Hybrid Acceleration using Real Vector Automata

Bernard Boigelot, Frédéric Herbreteau, Sébastien Jodogne

Hybrid Acceleration using Real Vector Automata* (extended abstract)

Bernard Boigelot, Frédéric Herbreteau, and Sébastien Jodogne**

Université de Liège,
Institut Montefiore, B28,
B-4000 Liège, Belgium
{boigelot,herbreteau,jodogne}@montefiore.ulg.ac.be,
http://www.montefiore.ulg.ac.be/~{boigelot,herbreteau,jodogne}/

Abstract. This paper addresses the problem of computing an exact and effective representation of the set of reachable configurations of a linear hybrid automaton. Our solution is based on accelerating the state-space exploration by computing symbolically the repeated effect of control cycles. The computed sets of configurations are represented by *Real Vector Automata (RVA)*, the expressive power of which is beyond that of the first-order additive theory of reals and integers. This approach makes it possible to compute in finite time sets of configurations that cannot be expressed as finite unions of convex sets. The main technical contributions of the paper consist in a powerful sufficient criterion for checking whether a hybrid transformation (i.e., with both discrete and continuous features) can be accelerated, as well as an algorithm for applying such an accelerated transformation on RVA. Our results have been implemented and successfully applied to several case studies, including the well-known leaking gas burner, and a simple communication protocol with timers.

1 Introduction

The reachability problem, which consists in checking whether a given set of system configurations can be reached from the initial state, is central to verifying safety properties of computerized systems. A more general form of this problem is to compute the set of all reachable configurations of a given system. Since the reachability set is in general infinite, the result of the computation has to be expressed symbolically rather than explicitly. The goal is to obtain a symbolic representation of this set that is both *exact*, i.e., containing sufficient information for checking without approximation whether a given configuration is reachable

* This work was partially funded by a grant of the "Communauté française de Belgique - Direction de la recherche scientifique - Actions de recherche concertées", and by the European IST-FET project ADVANCE (IST-1999-29082).

** Research Fellow ("aspirant") for the Belgian National Fund for Scientific Research (FNRS).

or not, and *effective*, meaning that the safety properties of interest should be easily and efficiently decidable on the basis of the set representation.

This paper addresses the problem of computing the exact reachability set of hybrid automata, which are finite-state machines extended with real variables that obey continuous and discrete dynamical laws [AHH93,ACH+95,Hen96]. Hybrid automata have been introduced as natural models for dynamical systems with both discrete and continuous features, such as embedded control programs and timed communication protocols.

A simple way of computing the reachability set of an automaton extended with variables is to explore its state space, starting from the initial configurations, and propagating the reachability information along the transition relation. In the case of hybrid automata, this approach is faced with two difficulties. First, the dynamical laws governing the evolution of real variables are such that a configuration has generally an infinite number of direct successors, due to the continuous nature of time. This problem can be overcome by grouping into *regions* configurations that can be handled together symbolically during the state-space exploration. The second problem is that the set of such reachable regions can generally not be obtained after a finite number of exploration steps.

For restricted classes of hybrid automata such as *Timed Automata* [AD94], it has been shown that exploring a finite region graph is sufficient for deciding the reachability problem. More general classes, such as *Initialized Rectangular Automata* [HKPV98] and *O-Minimal Hybrid Automata* [PLY99] can also be handled by reducing their analysis to the reachability problem for timed automata. The main drawback of this approach is that the size of the region graph grows exponentially with respect to the magnitude of constants that appear in the model. Besides, analyzing more general models such as *Linear Hybrid Automata* leads to region graphs that are inherently not reducible to finite ones.

However, techniques have been developed for exploring infinite structures using a finite amount of resources. In particular, *meta-transitions* [BW94,Boi99] are objects that can be added to the transition relation of an extended automaton in order to speed up its state-space exploration. A meta-transition is usually associated to a cycle in the automaton control graph. During state-space exploration, following a meta-transition leads in one step to all the configurations that could be reached by following the corresponding cycle any possible number of times. Symbolic state-space exploration using meta-transitions thus makes it possible to go arbitrarily deep in the exploration tree by executing a finite number of operations, and can be seen as an *acceleration* method for step-by-step exploration. This method requires a symbolic representation system for the sets of configurations produced by meta-transitions, as well as a decision procedure for checking whether a given cycle can be turned into a meta-transition (in other words, whether one can compute its unbounded repeated effect, as well as effectively perform this computation with represented sets). Representation systems, decision procedures and computation algorithms have been developed for several classes of infinite-state systems including automata extended with

unbounded integer variables [WB95,Boi99] and state machines communicating via unbounded FIFO channels [BG96,BH97,Boi99].

We argue that applying acceleration methods to hybrid automata is essential to being able to analyze exactly systems that cannot be reduced to small finite models. Meta-transitions cannot be straightforwardly applied to hybrid automata, due to the both discrete and continuous nature of these: the data transformation associated to a control cycle of a hybrid automaton is generally non functional (the image of a single configuration by such a transformation may contain more than one configuration), for the time elapsed in each visited location may differ from one run to another. Moreover, one needs a symbolic representation for sets with discrete and continuous features.

