dynamics of a hybrid automaton, we have chosen a definition, in this article, that uses the solutions instead, for reasons that will be made obvious later. Note that, in general, TISG flows are not required to have continuous derivatives.

B. Hybrid Automata Semantics

The behavior of a hybrid system modeled by a hybrid automaton $\mathcal{H}$ is studied by considering what is called the *semantics* of a hybrid automaton. The semantics of a hybrid automaton $\mathcal{H}$ is a transition system $[\![\mathcal{H}]\!] = (Q, Q_0, \rightarrow)$, where

- $Q = Loc \times Cont$ is the set of states.
- $Q_0 = Loc_0 \times Cont_0$ is the set of initial states.
- Its transition relation $\rightarrow$ is the union of *time transitions* $\rightarrow_t$ and discrete transitions $\rightarrow_d$ given by
  - $(q_1, x_1) \rightarrow_t (q_2, x_2)$ iff $q_1 = q_2 = q$ and $\exists t \in \mathbb{R}_{\geq 0}$ such that $x_2 = Flow_{(q,x_1)}(t)$ and $\forall t' \in [0, t]$, $Flow_{(q,x_1)}(t') \in Inv_q$.
  - $(q_1, x_1) \rightarrow_d (q_2, x_2)$ iff there is an edge $(q_1, q_2) \in Edge$ such that $x_1 \in Inv_{q_1}$, $x_2 \in Inv_{q_2}$, $x_1 \in Guard_{(q_1,q_2)}$, and $(x_1, x_2) \in Reset_{(q_1,q_2)}$.

We will use $[\![\mathcal{H}]\!]_t$ to denote the restriction of the transition system $[\![\mathcal{H}]\!]$ to its continuous transitions, that is, $[\![\mathcal{H}]\!]_t = (Q, Q_0, \rightarrow_t)$.

In words, during a time transition, the discrete part q_1 does not change, but the continuous part of the state evolves according to the flow $Flow_{(q_1,x_1)}$ and always remains within the invariant Inv_{q_1}. During a discrete transition, the control state changes according to an edge (q_1, q_2) in the control graph, such that just before the transition is taken the continuous part of the state is satisfies the guard associated with (q_1, q_2). In addition, the result of taking the transition changes (or "resets") the continuous state to a new continuous state such that the pair of continuous states (before and after the transition) satisfy the reset condition associated with the edge.

An *execution* starting from a state (q, x) is a sequence of states (q_1, x_1), (q_2, x_2), $\ldots$, (q_k, x_k) such that $(q_1, x_1) = (q, x)$, and $\forall i < k$, $(q_i, x_i) \rightarrow (q_{i+1}, x_{i+1})$. Note that a prefix of an execution is also an execution.

Let us denote the number of discrete transitions in an execution $e = (q_1, x_1), (q_2, x_2), \ldots, (q_k, x_k)$ by $\texttt{NumDisTran}(e)$, where:

$$\texttt{NumDisTran}(e) := |\{i : 0 < i < k \wedge (q_i, x_i) \rightarrow_d (q_{i+1}, x_{i+1})\}|$$

If there exists an execution $(q_1, x_1), (q_2, x_2), \ldots, (q_k, x_k)$ then (q_k, x_k) is said to be *reachable* from (q, x). For a HA $\mathcal{H}$, we say that a control state q is *reachable*, if for some $x \in$ *Cont*, (q, x) is reachable from an initial state (q_0, x_0). Given a hybrid automaton $\mathcal{H}$, the *reachability problem* is to determine whether a given control state is reachable or not. Similarly, the "state-to-state" reachability problem is to decide if a given state (q_2, x_2) is reachable from a given state (q_1, x_1). Also, given two sets of states $R_1, R_2 \subseteq$ *Loc* $\times$ *Cont*, deciding whether there exists some state in R_2 which is reachable from some state in R_1, is called the "region-to-region" reachability problem.

C. Definability in an o-minimal structure

A hybrid system $\mathcal{H}$ is said to be *definable in a structure* $\mathcal{A} = \{A, \leq, \ldots\}$ or simply $\mathcal{A}$-definable, if all its initial conditions, invariants, flows, resets and guards are definable in $\mathcal{A}$. When $\mathcal{A}$ is some o-minimal structure then $\mathcal{H}$ is said to be *o-minimally definable*.

Remark 3.1: The term *o-minimal hybrid automata* in previous literature [1] refers to hybrid automata as defined above with the additional restriction that all resets are *strong*. This means that for any edge (p, q) the reset $\mathcal{R}_{(p,q)}$ is of the form $\mathcal{G}_{(p,q)} \times$ *Cont*$'$ for some *Cont*$' \subseteq$ *Cont*. This decouples the system into separate dynamical systems, with the discrete transitions nondeterministically placing the continuous state on some set at each discrete step. Our result does not require this decoupling, i.e. we do not require strong resets but we make use of o-minimal definability, therefore we have chosen a slightly different nomenclature and call our systems "o-minimally definable" as above.

D. Bounded-discrete-horizon Hybrid Automata

From now on we will consider HA for which there is an upper bound $\delta_{max} < \infty$ on the number of discrete transitions in any execution. Bounded discrete horizon hybrid automata are a class of hybrid automata which are similar in spirit to bounded model-checking which is the problem of deciding if every path whose length is within a given bound satisfies a property [17]. As we will see later, various natural restrictions on the class of hybrid automata lead to bounded discrete horizon hybrid automata.

Definition 3.2: A HA $\mathcal{H} =$ (*Loc*, *Edge*, *Cont*, *Cont*$_0$, *Loc*$_0$, *Inv*, *Flow*, *Guard*, *Reset*) has a *bounded discrete horizon* if there exists a positive integer δ_{max} such that all its executions have at most δ_{max} discrete transitions, that is, for every execution e, $\texttt{NumDisTran}(e) \leq \delta_{max}$.

IV. Finite Bisimulation Result

We are now ready to present our main result which follows after some facilitating definitions. Let us fix a hybrid automaton $\mathcal{H} =$ (*Loc*, *Edge*, *Cont*, *Cont*$_0$, *Loc*$_0$, *Inv*, *Flow*, *Guard*, *Reset*) for the rest of this section.

Definition 4.1: For partition $\mathcal{V}$ of *Loc* $\times$ *Cont* (recall that $\mathcal{V}$ is an equivalence relation on *Loc* $\times$ *Cont*), define $F_t^\star(\mathcal{V})$ to be the largest bisimulation of $[\![\mathcal{H}]\!]_t$ that respects $\mathcal{V}$. Also, let $F_d(\mathcal{V}) := \{(s_1, s_2) | \big(\exists s_1' \,.\, s_1 \rightarrow_d s_1'\big) \Rightarrow \big(\exists s_2' \,.\, s_2 \rightarrow_d s_2' \wedge s_1' \mathcal{V} s_2'\big), \big(\exists s_2' \,.\, s_2 \rightarrow_d s_2'\big) \Rightarrow \big(\exists s_1' \,.\, s_1 \rightarrow_d s_1' \wedge s_1' \mathcal{V} s_2'\big)\} \cap \mathcal{V}$. Observe that $F_d(\mathcal{V})$ is an equivalence relation when $\mathcal{V}$ is. In fact, $F_d(\mathcal{V})$ is the largest refinement of $\mathcal{V}$ such that for any equivalence class P of $F_d(\mathcal{V})$, the set of states reached by a single discrete transition from P are contained in some equivalence class of $\mathcal{V}$. In contrast F_t^* denotes the refinement of $\mathcal{V}$ which has the above property with respect to arbitrary number of continuous transitions, hence the $*$.

Definition 4.2: For a hybrid system, we define the i-th neighborhood $N_i \in$ *Loc* $\times$ *Cont* to be the set of all states starting from which there is no execution that can have more than i discrete transitions. Note that $N_{i+1} \supseteq N_i$.

An example of a state belonging to an i-th neighborhood is given in Section VII-C.

The following are some observations about the functionals $F_t^\star$ and F_d.

Proposition 4.1: (a) The functionals $F_t^\star(\cdot)$ and $F_d(\cdot)$ are both monotonic, that is, given partitions $\mathcal{V}_1 \subseteq \mathcal{V}_2$, $F_t^\star(\mathcal{V}_1) \subseteq F_t^\star(\mathcal{V}_2)$ and $F_d(\mathcal{V}_1) \subseteq F_d(\mathcal{V}_2)$;

(b) $F_t^\star(\mathcal{V})$ is a refinement of $\mathcal{V}$ and so is $F_d(\mathcal{V})$, i.e., $F_t^\star(\mathcal{V}) \subseteq \mathcal{V}$ and $F_d(\mathcal{V}) \subseteq \mathcal{V}$;

(c) $F_t^\star(\cdot)$ is idempotent, i.e., $F_t^\star(F_t^\star(\mathcal{V})) = F_t^\star(\mathcal{V})$.

Note that the idempotency of $F_t^\star$ follows from the fact that $F_t^\star(\mathcal{V})$ is the coarsest bisimulation of $[\![\mathcal{H}]\!]_t$ respecting $\mathcal{V}$, and since $F_t^\star(\mathcal{V})$ is a bisimulation of $[\![\mathcal{H}]\!]_t$, the coarsest bisimulation of $[\![\mathcal{H}]\!]_t$ respecting $F_t^\star(\mathcal{V})$ is $F_t^\star(\mathcal{V})$ itself.

Lemma 4.2: Le $\mathcal{H} =$ (*Loc*, *Edge*, *Cont*, *Cont*$_0$, *Loc*$_0$, *Inv*, *Flow*, *Guard*, *Reset*) be a hybrid automaton o-minimally definable in $\mathcal{A}$ and let $\mathcal{P}$ be an $\mathcal{A}$-definable partition of its state space *Loc* $\times$ *Cont*. Let $\cong$ be the coarsest bisimulation (which need not be of finite index) relation on $\mathcal{H}$ refining $\mathcal{P}$. Define a sequence of partitions $\{W_0, W_1, \ldots\}$ inductively by setting $W_0 = F_t^\star(\mathcal{P})$ and $W_{i+1} = F_t^\star(F_d(W_i))$. The following hold for all $i \geq 0$:

(a) W_i is a finite partition definable in the o-minimal theory.

(b) $\cong \subseteq W_i$.

(c) W_i is a bisimulation on the i-th neighborhood N_i and refines $\mathcal{P}$.

Proof: We prove the claims one after the other.

Claim(a): The proof follows by an induction on i. We know that for any finite partition $\mathcal{P}$ definable in the o-minimal structure $\mathcal{A}$,$F_t^\star(\mathcal{P})$ has only finitely many equivalence classes and is definable in the o-minimal theory [18]. In particular, Theorem 12.3.5 in [19] shows that $F_t^\star(\mathcal{P})$ is of finite index when the dynamics satisfies a "suffix determinism" property, which is trivially true in our case due to the semi-group property which ensures that the suffix of any flow passing through a continuous state x starting from x is the same.

The above result uses the o-minimality of the structure $\mathcal{A}$. Therefore, W_0 is o-minimal definable; hence, the claim is true for $i = 0$. It is easy to see from the definition of F_d that it is also o-minimal definable. In addition, for any partition

$\mathcal{P}$ which is definable in the o-minimal theory, $F_d(\mathcal{P})$ has finitely many equivalence classes because there are finitely many discrete transitions possible from each part of $\mathcal{P}$. Thus, from these observations we have W_{i+1} is definable in the o-minimal theory and has finitely many equivalence classes, if W_i does, since $W_{i+1} = F_t^\star(F_d(\mathcal{P}))$.

Claim (b): Proof by induction on i.

Case $i = 0$: From the definition of $\cong$, $\cong \subseteq \mathcal{P}$. Since $F_t^\star(\cdot)$ is monotonic (from Proposition 4.1 (a)), $F_t^\star(\cong) \subseteq F_t^\star(\mathcal{P})$. But since $F_t^\star(\cong) = \cong$ (since $\cong$ is a bisimulation on $[\![\mathcal{H}]\!]$, it is also a bisimulation on $[\![\mathcal{H}]\!]_t$), we have $\cong \subseteq F_t^\star(\mathcal{P}) = W_0$. Hence $\cong \subseteq W_0$.

Case $i \geq 1$: By induction hypothesis $\cong \subseteq W_{i-1}$. By monotonicity of the functionals $F_t^\star(\cdot)$ and $F_d(\cdot)$ (from Proposition 4.1(a)), we have $F_t^\star(F_d(\cong)) \subseteq F_t^\star(F_d(W_{i-1}))$. But since $\cong$ is a bisimulation, $F_t^\star(F_d(\cong)) = \cong$. Hence $\cong \subseteq F_t^\star(F_d(W_{i-1}))$ and therefore $\cong \subseteq W_i$.

Claim (c): We will prove the claim by induction on i.

Case $i = 0$: Let $(q, x), (p, y) \in N_0$, and $(q, x)W_0(p, y)$. Suppose $(q, x) \rightarrow (q_1, x_1)$. Since $(q, x) \in N_0$, it cannot take a discrete transition, hence $(q, x) \rightarrow_t (q_1, x_1)$. But since $W_0 = F_t^\star(\mathcal{P})$, $(q, x)F_t^\star(\mathcal{P})(p, y)$ and hence there exists (p_1, y_1) such that $(p, y) \rightarrow_t (p_1, y_1)$ and $(q_1, x_1)W_0(p_1, y_1)$. Therefore (p, y) simulates (q, x). We can argue similarly that (q, x) simulates (p, y). Hence W_0 is a bisimulation on N_0 refining $\mathcal{P}$.

Case $i > 1$: By induction hypothesis W_i is a bisimulation on all states in N_i refining $\mathcal{P}$. We need to prove that W_{i+1} is a bisimulation relation on all states in N_{i+1}. W_{i+1} is a refinement of $\mathcal{P}$ since W_i is a refinement of $\mathcal{P}$ and $W_{i+1} = F_t^\star(F_d(W_i))$. Given two states (q, x) and (p, y) in N_{i+1} that satisfy $(q, x)W_{i+1}(p, y)$, we will prove that (p, y) simulates (q, x) and by symmetry, the reverse will also be true.