In existing work, these drawbacks are avoided in the following ways. In [HPR94,AHH93], *widening* operators are introduced in order to force the result of every acceleration step to be representable as a finite union of convex polyhedra. This approach guarantees the termination of state-space exploration, but is generally only able to produce an upper approximation of the reachability set. In [HL02], an exact acceleration technique is developed for timed automata, using *Difference Bounds Matrices (DBM)*, i.e., restricted convex polyhedra, as a symbolic representation system. This method speeds up the exploration of finite region graphs but, because DBM are generally not expressive enough to represent infinite unions of convex sets, it cannot be directly applied to more general hybrid models. In [CJ99], the authors introduce a cycle acceleration technique for flat automata extended with integer, rational, or real variables. This approach can be applied to the analysis of timed automata, but the considered model is not expressive enough to handle more general hybrid systems. In [AAB00], the analysis of timed automata extended with integer variables and parameters is carried out using *Parametric Difference Bounds Matrices (PDBM)* as representations, as well as an extrapolation technique in order to ensure termination. However, the expressiveness of PDBM is still not sufficient for verifying hybrid systems. In [BBR97], a more expressive representation system, the *Real Vector Automaton (RVA)* is introduced in order to represent sets of vectors with real and integer components. The acceleration method proposed in [BBR97] can only associate meta-transitions to cycles that correspond to functional transformations. This strong restriction prevents this solution from being applied to systems whose timed features make their transition graph inherently non deterministic.

In this paper, we build upon these prior results and develop an acceleration method, based on cyclic meta-transitions, that takes into account both discrete and continuous features of the accelerated transformations, hence the name *hybrid acceleration*. We focus on an exact computation of the reachability set, therefore, owing to the undecidable nature of this problem, our solution takes the form of a partial algorithm, i.e., one that may not terminate. This algorithm relies on Real Vector Automata for representing the computed sets of configurations, which are expressed in the first-order additive theory of integers and reals [BRW98,BJW01]. The technical contributions of the paper consist of

a powerful sufficient criterion for checking whether the repeated iteration of a given control cycle of a linear hybrid automaton produces a set that stays within that theory, as well as an associated algorithm for computing on RVA the effect of such accelerations. Our method has the main advantage of being applicable to systems for which other direct approaches fail. This claim is substantiated by a prototype implementation in the framework of the tool LASH [LASH], that has been successfully applied to several case studies. Those include a direct reachability analysis of the well-known leaking gas burner model [CHR91] and of a simple parametrized communication protocol with timers.

2 Linear Hybrid Automata

We use the term *convex linear constraint* to denote a finite conjunction of closed linear constraints with integer coefficients, i.e., a set $\{\boldsymbol{x} \in \mathbb{R}^n \mid P\boldsymbol{x} \leq \boldsymbol{q}\}$, with $P \in \mathbb{Z}^{m \times n}$ and $\boldsymbol{q} \in \mathbb{Z}^m$. The term *linear transformation* denotes a relation of the form $\{(\boldsymbol{x}, \boldsymbol{x}') \in \mathbb{R}^n \times \mathbb{R}^n \mid \boldsymbol{x}' = A\boldsymbol{x} + \boldsymbol{b}\}$, with $A \in \mathbb{Z}^{n \times n}$ and $\boldsymbol{b} \in \mathbb{Z}^n$.

Definition 1. *A* Linear Hybrid Automaton (LHA) *[AHH93,ACH+95,Hen96] is a tuple* $(\boldsymbol{x}, V, E, v_0, X_0, G, A, I, R)$*, where*

- $\boldsymbol{x}$ *is a vector of* n *real-valued* variables*, with* $n > 0$*;*
- (V, E) *is a finite directed* control graph*, the vertices of which are the* locations *of the automaton. The* initial location *is* v_0*;*
- X_0 *is an* initial region*, defined by a convex linear constraint;*
- G *and* A *respectively associate to each edge in* E *a* guard*, which is a convex linear constraint, and an* assignment*, which is a linear transformation;*
- I *and* R *respectively associate to each location in* V *an* invariant*, which is a convex linear constraint, and a* rectangular activity $(\boldsymbol{l}, \boldsymbol{u}) \in \mathbb{Z}^n \times \mathbb{Z}^n$*, which denotes the constraint* $\boldsymbol{l} \leq \dot{\boldsymbol{x}} \leq \boldsymbol{u}$*, where* $\dot{\boldsymbol{x}}$ *is the first derivative of* $\boldsymbol{x}$*.*

The semantics of a LHA $(\boldsymbol{x}, V, E, v_0, X_0, G, A, I, R)$ is defined by the transition system $(Q, Q_0, (\rightarrow_\delta \cup \rightarrow_\tau))$, where

- $Q = V \times \mathbb{R}^n$ is the set of *configurations*;
- $Q_0 = \{(v, \boldsymbol{x}) \in Q \mid v = v_0 \wedge \boldsymbol{x} \in X_0 \cap I(v_0)\}$ is the set of *initial configurations*;
- The *discrete-step* transition relation $\rightarrow_\delta \subseteq Q \times Q$ is such that $(v, \boldsymbol{x}) \rightarrow_\delta (v', \boldsymbol{x}')$ iff there exists $e \in E$ such that $e = (v, v')$, $\boldsymbol{x} \in G(e)$ and $(\boldsymbol{x}, \boldsymbol{x}') \in A(e)$, and $\boldsymbol{x}' \in I(v')$. Such a transition can also be denoted $(v, \boldsymbol{x}) \stackrel{e}{\rightarrow}_\delta (v', \boldsymbol{x}')$ when one needs to refer explicitly to e;
- The *time-step* transition relation $\rightarrow_\tau \subseteq Q \times Q$ is such that $(v, \boldsymbol{x}) \rightarrow_\tau (v', \boldsymbol{x}')$ iff $v' = v$, there exists $t \in \mathbb{R}_{\geq 0}$ such that $\boldsymbol{x} + t\boldsymbol{l} \leq \boldsymbol{x}' \leq \boldsymbol{x} + t\boldsymbol{u}$, with $(\boldsymbol{l}, \boldsymbol{u}) = R(v)$, and $\boldsymbol{x}' \in I(v)$.

Let $\rightarrow$ denote the relation $(\rightarrow_\delta \cup \rightarrow_\tau)$, and let $\rightarrow^*$ be the reflexive and transitive closure of $\rightarrow$. A configuration $(v', \boldsymbol{x}') \in Q$ is *reachable from* a configuration $(v, \boldsymbol{x}) \in Q$ iff $(v, \boldsymbol{x}) \rightarrow^* (v', \boldsymbol{x}')$. A configuration is *reachable* iff it is reachable from some configuration in Q_0. The *reachability set* $Post^*(H)$ of a LHA H is the set of its reachable configurations.

3 Symbolic State-Space Exploration

3.1 Introduction

We address the problem of computing the reachability set of a given LHA $H = (\boldsymbol{x}, V, E, v_0, X_0, G, A, I, R)$. This set can generally not be enumerated, since a configuration in the semantic transition graph $(Q, Q_0, (\rightarrow_\delta \cup \rightarrow_\tau))$ of H may have uncountably many direct time-step successors, due to the dense nature of time.

One thus uses *symbolic* methods, which basically consist in handling *regions* (v, S), with $v \in V$ and $S \subseteq \mathbb{R}^n$, instead of single configurations. The transition relations $\rightarrow_\delta$ and $\rightarrow_\tau$ can straightforwardly be extended to regions:

- $(v, S) \rightarrow_\delta (v', S')$ iff there exists $e \in E$ such that $e = (v, v')$, and $S' = \{\boldsymbol{x'} \in I(v') \mid (\exists \boldsymbol{x} \in S)(\boldsymbol{x} \in G(e) \wedge (\boldsymbol{x}, \boldsymbol{x'}) \in A(e))\}$. Such a transition can also be denoted $(v, S) \xrightarrow{e}_\delta (v', S')$;
- $(v, S) \rightarrow_\tau (v', S')$ iff $v' = v$, and $S' = \{\boldsymbol{x'} \in I(v) \mid (\exists \boldsymbol{x} \in S)(\exists t \in \mathbb{R}_{\geq 0})(\boldsymbol{x} + t\boldsymbol{l} \leq \boldsymbol{x'} \leq \boldsymbol{x} + t\boldsymbol{u})\}$, with $(\boldsymbol{l}, \boldsymbol{u}) = R(v)$.

The symbolic exploration of the state space of H starts from the initial region Q_0, and computes the reachable regions by adding repeatedly to the current set the regions that can be reached by following the relations $\rightarrow_\delta$ and $\rightarrow_\tau$. The computation ends when a fixpoint is reached.

Termination of state-space exploration is clearly not guaranteed, due to the undecidability of the reachability problem for LHA [ACH+95,Hen96]. However, the simple algorithm outlined above can substantially be improved by applying *acceleration*, the purpose of which is to explore an infinite number of reachable regions in finite time. Acceleration methods are introduced in Section 3.3, and are then applied to LHA in Section 4.

Now, in order to be able to carry out algorithmically symbolic state-space exploration, one needs a *symbolic representation* for the regions that must be manipulated. This representation has to satisfy some requirements. First, it must be closed under the set operations that need to be performed during exploration. These include the classical set-theory operations $\cup$, $\subseteq$, $\times$, ..., as well as computing the successors of regions by the discrete-step and time-step relations. Second, one should be able to carry out the acceleration operations with represented sets. Finally, the system properties that are to be checked must be easily decidable from the representation of the computed reachability set.

The traditional symbolic representations used in the framework of hybrid automata analysis are the finite unions of convex polyhedra [HPR94,HH94], and the *Difference Bounds Matrices (DBM)* [Dil89]. These representations are not able to handle sets that cannot be expressed as finite unions of convex polyhedra, such as the reachability set of the leaking gas burner [CHR91]. In Section 3.2, we recall powerful representations for sets of real vectors, the *Real Vector Automata*, that have the advantage of being much more expressive than geometrical representations, and have good properties that allow an efficient manipulation of represented sets.

3.2 Real Vector Automata

Consider an integer base $r > 1$. Using the positional number system in base r, a positive real number x can be written as an infinite word over the alphabet $\Sigma = \{0, 1, \ldots, r-1, \star\}$, where "$\star$" denotes a separator between the integer and the fractional parts. For example, $5/2$ can be written as $10 \star 1(0)^\omega$, where "$^\omega$" denotes infinite repetition. In a similar way, negative numbers and real vectors with a fixed dimension n can also be encoded in base r as infinite words over the finite alphabet Σ [BBR97,BRW98,BJW01].

Given a set $S \subseteq \mathbb{R}^n$, let $L(S) \subseteq \Sigma^\omega$ denote the language of all the encodings of all the vectors in S. If $L(S)$ is ω-rational, then it is accepted by a Büchi automaton A, which is said to be a *Real Vector Automaton (RVA)* that *represents* the set S.

It is known that the sets of real vectors that can be represented by RVA are those that are definable in a base-dependent extension of the first-order theory $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ [BRW98]. RVA are thus expressive enough to handle linear constraints over real and integer variables, as well as periodicities, and are closed under Boolean operators and first-order quantifiers. Moreover, it has been shown that staying within $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ makes it possible to avoid the use of the complex and costly algorithms that are usually required for handling infinite-words automata [BJW01]. Moreover, RVA admit a normal form [Löd01], which speeds up set comparisons and prevents the representations from becoming unnecessarily large. An implementation of RVA restricted to $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ is available in the framework of the LASH toolset [LASH].