(1) Suppose $(q, x) \rightarrow_t (q_1, x_1)$. Since $W_{i+1} = F_t^\star(F_d(W_i))$, we know that there is a (p_1, y_1) such that $(p, y) \rightarrow_t (p_1, y_1)$ and $(q_1, x_1)W_{i+1}(p_1, y_1)$. In addition both (q_1, x_1) and (p_1, y_1) will still be in N_{i+1} since no discrete transition has occurred.

(2) Suppose $(q, x) \rightarrow_d (q_1, x_1)$. Since $W_{i+1} = F_t^\star(F_d(W_i)) \subseteq F_d(W_i)$, we have $(q, x)W_{i+1}(p, y)$ implies $(q, x)F_d(W_i)(p, y)$. Thus by the definition of F_d, we know that there is a (p_1, y_1) such that $(p, y) \rightarrow_d (p_1, y_1)$ and $(q_1, x_1)W_i(p_1, y_1)$. Note that both (q_1, x_1) and (p_1, y_1) will now be in N_i. Since W_i and W_{i+1} agree on N_i (follows from the fact that all transitions starting from a particular N_i go into N_i) we have $(q_1, x_1)W_{i+1}(p_1, y_1)$.

Therefore W_{i+1} is a bisimulation relation on all states in N_{i+1} and refines $\mathcal{P}$. This concludes the induction proof of Claim (c).

Theorem 4.3: Given a HA $\mathcal{H}$ which is definable in some o-minimal structure $\mathcal{A}$, with a δ_{max}-bounded discrete horizon and given a $\mathcal{A}$-definable partition $\mathcal{P}$ of its state space, the transition system of $\mathcal{H}$ has a finite bisimulation that respects $\mathcal{P}$. In addition, if $T(\mathcal{A})$ is decidable, then the bisimulation can be constructed.

Proof: By Lemma 4.2, $W_{\delta_{max}}(\mathcal{P})$ is a bisimulation on the δ_{max}-th neighborhood, $N_{\delta_{max}}$. However, due to the bounded horizon property of $\mathcal{H}$, $N_{\delta_{max}}$ is all of the state space. By the same lemma, $W_{\delta_{max}}(\mathcal{P})$ is definable partition of finite and it respects $\mathcal{P}$. If $\mathcal{A}$ is decidable, then there is an effective algorithm that constructs this finite bisimulation quotient, since for such theories one can determine if there is a transition between two partitions of the bisimulation quotient by reducing it to the problem of satisfiability of a formula in the logic. (Note that the one can effectively construct the formulas defining the partitions).

The bisimulation construction gives us the ability to verify temporal properties of a hybrid system that include reachability, safety, and others. Verifying these and other more complex properties is equivalent to verifying a CTL property on a bisimilar finite-state transition system, which is decidable [12]. It is also possible to verify simple properties by quantifier elimination of first logic formulas directly as we will see in an example in Section VII.

In the next two sections we review STORMED hybrid systems, a subclass briefly introduced in [8] and show how they satisfy the bounded discrete horizon property.

V. STORMED HYBRID SYSTEMS AND EXTENSIONS

STORMED hybrid systems [8] form a special subclass of hybrid automata in that their specifications arise from the design of engineering systems and at the same time they have nice decidability properties since they are o-minimally definable and have a bounded discrete horizon. Again, we start with some auxiliary definitions. We will use . to represent the standard vector dot product and $\|\ \|$ to denote the euclidean distance.

A. Separable guards

A hybrid system $\mathcal{H} = (\mathit{Loc}, \mathit{Edge}, \mathit{Cont}, \mathit{Cont}_0, \mathit{Loc}_0, \mathit{Inv}, \mathit{Flow}, \mathit{Guard}, \mathit{Reset})$ has *separable guards* if $\exists\ d_{min} > 0$ such that $\forall\ (p_1, q_1), (p_2, q_2) \in \mathit{Edge}$, where $p_1 \neq p_2$, we have $\mathit{min}\{\|x_1 - x_2\| : x_1 \in \mathcal{G}_{(p_1,q_1)}\ \wedge\ x_2 \in \mathcal{G}_{(p_2,q_2)}\} \geq d_{min}$. The guards are said to be d_{min}-separable.

This property, coupled with equality resets along discrete jumps, essentially helps remove Zeno behavior, avoiding unbounded number of discrete steps in finite time. Note that guards originating from the same discrete state are not required to be separable, thus allowing non-determinism. Technically we will see that equality resets are not required when more constraints are added.

In embedded and digital systems, separability of guards is a natural consequence of control signals, which often have a minimum actuation or sensing period, or reaction time, due to the use of microprocessors and digital clocks.

B. Monotonic Flows and Resets

The flow $\mathcal{F}$ of $\mathcal{H}$ is *monotonic* with respect to some vector $\phi \in \mathit{Cont}$, if $\exists\ \epsilon > 0$ such that $\forall\ q \in \mathit{Loc}, x \in \mathit{Cont}$, and $\forall\ t, \tau \geq 0$,

$$\phi \cdot (\mathit{Flow}_{(q,x)}(t + \tau) - \mathit{Flow}_{(q,x)}(t)) \geq \epsilon \|\mathit{Flow}_{(q,x)}(t + \tau) - \mathit{Flow}_{(q,x)}(t)\|,$$

We call such a set of flows (ϵ, ϕ)-monotonic.

Monotonicity implies that as the continuous state evolves with time, the projection on ϕ increases at a minimum rate ϵ. This guarantees that the projection on ϕ will never decrease.

Some examples of monotonic flows are:

1) Linear flows of the form $\mathit{Flow}_{(q,x)}(t) = x + \alpha_q(t)$, where $x \in \mathbb{R}^n$, and $\alpha_q \in (\mathbb{R}_{\geq 0} \ -\{0\})^n$.
2) Analytic flows with their time-derivatives ranging on only one half-space, i.e., there exists a ϕ such that for all $q \in \mathit{Loc}$ and $x \in \mathit{Cont}$, we have $\nabla_t \mathit{Flow}_{(q,x)}(t) \cdot \phi > \epsilon \|\nabla_t \mathit{Flow}_{(q,x)}(t)\|$.

Monotonic flows appear in the form of Lyapunov function decrement, energy depletion, progress towards a goal, time passing,

The reset function $\mathcal{R}$ of $\mathcal{H}$ is said to be *monotonic* with respect to some $\phi \in \mathit{Cont}$, if $\exists\ \epsilon, \zeta > 0$ such that $\forall\ (p, q) \in \mathit{Edge}$ and $x_1, x_2 \in \mathit{Cont}$ s.t. $(x_1, x_2) \in \mathcal{R}_{(p,q)}$, we have:

(i) if $p = q$, then either $x_1 = x_2$ or $\phi \cdot (x_2 - x_1) \geq \zeta$, and
(ii) If $p \neq q$, then $\phi \cdot (x_2 - x_1) \geq \epsilon \|x_2 - x_1\|$.

Such a reset function is called (ϵ, ζ, ϕ)-monotonic.

Remark 5.1: An important fact is that monotonicity and separability are $\mathcal{A}$-definable properties of an $\mathcal{A}$-definable hybrid automaton and can be verified for decidable theories automatically.

Remark 5.2: When the discrete state changes, it is not required to move the continuous state along ϕ by any minimum value. Note that we also allow identity resets.