3.3 Acceleration Methods

The idea behind acceleration is to capture the effect of selected *cycles* in the control graph (V, E) of the LHA H being analyzed. Let $\sigma = e_1; e_2; \ldots; e_p$ be such a cycle, where for each $i \in [1, \ldots, p]$, v_i is the origin of e_i. The transition e_i has the destination v_{i+1} if $i < p$, and v_1 if $i = p$. The transformation $\theta : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$ associated to σ is such that $\theta(S) = S'$ iff there exist $S_1, S'_1, S_2, S'_2, \ldots, S'_{p-1}, S_p \subseteq \mathbb{R}^n$ such that $S_1 = S$ and $S_p = S'$, and for each $i \in [1, \ldots, p-1]$, $(v_i, S_i) \to_\tau (v_i, S'_i)$ and $(v_i, S'_i) \xrightarrow{e_i}_\delta (v_{i+1}, S_{i+1})$.

The *meta-transition* [WB95,BBR97,Boi99] corresponding to σ is defined as a relation $\to_\sigma$ over the regions of H such that $(v, S) \to_\sigma (v', S')$ iff $v = v' = v_1$ and $S' = \theta^*(S)$, where $\theta^* = \cup_{i \geq 0} \theta^i$. Clearly, meta-transitions preserve reachability, in the sense that (v', S') is reachable if (v, S) is reachable. They can thus be added to the semantic transition system of an LHA in order to speed up its state-space exploration. A meta-transition is able to produce in one step regions that could only be obtained after an unbounded number of iterations of the corresponding cycle. Meta-transitions thus provide an acceleration strategy that makes it possible to explore infinite region graphs (though not all of them) in finite time. Meta-transitions can either be selected manually, or discovered by automatic or semi-automatic methods [Boi99].

The use of meta-transitions requires a decision procedure for checking whether the closure θ^* of a given transformation θ can effectively be constructed, and whether the image by this closure of a set of configurations can be computed within the symbolic representation used during the analysis. Those problems are tackled in Section 4, in the case of LHA analyzed using Real Vector Automata.

4 Hybrid Acceleration

4.1 Linear Hybrid Transformations

We establish now the general form of the data transformations on which acceleration can be applied. We first consider the case of a path σ performing a time step at some location v followed by a discrete step along an edge e from v to v'. The data transformation θ associated to σ is such that

$$\theta(S) = \{\boldsymbol{x}' \in \mathbb{R}^n \mid (\exists \boldsymbol{x} \in S)(\exists \boldsymbol{x}'' \in \mathbb{R}^n)(\exists t \in \mathbb{R}_{\geq 0})(\boldsymbol{x} + t\boldsymbol{l} \leq \boldsymbol{x}'' \leq \boldsymbol{x} + t\boldsymbol{u} \\ \wedge\, \boldsymbol{x}'' \in I(v) \wedge \boldsymbol{x}'' \in G(e) \wedge (\boldsymbol{x}'', \boldsymbol{x}') \in A(e) \wedge \boldsymbol{x}' \in I(v'))\},$$

where $(\boldsymbol{l}, \boldsymbol{u}) \in R(v)$.

By Definition 1, the constraints $I(v)$, $G(e)$, $A(e)$, and $I(v')$ are systems of linear equalities and inequalities. The previous formula can thus be rewritten as $\theta(S) = \{\boldsymbol{x}' \in \mathbb{R}^n \mid (\exists \boldsymbol{x} \in S)(\exists \boldsymbol{x}'' \in \mathbb{R}^n)(\exists t \in \mathbb{R}_{\geq 0})\varphi(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}'', t)\}$, where φ is a conjunction of linear inequalities, in other words a closed convex polyhedron in $\mathbb{R}^{3n+1}$. Such polyhedra are closed under projection. One can therefore project out the quantified variables $\boldsymbol{x}''$ and t, which yields $\theta(S) = \{\boldsymbol{x}' \in \mathbb{R}^n \mid (\exists \boldsymbol{x} \in S)\varphi'(\boldsymbol{x}, \boldsymbol{x'})\}$, where φ' is a conjunction of linear inequalities. This prompts the following definition.

Definition 2. *A* Linear Hybrid Transformation (LHT) *is a transformation of the form*

$$\theta : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n} : S \mapsto \left\{ \boldsymbol{x}' \in \mathbb{R}^n \,\middle|\, (\exists \boldsymbol{x} \in S) \left(P \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}' \end{bmatrix} \leq \boldsymbol{q} \right) \right\},$$

with $n > 0, P \in \mathbb{Z}^{m \times 2n}$, $\boldsymbol{q} \in \mathbb{Z}^m$ *and* $m \geq 0$.

Notice that a LHT is entirely defined by the linear system induced by P and $\boldsymbol{q}$. In the sequel, for simplicity sake, we denote such a LHT by the pair $(P, \boldsymbol{q})$.

We have just established that data transformations associated to a time step followed by a discrete step can be expressed as LHT. Thanks to the following result, the transformations corresponding to arbitrary control paths of LHA can be described by LHT as well.

Theorem 1. *Linear Hybrid Transformations are closed under composition.*

Proof Sketch. Immediate by quantifying away the intermediate variables. □

Note that the image by a LHT $(P, \boldsymbol{q})$ of a single vector $\boldsymbol{x}_0 \in \mathbb{R}^n$ is the set $\{\boldsymbol{x} \in \mathbb{R}^n \mid P''\boldsymbol{x} \leq \boldsymbol{q} - P'\boldsymbol{x}_0\}$, where $[P'; P''] = P$ with $P', P'' \in \mathbb{Z}^{m \times n}$, which defines a closed convex polyhedron. Each constraint of this polyhedron has the form $\boldsymbol{p''}.\boldsymbol{x} \leq q - \boldsymbol{p'}.\boldsymbol{x}_0$, hence has coefficients that are independent from the initial vector $\boldsymbol{x}_0$, and an additive term that depends linearly on $\boldsymbol{x}_0$.

4.2 Iterating Transformations

We now address the problem of computing within $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ the closure of a transformation θ defined by a Linear Hybrid Transformation $(P, \boldsymbol{q})$.

Clearly, there exist transformations θ and sets $S \subseteq \mathbb{R}^n$ definable in $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ such that $\theta^*(S)$ does not belong to that theory[1]. Determining exactly whether $\theta^*(S)$ is definable in $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ for all definable sets S is a hard problem. Indeed, a solution to this problem would also cover the simpler case of guarded discrete linear transformations, for which no decision procedure is known [Boi99]. A realistic goal is thus to obtain a *sufficient* criterion on θ, provided that it is general enough to handle non trivial hybrid accelerations, and that the closure of every transformation that satisfies the criterion can be effectively computed.

A first natural restriction consists of requiring that the k-th image by θ, denoted $\theta^k(S)$, of a set S definable in $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ is definable in that theory in terms of elements $\boldsymbol{x}_0$ of S and k. The set $\theta^*(S)$ could then be computed by existentially quantifying $\theta^k(S)$ over $k \geq 0$ and $\boldsymbol{x}_0 \in S$.

Consider an arbitrary $\boldsymbol{x}_0 \in S$ and a fixed value $k > 0$. Since by Theorem 1, the transformation θ^k can be expressed as a LHT, the set $\theta^k(\{\boldsymbol{x}_0\})$ is a closed convex polyhedron Π_k. Such a polyhedron is uniquely characterized by its set of *vertices* and *extremal rays* [Wey50]. We now study the evolution of the vertices and rays of Π_k, for all $k > 0$, as functions of $\boldsymbol{x_0}$ and k.

We have $\Pi_1 = \{\boldsymbol{x} \in \mathbb{R}^n \mid P''\boldsymbol{x} \leq \boldsymbol{q} - P'\boldsymbol{x_0}\}$, with $[P'; P''] = P$. Let $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_p$ be the vertices and rays of Π_1. Each $\boldsymbol{r}_i$ is uniquely characterized as the intersection of a particular subset of constraint boundaries. In other words, $\boldsymbol{r}_i$ is the only solution to an equation of the form $P_i\boldsymbol{r}_i = \boldsymbol{q}_i(\boldsymbol{x}_0)$, where P_i does not depend on $\boldsymbol{x}_0$, and $\boldsymbol{q}_i(\boldsymbol{x}_0)$ depends linearly on $\boldsymbol{x}_0$. From this property, we define the *trajectory* of $\boldsymbol{r}_i$ with respect to θ as the infinite sequence $\boldsymbol{r}_i^0, \boldsymbol{r}_i^1, \boldsymbol{r}_i^2, \ldots$ where $\boldsymbol{r}_i^0 = \boldsymbol{r}_i$, and for each $j > 0$, $P_i\boldsymbol{r}_i^j = \boldsymbol{q}_i(\boldsymbol{r}_i^{j-1})$.

It is known [Wei99] that the sets of discrete values that are definable in $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ are *ultimately periodic*, i.e., they can be expressed as a finite union of sets of the form $\{a + jb \mid j \in \mathbb{N}\}$. Thus, a natural restriction on θ is to require that each vertex or ray of the image polyhedron θ follows a periodic trajectory. Unfortunately, such a criterion would be rather costly to check explicitly, for the number of vertices and rays of a n-dimensional polyhedron may grow exponentially in n. However, since each vertex or ray is an intersection of constraint boundaries which take the form of $(n-1)$-planes, its trajectory is always periodic whenever these hyperplanes follow periodic trajectories themselves. It is therefore sufficient to impose the periodicity restriction on the *linear constraints* that characterize θ rather than on the vertices and rays of its image polyhedron. We are now ready to formally characterize the Linear Hybrid Transformations on which acceleration can be applied.

Definition 3. *Let $\theta = (P, \boldsymbol{q})$, and let C be a constraint of θ, i.e., a row $\boldsymbol{p''}.\boldsymbol{x'} \leq q - \boldsymbol{p'}.\boldsymbol{x}$ of the underlying linear system of θ. Let θ_C be the LHT induced by the boundary of C, i.e., transforming $\boldsymbol{x}$ into $\boldsymbol{x'}$ such that $\boldsymbol{p''}.\boldsymbol{x'} = q - \boldsymbol{p'}.\boldsymbol{x}$.*

[1] Consider for instance the transformation $x \mapsto 2x$ with $S = \{1\}$.

For each $d \in \mathbb{R}$, let $\Gamma_{(C,d)}$ denote the set $\{\boldsymbol{x}' \in \mathbb{R}^n \mid \boldsymbol{p}''.\boldsymbol{x}' = d\}$ (that is, the general form of the image by θ_C of a single vector).