One can see that, when the flow of the hybrid systems is (ϵ, ϕ)-monotonic, resets are (ϵ, ζ, ϕ)-monotonic and its guards are d_{min}-separable, a minimum distance of $\min\{\zeta, \epsilon d_{min}\}$ has to be traveled along ϕ between two successive discrete transitions. The only exception is when the discrete state does not change and the reset is the identity map. We call such a transition *trivial* since the system behaves as if it was never taken. In all other cases, condition the constraints avoid Zeno behaviors in a discrete self-loop, and ensure that we cannot have infinitely fast switching along ϕ when the guards are separable. This will be proved in the next section.

Definition 5.1 (STORMED Hybrid Systems): A *STORMED hybrid system* is a tuple $(\mathcal{H},\ \mathcal{A},\ \phi,\ b_-,\ b_+,\ d_{min},\ \epsilon,\ \zeta)$, where $\mathcal{H} = (\mathit{Loc},\ \mathit{Edge},\ \mathit{Cont},\ \mathit{Cont}_0,\ q_0,\ \mathit{Inv},\ \mathit{Flow},\ \mathit{Guard}, \mathit{Reset})$ is a hybrid automaton, $\mathcal{A}$ is an o-minimal structure, $b_-, b_+, d_{min}, \epsilon, \zeta \in \mathbb{R}$, and $\phi \in \mathit{Cont}$ is a vector such that the following conditions are satisfied:

- **(S)** All guards of $\mathcal{H}$ are d_{min}-**S**eparable.
- **(T)** All flows of $\mathcal{H}$ satisfy the **T**ime-Independent Semi-Group property mentioned earlier (TISG).
- **(O)** $\mathcal{H}$ is definable in the **O**-minimal structure $\mathcal{A}$.
- **(RM)** **R**esets and flows are **M**onotonic: Resets are (ϵ, ζ, ϕ)-monotonic and flows are (ϵ, ϕ)-monotonic respectively.
- **(ED)** **E**nds are **D**elimited: for all $(p, q) \in \mathit{Edge}$ we have $\{\phi \cdot x : x \in \mathit{Guard}_{(p,q)} \in (b_-, b_+)$, meaning that the projection of each of the guard sets on ϕ is bounded below by (or is greater than) b_- and bounded above by (or is less than) b_+.

Remark 5.3: Consider a hybrid automaton system with a 1-dimensional continuous state space and N discrete states. If the guards are defined as $\forall 1 < i, j < N \quad \mathit{Guard}_{(i,j)} = [3i, 3i+1]$ then they are separable by unit distance. In addition, since $N < \infty$ for hybrid automata, we have ends are delimited by $[0, 3N+3]$. On the other hand, if the guards were defined as $\forall 1 < i, j < N \quad \mathit{Guard}_{(i,j)} = [3i, 3i+3)$ then they would not be separable. Now let us consider another hybrid automaton with a 2-dimensional continuous state space An example of o-minimally definable monotonic flows could be

$$\mathit{Flow}_1(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, t) = \begin{bmatrix} x_1 + 3t \\ x_2 + 2t \end{bmatrix}$$

,

$$\mathit{Flow}_2(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, t) = \begin{cases} \begin{bmatrix} x_1 + t \\ x_2 + 3t \end{bmatrix} & \text{when } x_1 > 0 \\ \begin{bmatrix} x_1 + t \\ x_2 \exp(2t) \end{bmatrix} & \text{otherwise} \end{cases}$$

which is monotonic along $\phi = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. However, if, for example

$$\mathit{Flow}_1(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, t) = \begin{bmatrix} x_1 + t \\ x_2 \end{bmatrix}, \mathit{Flow}_2(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, t) = \begin{bmatrix} x_1 \\ x_2 + 3t \end{bmatrix},$$

$$\mathit{Flow}_3(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, t) = \begin{bmatrix} x_1 \\ x_2 - 2t \end{bmatrix}, \mathit{Flow}_4(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, t) = \begin{bmatrix} x_1 \exp(t) \\ x_2 \end{bmatrix}$$

, then there is no monotonic direction for the system. In the latter case, it is even possible to increment and decrement the continuous variables independently, although this is not required to break monotonicity.

Lemma 5.1: Any STORMED hybrid system ($\mathcal{H}$, $\mathcal{A}$, ϕ, b_-, b_+, d_{min}, ϵ, ζ), has a ν-bounded discrete horizon, where $\nu = \lceil \frac{b_+ - b_-}{\min\{\zeta, \epsilon d_{min}\}} \rceil$

Proof: We can prove this by showing that there is a minimum distance the continuous part of the state travels along ϕ between two consecutive discrete transitions. Let the state be (q_1, x_1) from which a discrete transition is taken to (q_2, x_2). Then a continuous transition is taken to (q_3, x_3) from which a discrete transition is taken. Assume all the discrete transitions are non-trivial. We will show that $\phi \cdot (x_3 - x_1) \geq \mathit{min}(\zeta, \epsilon d_{\mathit{min}})$. $\phi \cdot (x_2 - x_1) \geq \mathit{min}(\zeta, \epsilon \|x_2 - x_1\|)$ by monotonicity of resets. Further, $\phi \cdot (x_3 - x_2) \geq \epsilon \|x_3 - x_2\|$ by monotonicity of flows. Hence $\phi \cdot (x_3 - x_1) = \phi \cdot (x_2 - x_1) + \phi \cdot (x_3 - x_2) \geq \mathit{min}(\zeta, \epsilon \|x_2 - x_1\|) + \epsilon \|x_3 - x_2\| \geq \mathit{min}(\zeta, \epsilon(\|x_2 - x_1\| + \|x_3 - x_2\|))$. By triangle inequality, we have $\|x_2 - x_1\| + \|x_3 - x_2\| \geq \|x_3 - x_1\|$. Hence $\phi \cdot (x_3 - x_1) \geq \mathit{min}(\zeta, \epsilon \|x_3 - x_1\|)$. Therefore, the minimum distance traveled along ϕ between two discrete jumps is at least $\eta := \min\{\zeta, \epsilon d_{min}\}$. Since, by the ends-delimited property, we can only have discrete transitions from $\{(q, x) : \phi \cdot x \in (b_-, b_+)\}$, we can conclude that we can have at most $\nu = \lceil \frac{b_+ - b_-}{\eta} \rceil = \lceil \frac{b_+ - b_-}{\min\{\zeta, \epsilon d_{min}\}} \rceil$ discrete transitions in any execution.

Theorem 5.2: Verification of a $CTL^\star$ property ϕ on the semantics of a hybrid automaton $\mathcal{H}$ with STORMED specifications is decidable.

Proof: Given STORMED parameters ($\mathcal{H}$, $\mathcal{A}$, ϕ, b_-, b_+, d_{min}, ϵ, ζ), since $\mathcal{H}$ has at a discrete horizon bound of $\delta_{max} := \lceil \frac{b_+ - b_-}{\min\{\zeta, \epsilon d_{min}\}} \rceil$ we form a finite bisimulation $W_{\delta_{max}}(\mathcal{P})$ where

$\mathcal{P}$ is a partition of $Loc \times Cont$ based on the atomic propositions of the formula. By the decidability of $\mathcal{A}$, this is constructable. Standard finite transition system methods [12], [20] verify the property on a bisimilar finite automaton.

Remark 5.4: Linear differential equations (LDEs) are the most common model for autonomous dynamical systems used in control theory and similar fields. Although there are only a few subclasses of such systems that give rise to o-minimal trajectories, two main facts indicate that our specifications can be theoretically appropriate for such systems:

1) trajectories generated by LDEs become o-minimal when time-restricted [14]
2) successive approximations can be used to generate o-minimal trajectories from general LDE solutions with reachability still verifiable for over-approximate dynamics.

C. Undecidable relaxations

We justify the tightness of the geometric STORMED constraints by showing that relaxing any of them renders the reachability problem undecidable. The semi-group property of the flows and o-minimal definability of the system are assumed to be intrinsic to the model.

We will incorporate the methods from [3] to show our undecidability result, so we first outline the proof of the undecidability of the reachability problem of multirate timed automata. That proof will be then modified for our purposes. We prove undecidability of the properties of STORMED h.s. with relaxed constraints by reducing the equivalent problem on a two counter-machines to one on a multirate automata which satisfy the all STORMED specifications except the relaxed one. From basic automata theory [21] we know that any property of a language accepted by a two counter machine, i.e. any property of its executions, is undecidable.

Consider a two counter-machine M with counters C and D. The multirate automaton A simulating it, has two variables x and y storing the values corresponding to the values of the counters. For a counter value of n the corresponding variable value is $1/2^n$. A counter increment therefore will halve the value of the variable and similarly a decrement will double the value. M will have an i-th configuration at location p with counter values m and n whenever A at time $2i$ is in state p with with counter values $1/2^m$ and $1/2^n$. In figure 1 we depict the increment, decrement, and test-for-0 for the automaton. Variable g keeps track of the global time. The values of the variable flows not shown are by default assumed to be 0.

In the following, STOED hybrid systems is the class of hybrid systems whose definition is similar to the definition of STORMED hybrid systems, except that it does not contain the constraint corresponding to RM, and similarly, STORM is the class of hybrid systems which excludes the constraint ED from the definition of STORMED. Note that the system in Fig. 1 is missing the monotonicity property since the variables x and g are reset to zero at various locations, and therefore there is no monotonic progress along any direction in those two dimensions.

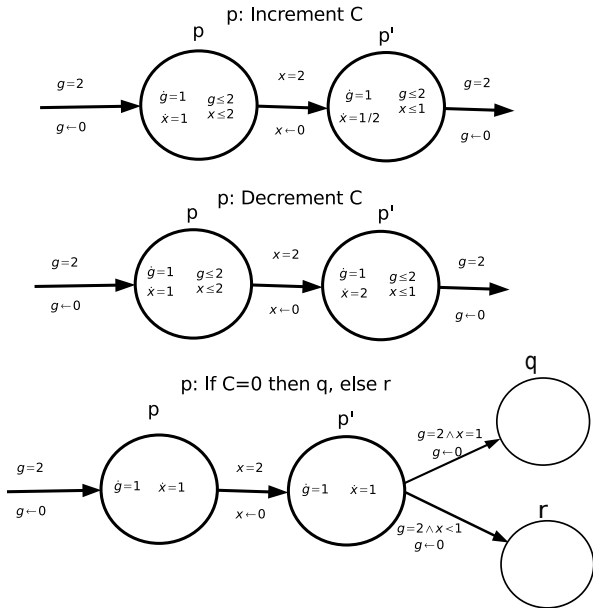

Fig. 1. The parts of the multirate automaton A corresponding to the operations increment, decrement and test for zero of the two-counter machine M.

Theorem 5.3: The reachability problem for STORMED hybrid systems becomes undecidable when monotonicity is removed, that is, the reachability problem is undecidable for the class of STOED hybrid systems.

Proof: From automaton A, which satisfies all the STORMED constraints except monotonicity and separable guards, we will obtain automaton A_s which has separable guards, in order to prove this theorem. Note that when an automaton is not guaranteed to be monotonic in any direction, there is no use of delimiting its guards or invariants. However, the rest of the STORMED specifications will remain, and our reduction, as in the case of A, proves undecidability. For this we associate an even number h_q with each state q. We add a new variable v, and a constraint $v \in (h_p, h_p + 1]$ in the transition going out of p. When there is only a single transition out of p', we add the constraint $v \in (h_{p'}, h_{p'} + 1]$ to its guard; otherwise, we add the constraint $v = h_{p'} + 1$ to $p' \to q$, and the constraint $v \in (h_{p'}, h_{p'} + 1/2]$ to $p' \to r$. We add 3 more variables g', x' and y' whose values equal those of g, x and y, respectively, when entering any state. However, the values of x' and y' do not change while in state p and the value of g' does not change in state p'. It is easy to see that this can be ensured by treating the variables x', y' and g' similar to x, y and g respectively everywhere, except that in state p, $\dot{x}' = 0$ and $\dot{y}' = 0$ and in state p', $\dot{g}' = 0$. Finally we set $\dot{v} = h_p/(2 - x') + x'/(2 - x')$ in state p corresponding to an operation on counter C. In state p' we set $\dot{v} = h_{p'}/(2 - g') + g'/(2 - g')$. The value of v upon exiting p would therefore be $h_p + x_0$ and upon exiting p' would be $h_{p'} + x_0$, where x_0 is the value of x when entering p. This is depicted in figure 2. The transitions that are enabled in A_s are always the same as in A, therefore the undecidability proof for A is applicable here, which completes our proof.

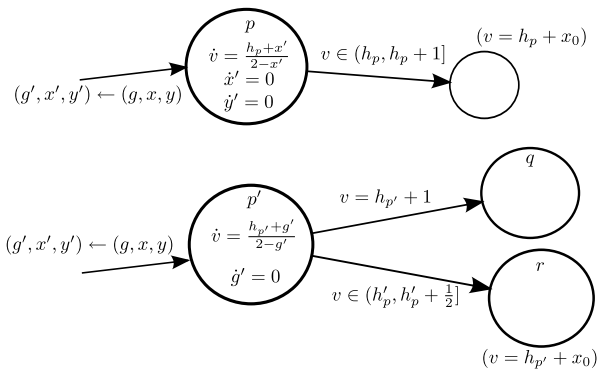

Fig. 2. The automaton A_s state transitions with most important assignments guards and invariants shown. Separability of guards is on v. Variable v becomes equal to $h_p + x_0$ upon leaving the state, where x_0 is the value of x when entering.

Theorem 5.4: The reachability problem for STORMED hybrid systems becomes undecidable when the "ends limited" constraint is removed, that is, the reachability problem is undecidable for the class of STORM hybrid systems.

Proof: Starting from automaton A_s we construct now an automaton A_u which establishes the monotonicity of resets by adding a new variable n which increases monotonically at rate 1. However, its "ends" will no longer be "delimited". The monotonicity is now along the flow of n.

Relaxing combinations of the STORMED constraints causes undecidability at very low dimensions. Without separability of guards and ends-delimited, we have undecidability in four dimensions. This follows from the results of [6] where piecewise constant derivative systems (PCDs) with delimited ends in three dimensions are shown undecidable. PCD flows are not monotonic, but they can be made monotonic by introducing a fourth dimension along which the flows are monotonic. The results in [6] also imply that the reachability problem for STORMED hybrid systems without guard separability or monotonicity is undecidable in three dimensions. With just the relaxation on separability of guards, it follows from the results in [22] that finite bisimulation does not exist even in two dimensions.