The transformation θ_C is periodic *if $\boldsymbol{p}'' = \boldsymbol{0}$ (in which case the image of every vector is the empty set or the universal set), or if the following conditions are both satisfied:*

- *For every $d \in \mathbb{R}$, there exists a single $d' \in \mathbb{R}$ such that $\theta_C(\Gamma_{(C,d)}) = \Gamma_{(C,d')}$ (this condition ensures that the image of the $(n-1)$-plane $\Gamma_{(C,d)}$ is a $(n-1)$-plane parallel to it);*
- *Let $d_0, d_1, \ldots \in \mathbb{R}$ be such that for every $j > 0$, $\theta_C(\Gamma_{(C,d_j)}) = \Gamma_{(C,d_{j-1})}$. The sequence $d_0, d_1, \ldots$ is such that for every $j > 0$, $d_j - d_{j-1} = d_{j+1} - d_j$ (in other words, the sequence must be an arithmetic progression).*

Definition 4. *A LHT is* periodic *if the transformations induced by all its underlying constraints are periodic.*

The periodicity of a LHT can be checked easily thanks to the following result.

Theorem 2. *Let θ_C be a LHT transforming $\boldsymbol{x}$ into $\boldsymbol{x}'$ such that $\boldsymbol{p}''.\boldsymbol{x}' = q - \boldsymbol{p}'.\boldsymbol{x}$. This transformation is periodic iff $\boldsymbol{p}'' = \boldsymbol{0}$ or $\boldsymbol{p}' = \lambda \boldsymbol{p}''$, with $\lambda \in \{0, -1\}$.*

Proof Sketch. If $\boldsymbol{p}'' = \boldsymbol{0}$, the result is immediate. If $\boldsymbol{p}'' \neq \boldsymbol{0}$, then the image by θ_C of a set $\Gamma_{(C,d_i)}$ is a $(n-1)$-plane if and only if $\boldsymbol{p}'$ and $\boldsymbol{p}''$ are colinear (otherwise, the image is $\mathbb{R}^n$). Let $\lambda \in \mathbb{R}$ be such that $\boldsymbol{p}' = \lambda \boldsymbol{p}''$. The periodicity condition on a sequence $d_0, d_1, \ldots$ constructed from an arbitrary $d_0 \in \mathbb{R}$ can only be fulfilled if $\lambda \in \{0, -1\}$. The details of the last step are omitted from this proof sketch. □

The previous theorem leads to a simple characterization of periodic transformations.

Theorem 3. *A LHT $(P, \boldsymbol{q})$ is periodic if and only if its underlying linear system is only composed of constraints of the form $\boldsymbol{p}.\boldsymbol{x} \leq q$, $\boldsymbol{p}.\boldsymbol{x}' \leq q$, and $\boldsymbol{p}.(\boldsymbol{x}' - \boldsymbol{x}) \leq q$.*

4.3 Image Computation

In this section, we address the problem of computing effectively $\theta^*(S)$, given a periodic LHT $\theta = (P, \boldsymbol{q})$ and a set S represented by a RVA.

Since θ is periodic, its underlying linear system can be decomposed into $P_0\boldsymbol{x} \leq \boldsymbol{q}_0 \wedge P_1(\boldsymbol{x}' - \boldsymbol{x}) \leq \boldsymbol{q}_1 \wedge P_2\boldsymbol{x}' \leq \boldsymbol{q}_2$, with $P_0 \in \mathbb{Z}^{m_0 \times n}$, $P_1 \in \mathbb{Z}^{m_1 \times n}$, $P_2 \in \mathbb{Z}^{m_2 \times n}$, $\boldsymbol{q}_0 \in \mathbb{Z}^{m_0}$, $\boldsymbol{q}_1 \in \mathbb{Z}^{m_1}$, $\boldsymbol{q}_2 \in \mathbb{Z}^{m_2}$, and $m_0, m_1, m_2 \in \mathbb{N}$. We first study transformations for which $m_0 = m_2 = 0$.

Theorem 4. *Let θ_1 be the LHT characterized by the linear system $P_1(\boldsymbol{x}' - \boldsymbol{x}) \leq \boldsymbol{q}_1$. The LHT induced by the system $P_1(\boldsymbol{x}' - \boldsymbol{x}) \leq k\boldsymbol{q}_1$, with $k > 0$, is equal to the k-th power θ_1^k of θ_1.*

Proof Sketch. The proof is based on a convexity argument, showing that the trajectory of a vector by the transformation θ_1 can always be constrained to follow a straight line. The details are omitted from this extended abstract due to space requirements. □

Let now consider a periodic LHT θ for which m_0 and m_2 are not necessarily equal to zero. Let $P_0\boldsymbol{x} \leq \boldsymbol{q}_0 \wedge P_1(\boldsymbol{x}' - \boldsymbol{x}) \leq \boldsymbol{q}_1 \wedge P_2\boldsymbol{x}' \leq \boldsymbol{q}_2$ be its underlying linear system, θ_1 be the LHT induced by $P_1(\boldsymbol{x}' - \boldsymbol{x}) \leq \boldsymbol{q}_1$, and let $C_\theta = \{\boldsymbol{x} \in \mathbb{R}^n \mid P_0\boldsymbol{x} \leq \boldsymbol{q}_0 \wedge P_2\boldsymbol{x} \leq \boldsymbol{q}_2\}$.

Theorem 5. *Periodic θ are such that for every $S \subseteq \mathbb{R}^n$ and $k \geq 2$, $\theta^k(S) = \theta(\theta_1^{k-2}(\theta(S) \cap C_\theta) \cap C_\theta)$.*

Proof Sketch. The proof is based on a convexity argument. □

Theorems 4 and 5 give a definition within $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ of $\theta^k(S)$ as a function of k and S, for all periodic LHT θ. The expression of $\theta^k(S)$ can be turned into an algorithm for applying θ^k to sets represented by RVA. By quantifying k over $\mathbb{N}$, this algorithm can be used for computing $\theta^*(S)$ given θ and a representation of S.