VI. Extensions

Even though the above relaxations cause undecidability, the conditions in the STORMED specification are not necessary for the existence of finite bisimulation. Relaxations of the conditions that force the number of discrete transitions in any execution to be bounded suffice for our proof of existence of finite bisimulation to go through. For example, assume that we have all guards bounded above by b_+, that is,

$$\exists b_+ \in \mathbb{R} \;.\; \forall (p,q) \in \textit{Edge} \quad \forall x \in \textit{Guard}_{(p,q)} \quad x_i < b_+.$$

Then instead of the guard separability and monotonicity conditions, all we require is that between any two consecutive edges the continuous state moves by a minimum distance towards b_+.

Also, the guard separability condition in the STORMED specification can be relaxed as follows: Instead of requiring the guards on any two edges to be separable, it is enough to have separability between the guards of edges which belong to the same maximal strongly connected component[4] of the underlying graph of the STORMED system. This is because the number of occurrences of the transitions between locations belonging to different strongly connected components in any path of the hybrid system can be bounded by the number of such transitions in the system. Hence we only need to bound the number of discrete transitions between the same component.

A. Realizations of STORMED specifications in physical systems

STORMED hybrid systems can describe many system models, and the constraints imposed by STORMED hybrid systems are realized in many physical systems as explained below.

Monotonicity can be associated with energy or time depletion, or with non-decreasing trajectories in vehicle control problems. The *ends-delimited* property can be present as the actual deadline on the monotonic direction or a spatial confinement. *Separability of guards* represents infrequency in making control decisions, also based on location or time, and therefore it is an intrinsic feature of a system with constant sampling time decision making.

Time-independent vector field flows that satisfy the semigroup property arise naturally, as they appear whenever the continuous states are described as vector fields. Though *o-minimality* is not necessarily a common property, one can often find approximations of a system which are o-minimal [23]. Linearization and other model reductions may also result in o-minimal realizations. In general, the solutions of Ordinary Differential Equations, used to describe many systems, are not o-minimal trajectories. When a feedback law is used to force an open system to track a polynomial or exponential trajectory o-minimal flows arise on the closed system. Uncertainty in the model or in the initial condition can be represented by nondeterminism in a *reset*.

VII. Example

A. Description

This example examines a tracking controller for a tool-path on a 2D (x-y) robot. The motion of the tool-tip of a two-axis robot arm along a given path on a plane is controlled by motor drivers connected to a micro-controller. The micro-controller is given a path consisting of N 2D reference points $X_1, \cdots, X_N$ to track. For the sake of simplicity we assume that the distance between any two consecutive points is equal (See Figure 3). The tool-tip is initially at X_1. The micro-controller calculates an acceleration for the tool-tip to track X_2. The tool-tip then accelerates to a value depending on the acceleration computed by the micro-controller and a disturbance term. Further, a new acceleration is computed every T seconds after consulting a noisy reading of the current position of the tool-tip and its last

[4] A strongly-connected component of a graph is a subgraph in which any two vertices are connected to each other by paths (here this includes self-loops on vertices).

2 positions. When the tool-tip is within distance ϵ_x from X_2, the micro-controller switches to tracking X_3.

We would like to verify if the control law is good enough to follow the path closely within a time interval T_{total}. We will instead verify a sufficient condition for tracking[5]. Instead of checking the entire path explicitly, we will try to show that whenever the tool-tip is currently somewhere in the proximity of reference point X_i with velocity at most ϵ_v and acceleration at most ϵ_a, it will be within the same proximity of the next reference point X_{i+1} within time $T_P = T_{total}/(N-1)$, again with velocity at most ϵ_v and acceleration at most ϵ_a. Then by induction, all the reference points will be tracked in time less than T_{total}.

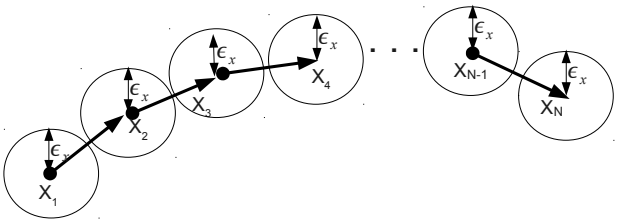

Fig. 3. N tracking points and ϵ balls around them.

B. Specification

We will model the above problem as a STROMED system. The continuous statespace is given by $\mathbb{R}^{17}$, where the components of a state can be represented as a tuple of 9 elements $(x, v, a, \hat{x}_0, \hat{x}_{-1}, \hat{x}_{-2}, \hat{a}, \hat{v}, \tau)$, where:

- $x \in \mathbb{R}^2$ is the current position of the tool-tip on the plane,
- $v \in \mathbb{R}^2$ is the current velocity of the tool-tip,
- $a \in \mathbb{R}^2$ is the current acceleration of the tool-tip,
- $\hat{x}_0, \hat{x}_{-1}, \hat{x}_{-2} \in \mathbb{R}^2$ are the last three sampled positions of the tool-tip,
- $\hat{a} \in \mathbb{R}^2$ is the last acceleration computed by the controller,
- $\hat{v} \in \mathbb{R}^2$ is the last velocity estimate,
- $\tau \in \mathbb{R}_{\geq 0}$ is the time-component.

We will use the time-component τ as our monotonic direction since it always increases and it is bounded by the deadline T_P. Since the acceleration is computed by the micro-controller every T seconds, there are at most P discrete transitions between two tracking points, where P is the largest integer less than or equal to T_P/T.

Now we will give a STORMED model capturing the behavior of the system between two consecutive points X_i and X_{i+1}. It consists of two states, namely, q_1 and q_2. There is a self-loop on q_1 corresponding to the micro-controller's acceleration computations every T seconds. Also there is transition from q_1 to q_2 when the tool-tip is within ϵ_x distance from the current tracking point.

Formally, the set of locations $Loc = \{q_1, q_2\}$, and the set of edges $Edge = \{(q_1, q_1), (q_1, q_2)\}$. The continuous state space is $Cont = \mathbb{R}^{17}$, and the initial location is q_1. Let $\mathcal{B}^c_\epsilon(X)$ and $\mathcal{B}^o_\epsilon(X)$ denote a closed and an open ball of radius ϵ around X respectively. The initial continuous state space $Cont_0$ is given by the constraint $\hat{x} \in \mathcal{B}^c_{\epsilon_x}(X_i) \wedge \hat{v} \in \mathcal{B}^c_{\epsilon_v}(0) \wedge \hat{a} \in \mathcal{B}^c_{\epsilon_a}(0)$. The invariants are given as follows:

- $Inv(q_1) := \tau \in [0, T_P] \wedge \neg(\hat{x} \in \mathcal{B}^o_{\epsilon_x}(X_{i+1}) \wedge \hat{v} \in \mathcal{B}^o_{\epsilon_v}(0) \wedge \hat{a} \in \mathcal{B}^o_{\epsilon_a}(0))$, and
- $Inv(q_2) := \top$.

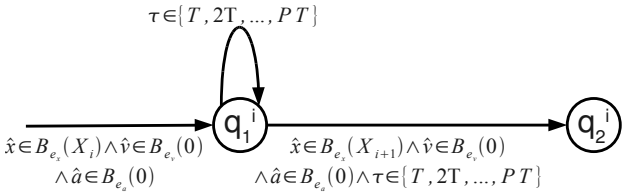

Fig. 4. The hybrid automaton of 2 discrete states, indexed by i, used to track reference point X_i. If the system is verified to be able to approach point X_{i+1} given that it has approached point X_i, then inductively it will track all points.

The guards are given by:

$$\begin{aligned} Guard_{(q_1,q_1)} :=\ & \tau \in \{T, 2T, 3T, \ldots, PT\}, \\ Guard_{(q_1,q_2)} :=\ & \hat{x} \in \mathcal{B}^c_{\epsilon_x}(X_{i+1}) \wedge \hat{v} \in \mathcal{B}^c_{\epsilon_v}(0) \ \wedge \\ & \hat{a} \in \mathcal{B}^c_{\epsilon_a}(0) \wedge \\ & \tau \in \{T, 2T, 3T, \ldots, PT\}. \end{aligned}$$

In order to force transitions at certain points, the guards and invariants are defined in such a way that they complement each other.

The flow in state q_1 will depend on the tracking control law. As an example, let us consider the proportional derivative feedback which gives a constant acceleration between samples, as in,

$$\texttt{accel}_{(\hat{x},\hat{v},X)} := -K_p(\hat{x} - X) - K_v\hat{v}.$$

Then the flow in state q_1 is:

$$\begin{aligned} &Flow_{(q_1,(x,v,a,\hat{x}_0,\hat{x}_{-1},\hat{x}_{-2},\hat{a},\hat{v},\tau))}(t) := \\ &\quad (x + vt + \frac{1}{2}at^2, v + at, a, \hat{x}_0, \hat{x}_{-1}, \hat{x}_{-2},\ \hat{a}, \hat{v}, \tau + t). \end{aligned}$$

For $j = 1, 2$, the resets are given by:

$$\begin{aligned} &Reset_{(q_1,q_j)} := \\ &\quad x' = x \wedge\ v' = v \wedge a' = \mathcal{B}^c_{\chi_1}(\hat{a}') \wedge \\ &\quad \hat{x}'_0 \in \mathcal{B}^o_{\chi_2}(x)\ \wedge\ \hat{x}'_{-1} = \hat{x}_0\ \wedge\ \hat{x}'_{-2} = \hat{x}_{-1}\ \wedge \\ &\quad \hat{v}' = \frac{3}{4}(\hat{x}_0 - \hat{x}_{-1}) - \frac{1}{4}(\hat{x}_{-1} - \hat{x}_{-2})\ \wedge \\ &\quad \hat{a}' = \texttt{accel}_{(\hat{x}_0,\hat{v},X_{i+j})}\ \wedge \tau' = \tau. \end{aligned}$$

The primed variables correspond to the value of the variables after the discrete transition and the unprimed variables are the values of the variables before the discrete transition.

We want to verify that starting in X_1 we can track the points $X_1, \cdots, X_N$ in order in time T_{total}. As said before, we only verify that starting in a point close to X_i, we can reach a point close to X_{i+1} in time T_P. This can be expressed formally using the LTL formula

$$\diamond(x \in \mathcal{B}^o_{\epsilon_x}(X_{i+1}) \wedge v \in \mathcal{B}^o_{\epsilon_v}(0) \wedge a \in \mathcal{B}^o_{\epsilon_a}(0) \wedge \tau \leq T_P).$$

[5]We will be checking what is often called an *invariant condition*.

This formula should be true in all the initial states of the automaton for the desired property to hold.

C. Verification

The system described above satisfies the extended STORMED specifications: (see Section VI):

- **(S)** Guards are T-**S**eparable along τ because only the edges of the guards belonging to a strongly connected component need to be separable. See section VI.
- **(T)** The flows satisfy the **T**ime-independent/semi-group property (double integrators).
- **(O)** Everything is algebraically definable – therefore **O**-minimally definable
- **(RM)** **R**esets and flows are monotonic along τ since τ increases by at least T between every any two discrete transitions.
- **(ED)** The invariants and guards are bounded along the monotonic direction τ by 0 and T_P,

Note that essentially the points which satisfy $\tau \geq (P-n)T$ with a non-negative integer $n < P$ are in the nth neighborhood, i.e. executions from these points can only possibly have at most n more discrete transitions. As a result we can also say that these points cannot be, for example, in the $n+1$ neighborhood, by the definition of the system. Since the system above is STORMED we can use Lemma 4.2 and Theorem 4.3 to construct a finite bisimulation and then check the LTL formula on the bisimulation quotient. While constructing the bisimulation the initial partition should respect all the atomic formulas in the formula, namely, $x \in \mathcal{B}^o_{\epsilon_x}(X_{i+1})$, $v \in \mathcal{B}^o_{\epsilon_v}(0)$, $a \in \mathcal{B}^o_{\epsilon_a}(0)$, and $\tau \leq T_P$.

VIII. Conclusions and Outlook

A. Contribution

We proved the existence of a finite bisimulation for o-minimally definable hybrid systems with bounded discrete horizon and showed that STORMED hybrid systems satisfy these specifications. Furthermore, we have shown that the STORMED specifications and various extensions are apparent in modern day digital-control and real-time systems, by example. We have also shown that the STORMED constraints are tight, in that relaxation of even one constraint leads to undecidability. We have therefore presented a subclass of hybrid automata which:

1) has decidable properties,
2) is at the boundary of decidability,
3) and at the same time is rich enough to express properties of real-time systems.

B. On algorithms and applications

An alternating bisimulation refinement algorithm [9] will produce the bisimulation whose existence we proved in this paper. This bisimulation can be used to model check any formula in μ-calculus. After the bisimulation quotient is computed, software such as SPIN [24] can be used to model check a given property. However, in practice, for model-checking various simple properties such as safety, a bisimulation is not required, and as a matter of fact, it can be incredibly computationally costly. In our example, we produced a formula in the first order logic on the real closed field which can be verified by quantifier elimination. This example is for illustration purposes on the range of the problems that are decidable. For a numerically solved example, the reader is referred to [25].

Regarding the bounded discrete horizon verification, and referring to the example we have presented, it is evident that desirable properties of dynamical systems are based on contractions (Lyapunov functions), convergence rates and other invariant properties which can also be expressed on a finite horizon. Automated invariant generation has been presented in [26], [27] where quantifier elimination techniques and guided theorem provers are used. On a more theoretical basis, in [9] a monotonically better-approximating abstraction algorithm is identified. Finally, in [25] we have studied when approximation techniques are guaranteed to terminate. It would be interesting to further study the structure of invariant properties relevant to digital control systems and their bounded discrete horizon verification in terms of numerical algorithms and their robustness.

References

[1] G. Lafferriere, G. Pappas, and S. Sastry, "O-minimal hybrid systems," 1998.

[2] R. Gentilini, "Reachability problems on extended o-minimal hybrid automata," in *FORMATS*, ser. Lecture Notes in Computer Science, P. Pettersson and W. Yi, Eds., vol. 3829. Springer, 2005, pp. 162–176.

[3] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, no. 1, pp. 3–34, 1995.