4.4 Reduction Step

The applicability of the acceleration method developed in Sections 4.2 and 4.3 is limited by the following observation: there exist LHT θ that do not satisfy the hypotheses of Theorem 3, but such that the sequence of sets $\theta(\boldsymbol{x}), \theta^2(\boldsymbol{x}), \theta^3(\boldsymbol{x}), \ldots$ induced by any vector $\boldsymbol{x}$ has a periodic structure. This is mainly due to the fact that the *range* of such θ, which is the smallest set covering the image by θ of every vector in $\mathbb{R}^n$, may have a dimension less than n. In such a case, since the behavior of θ outside of its range has no influence on its closure, it is sufficient to study the periodicity of θ in a subspace smaller than $\mathbb{R}^n$.

We solve this problem by reducing LHT in the following way, before computing their closure. The range $\theta(\mathbb{R}^n)$ of θ is the projection onto $\boldsymbol{x}'$ of its underlying linear system, expressed over the variable vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$. It hence takes the form of a closed convex polyhedron Π. The largest vector subspace that includes Π can be obtained by first translating Π by some vector $\boldsymbol{v}$ in order to make it cover the origin vector $\mathbf{0}$, and then extracting from the resulting polyhedron a finite vector basis $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{n'}$, i.e., a maximal set of linearly independent vectors (this can be done automatically). Then, if $n' < n$, we change variables from $\boldsymbol{x} \in \mathbb{R}^n$ to $\boldsymbol{y} \in \mathbb{R}^{n'}$ such that $\boldsymbol{x} = U\boldsymbol{y} + \boldsymbol{v}$, where $U = [\boldsymbol{u}_1; \boldsymbol{u}_2; \ldots; \boldsymbol{u}_{n'}]$. This operation transforms $\theta = (P, \boldsymbol{q})$ into the LHT

$$\theta' = \left(P \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix},\ \boldsymbol{q} - P \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{v} \end{bmatrix} \right),$$

the periodic nature of which can then be checked.

Theorem 6. *Let $\boldsymbol{x} \mapsto \boldsymbol{y} : \boldsymbol{x} = U\boldsymbol{y} + \boldsymbol{v}$ be a variable change operation transforming θ into θ'. For every $S \subseteq \mathbb{R}^n$, we have $\theta^*(S) = S \cup (U((\theta')^*(S')) + \boldsymbol{v})$, where $\theta(S) = US' + \boldsymbol{v}$.*

Proof Sketch. Immediate by simple algebra. □

It is worth mentioning that the transformation expressed by the previous theorem can be carried out within $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$.

5 Experiments

The acceleration technique developed in this paper has been implemented in the LASH toolset [LASH] and applied to two simple case studies.

The first experiment consisted in analyzing the *leaking gas burner* described in [CHR91,ACH+95]. We computed the reachability set of this model, after creating a meta-transition corresponding to the only simple cycle, and reducing its associated transformation to a subspace of dimension 2 (by applying the automatic procedure outlined in Section 4.4). The result of our analysis takes the form of a RVA that can be used for deciding quickly any safety property expressed in $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$.

Our second case study tackled the analysis of a generalization of the Alternating Bit Protocol [BSW69] to message ranges between 0 and $N - 1$, where $N \geq 2$ is an unbounded parameter. Moreover, the sender and receiver processes use timeouts to guess message losses, then reemitting until the expected acknowledgement is received.

The verification of this protocol has already been successfully achieved in previous work, however using techniques that required to abstract (at least) its temporal specifications. Using our acceleration method, we were able to compute exactly and in a direct way the reachability set of the full protocol, as well as to verify that the message sequence could not be broken.

6 Conclusions

This paper introduces a new acceleration method for computing the reachability set of Linear Hybrid Automata. Hybrid acceleration takes advantage of the periodicity of both discrete and continuous transitions, and favors the exact computation of the reachability set rather than its termination, as opposed to the widely spread approximative methods based on widening operators [HPR94,HH94]. Our method has been successfully applied to case studies for which other direct approaches fail, such as the analysis of the leaking gas burner model described in [CHR91].

Our work can be viewed as an extension of [BBR97], where only cycles that behave deterministically (w.r.t. to continuous steps) could be accelerated. It uses a more expressive model, and a more expressive symbolic representation than [CJ99,AAB00,HL02]. Furthermore, using hybrid acceleration with RVA, one can overcome the recently pointed out limitations of the representation of sets using Difference Bounds Matrices [Bou03].

Finally, it should be pointed out that Real Vector Automata provide a symbolic representation that is expressive enough to handle hybrid acceleration, and

efficient enough to handle nontrivial case studies. This motivates the integration of finite-state representations in verification tools for hybrid automata.

References

[AAB00] A. Annichini, E. Asarin, and A. Bouajjani. Symbolic Techniques for Parametric Reasoning about Counters and Clock Systems. In *Proc. of the 12th International Conference on Computer-Aided Verification (CAV)*, number 1855 in Lecture Notes in Computer Science, pages 419–434, Chicago, USA, July 2000. Springer-Verlag.

[ACH+95] 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*, 138(1):3–34, 6 February 1995.

[AD94] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 25 April 1994.

[AHH93] R. Alur, T. A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. In *Proc. 14th annual IEEE Real-Time Systems Symposium*, pages 2–11, 1993.