[4] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" in *Proc. 27th Annual ACM Symp. on Theory of Computing (STOC)*, 1995, pp. 373–382.

[5] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[6] E. Asarin, O. Maler, and A. Pnueli, "Reachability analysis of dynamical systems having piecewise-constant derivatives," *Theoretical Computer Science*, vol. 138, no. 1, pp. 35–65, 1995.

[7] E.Asarin, G.Schneider, and S.Yovine, "Algorithmic analysis of polygonal hybrid systems, part I: Reachability," *Theor. Comput. Sci.*, vol. 379, no. 1-2, pp. 231–265, 2007.

[8] V. Vladimerou, P. Prabhakar, M. Viswanathan, and G. E. Dullerud, "STORMED hybrid systems," in *Proceedings of the ICALP*, Reykjavík, Iceland, 2008, pp. 136–147.

[9] R. Gentilini, K. Schneider, and B. Mishra, "Successive abstractions of hybrid automata for monotonic ctl model checking," in *LFCS '07: Proceedings of the international symposium on Logical Foundations of Computer Science*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 224–240.

[10] M. Jenkins, J. Ouaknine, A. Rabinovich, and J. Worrell, "Alternating timed automata over bounded time," in *LICS*. IEEE Computer Society, 2010, pp. 60–69.

[11] J. Ouaknine, A. Rabinovich, and J. Worrell, "Time-bounded verification," in *CONCUR*, ser. Lecture Notes in Computer Science, M. Bravetti and G. Zavattaro, Eds., vol. 5710. Springer, 2009, pp. 496–510.

[12] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. MIT Press, 2000.

[13] L. van den Dries, *Tame Topology and O-minimal Structures*. Cambridge Univesity Press, 1998.

[14] L. van den Dries and C. Miller, "On the real exponential field with restricted analytic functions," *Israel Journal of MAthematics*, no. 85, pp. 19–56, 1994.

[15] A. Tarski, *A Decision Method for Elementary Algebra and Geometry*, 2nd ed. University of California Press, 1951.

[16] R. Milner, *Communication and Concurrency*. Prentice-Hall, Inc, 1989.
[17] A. Biere, A. Cimatti, E. M. Clarke, O. Strichman, and Y. Zhu, "Bounded model checking," *Advances in Computers*, vol. 58, pp. 118–149, 2003.
[18] T. Brihaye and C. Michaux, "On the expressiveness and decidability of o-minimal hybrid systems," *J. Complexity*, vol. 21, no. 4, pp. 447–478, 2005.
[19] T. Brihaye, "Verification and control of o-minimal hybrid systems and weighted timed automata," Ph.D. dissertation, Academie Universitaire Wallonie-Bruxelles, 2006.
[20] C. Baier and J.-P. Katoen, *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.
[21] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed. United States of America: Addison-Wesley, 2001.
[22] P. Prabhakar, V. Vladimerou, M. Viswanathan, and G. E. Dullerud, "A decidable class of planar linear hybrid systems," in *Proceedings of the HSCC*, St. Louis, MO, 2008, pp. 401–414.
[23] ——, "A3 decidable class of planar linear hybrid systems," in *Proceedings of the HSCC*, St. Louis, MO, 2008, pp. 401–414.
[24] G. J. Holzmann, *The SPIN Model Checker : Primer and Reference Manual*. Addison-Wesley Professional, September 2003.
[25] P. Prabhakar, V. Vladimerou, M. Viswanathan, and G. E. Dullerud, "Verifying tolerant systems using polynomial approximations," in *IEEE Real-Time Systems Symposium*, 2009, pp. 181–190.
[26] A. Platzer and E. M. Clarke, "Computing differential invariants of hybrid systems as fixedpoints," in *ICAV*, 2008, pp. 176–189.
[27] A. Platzer and E. Clarke, "Computing differential invariants of hybrid systems as fixedpoints," Pittsburg, PA, Tech. Rep. CMU-CS-08-103, February 2008.

Vladimeros Vladimerou was born in Nicosia, Cyprus, in 1978. He received a BSc degree as a Fulbright Scholar, an MSc degree, as well as a PhD in Electrical and Computer Engineering in 2009, all from the University of Illinois in Urbana-Champaign (UIUC). During 2009-2010 he was a research fellow at the Automatic Control Department of the University of Lund, Sweden. From 2011 onwards, he has been with Toyota Research and Development at the Toyota Technical Center in Ann Arbor, Michigan.

His research interests include networked autonomous vehicles, formal methods and verification, distributed task allocaion, and vehicle-to-vehicle communications and control systems.

Pavithra Prabhakar obtained her Bachelor's degree from the National Insitute of Technology at Warangal in 2004 and a Masters degree from the Indian Institute of Science at Bangalore in 2006, both in computer science. She then obtained her PhD in computer science and a masters in applied mathematics from the University of Illinois at Urbana-Champaign in 2011 and 2010, respectively. She is currently a CMI postdoctoral fellow at California Institute of Technology with a joint appointment on the faculty of IMDEA software at Madrid, Spain. Her research interest is in the area of formal methods with applications to the analysis of hybrid systems.

Mahesh Viswanathan received his Bachelor of Technology degree in Computer Science and Engineering from the Indian Institute of Technology, Kanpur, India in 1995, and his PhD in Computer and Information Science from the University of Pennsylvania in 2000. He was a post-doctoral fellow at the Center for Discrete Mathematics (DIMACS), Rutgers University, with a joint appointment at Telcordia Technologies in the academic year 2000–01. Since 2001, he has been on the faculty at the Department of Computer Science at the University of Illinois, Urbana-Champaign where he is currently an Associate Professor. He was a visiting faculty fellow at the Courant Institute of Mathematics, New York University during the academic 2009–10, and a visiting faculty at the École Normale Supérieure at Cachan in the summer of 2011. His research interests are in the core areas of logic, automata theory, and algorithm design with its applications to automated verification of computing systems, including hybrid systems.

Geir Dullerud was born in Oslo, Norway, in 1966. He received the BASc degree in Engineering Science, in 1988, and the MASc degree in Electrical Engineering, in 1990, both from the University of Toronto, Canada. In 1994 he received his PhD in Engineering from the University of Cambridge, England.

Since 1998 he has been a faculty member in Mechanical Science and Engineering at the University of Illinois, Urbana-Champaign, where he is currently Professor; he is a member of the Coordinated Science Laboratory, where he is Director of the Decision and Control Laboratory. During the academic year 2005–2006 he held a visiting faculty position at Stanford University in Aeronautics and Astronautics. ¿From 1996–1998 he was an assistant professor in Applied Mathematics at the University of Waterloo, Canada; he was a Research Fellow and Lecturer at the California Institute of Technology, in the Control and Dynamical Systems Department, in 1994 and 1995.

He has published two books: *A Course in Robust Control Theory* (with F. Paganini), Springer 2000, and *Control of Uncertain Sampled-data Systems*, Birkhauser 1996. Currently he is associate editor of the IEEE Transactions on Automatic Control, and has previously served in this role for Automatica. His areas of current research interest include networked and cooperative control, robotic vehicles, complex and hybrid dynamical systems. He received the National Science Foundation CAREER Award in 1999, and in 2005 the Xerox Faculty Research Award at UIUC. He became a Fellow of the IEEE in 2008 and ASME Fellow in 2011.