[BBR97] B. Boigelot, L. Bronne, and S. Rassart. An improved reachability analysis method for strongly linear hybrid systems. In *Proc. of the 9th International Conference on Computer-Aided Verification (CAV)*, number 1254 in Lecture Notes in Computer Science, pages 167–177, Haifa, Israël, June 1997. Springer-Verlag.

[BG96] B. Boigelot and P. Godefroid. Symbolic verification of communication protocols with infinite state spaces using QDDs. In *Proc. of the 8th International Conference on Computer Aided Verification (CAV)*, volume 1102 of *Lecture Notes in Computer Science*, pages 1–12, New Brunswick, NJ, USA, July/August 1996. Springer-Verlag.

[BH97] A. Bouajjani and P. Habermehl. Symbolic reachability analysis of FIFO channel systems with nonregular sets of configurations. In *Proc. of the 24th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1256 of *Lecture Notes in Computer Science*, pages 560–570, Bologna, Italy, 1997. Springer-Verlag.

[BJW01] B. Boigelot, S. Jodogne, and P. Wolper. On the use of weak automata for deciding linear arithmetic with integer and real variables. In *Proc. International Joint Conference on Automated Reasoning (IJCAR)*, volume 2083 of *Lecture Notes in Computer Science*, pages 611–625, Sienna, Italy, June 2001.

[Boi99] B. Boigelot. *Symbolic Methods for Exploring Infinite State Spaces*. Collection des publications de la Faculté des Sciences Appliquées de l'Université de Liège, Liège, Belgium, 1999.

[Bou03] P. Bouyer. Untameable timed automata! In *Proc. of 20th Ann. Symp. Theoretical Aspects of Computer Science (STACS)*, Lecture Notes in Computer Science, Berlin, Germany, February 2003. Springer-Verlag.

[BRW98] B. Boigelot, S. Rassart, and P. Wolper. On the expressiveness of real and integer arithmetic automata. In *Proc. of the 25th Colloquium on Automata, Programming, and Languages (ICALP)*, volume 1443 of *Lecture Notes in Computer Science*, pages 152–163. Springer-Verlag, July 1998.

[BSW69] K. A. Bartlett, R. A. Scantlebury, and P. T. Wilkinson. A note on reliable full-duplex transmission over half-duplex links. *Communications of the ACM*, 12(5):260–261, May 1969.

[BW94] B. Boigelot and P. Wolper. Symbolic verification with periodic sets. In D. L. Dill, editor, *Proc. of the 6th Int. Conf. on Computer Aided Verification (CAV)*, volume 818 of *Lecture Notes in Computer Science*, pages 55–67, Stanford, USA, June 1994. Springer-Verlag.

[CHR91] Z. Chaochen, C. A. R. Hoare, and A. P. Ravn. A calculus of durations. *Information Processing Letters*, 40:269–276, 1991.

[CJ99] H. Comon and Y. Jurski. Timed automata and the theory of real numbers. In *Proc. of the 10th Int. Conf. Concurrency Theory (CONCUR)*, volume 1664 of *Lecture Notes in Computer Science*, pages 242–257, Eindhoven, The Netherlands, August 1999. Springer-Verlag.

[Dil89] D. L. Dill. Timing assumptions and verification of finite-state concurrent systems. In *Proc. of Automatic Verification Methods for Finite-State Systems*, number 407 in Lecture Notes in Computer Science, pages 197–212. Springer-Verlag, 1989.

[Hen96] T. A. Henzinger. The theory of hybrid automata. In *Proc. of the 11th Annual IEEE Symposium on Logic in Computer Science*, pages 278–292, New Brunswick, New Jersey, 27–30 July 1996. IEEE Computer Society Press.

[HH94] T. A. Henzinger and P.-H. Ho. Model checking strategies for linear hybrid systems. In *Proc. of Workshop on Formalisms for Representing and Reasoning about Time*, May 1994.

[HKPV98] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57:94–124, 1998.

[HL02] M. Hendriks and K. G. Larsen. Exact acceleration of real-time model checking. *Electronic Notes in Theoretical Computer Science*, 65(6), April 2002.

[HPR94] N. Halbwachs, Y.-E. Proy, and P. Raymond. Verification of linear hybrid systems by means of convex approximations. In *Proc. of the Int. Symposium on Static Analysis*, volume 818 of *Lecture Notes in Computer Science*, pages 223–237. Springer-Verlag, 1994.

[LASH] The Liège Automata-based Symbolic Handler (LASH). Available at : `http://www.montefiore.ulg.ac.be/~boigelot/research/lash/`.

[Löd01] C. Löding. Efficient minimization of deterministic weak $\omega-$automata. *Information Processing Letters*, 79(3):105–109, 2001.

[PLY99] G. J. Pappas, G. Lafferriere, and S. Yovine. A new class of decidable hybrid systems. In *Proc. of Hybrid Systems: Computation and Control (HSCC)*, volume 1569 of *Lecture Notes in Computer Science*, pages 137–151. Springer-Verlag, 1999.

[WB95] P. Wolper and B. Boigelot. An automata-theoretic approach to Presburger arithmetic constraints. In *Proc. of the 2nd Int. Symp. on Static Analysis (SAS)*, volume 983 of *Lecture Notes in Computer Science*, pages 21–32. Springer-Verlag, 1995.

[Wei99] V. Weispfenning. Mixed real-integer linear quantifier elimination. In *Proc. of the 1999 International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 129–136, New York, July 1999. ACM Press.

[Wey50] H. Weyl. The elementary theory of convex polyhedra. *Annals of Math. Study*, 24, 1950.