The Power of Hybrid Acceleration

Bernard Boigelot, Frédéric Herbreteau

To cite this version:

Bernard Boigelot, Frédéric Herbreteau. The Power of Hybrid Acceleration. Computer Aided Verification, 18th International Conference, Aug 2006, Seattle, WA, United States. pp.438-451. inria-00335905

HAL Id: inria-00335905
https://hal.inria.fr/inria-00335905

Submitted on 31 Oct 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Power of Hybrid Acceleration

Bernard Boigelot[1]* and Frédéric Herbreteau[2]**

[1] Institut Montefiore, B28
Université de Liège
B-4000 Liège, Belgium
boigelot@montefiore.ulg.ac.be

[2] LaBRI
351, cours de la Libération
33405 Talence Cedex, France
frederic.herbreteau@labri.fr

Abstract. This paper addresses the problem of computing symbolically the set of reachable configurations of a linear hybrid automaton. A solution proposed in earlier work consists in exploring the reachable configurations using an *acceleration* operator for computing the iterated effect of selected control cycles. Unfortunately, this method imposes a periodicity requirement on the data transformations labeling these cycles, that is not always satisfied in practice. This happens in particular with the important subclass of *timed automata*, even though it is known that the paths of such automata have a periodic behavior.
The goal of this paper is to broaden substantially the applicability of hybrid acceleration. This is done by introducing powerful reduction rules, aimed at translating hybrid data transformations into equivalent ones that satisfy the periodicity criterion. In particular, we show that these rules always succeed in the case of timed automata. This makes it possible to compute an exact symbolic representation of the set of reachable configurations of a linear hybrid automaton, with a guarantee of termination over the subclass of timed automata. Compared to other known solutions to this problem, our method is simpler, and applicable to a much larger class of systems.

1 Introduction

Hybrid automata [Hen96] provide a convenient formalism for reasoning about systems that combine discrete and continuous features. A hybrid automaton is basically a finite-state machine extended with real variables, equipped with a dual semantics: A configuration can evolve either in a continuous way by spending some time at a control location (*time step*), or in a discrete way by following a transition (*discrete step*).

* The work of this author was done in part while visiting the LaBRI.

** The work of this author was supported by *Persée*, a project funded by the *ACI Sécurité Informatique* of the French Ministry of Scientific Research.

This paper considers *linear hybrid automata*, a subclass of hybrid automata with a semantics essentially defined in terms of linear constraints. Linear hybrid automata are well suited for modeling discrete systems operating in a real-time environment. Indeed, their variables can be used not only as real-valued clocks for dealing with continuous time, but also as general-purpose integer counters.

In order to analyze the reachability properties of a linear hybrid automaton, a classical solution is to explore symbolically its state space [ACH$^+$95]. This consists in starting from the initial configuration, and then repeatedly applying time steps and discrete steps to obtain new reachable configurations. Since a time step generally leads to an uncountable number of configurations, one groups such configurations into *regions* that can be manipulated implicitly, with the help of a suitable data structure.

The drawback of this approach is that state-space exploration terminates only if the reachable state space is covered by a finite number of regions, in which case the hybrid automaton is essentially equivalent to a finite-state *region automaton*. This prevents from analyzing models in which the expressive power of linear hybrid automata is used for describing unbounded discrete features. For instance, the model of an idealized communication protocol may define, in addition to the clocks dealing with the timed aspects, variables for representing unbounded message sequence numbers. A discrete variable may also be used as a parameter for reasoning about an infinite family of similar models.

However, techniques are known for exploring symbolically the state space of infinite discrete systems. A solution is to add to the semantics of a system *meta-transitions*, which are objects that capture the effect of iterating control cycles [Boi98]. With meta-transitions, state-space exploration algorithms are able to compute in one step the reachable configurations obtained by following arbitrarily many times some loops of the system under analysis. Meta-transitions thus *accelerate* state-space exploration, and can make it terminate in some cases.

In order to add meta-transitions to a linear hybrid automaton, one needs a data structure for representing sets of configurations, as well as a decision procedure for checking whether the unbounded iteration of a given loop can be computed over this data structure. For the former problem, a suitable representation system, the *Real Vector Automaton (RVA)* [BBR97,BJW05] has been developed. One can effectively represent with RVA all the sets that are definable in the first order additive theory of the integer and real numbers $\langle \mathbf{R}, \mathbf{Z}, +, \leq \rangle$. This covers linear constraints, but also discrete unbounded periodicities.

The latter problem has also been investigated in earlier work. In [BBR97], one adapts to linear hybrid automata the meta-transition computation algorithms known for unbounded integer systems. With this method, only cycles with a deterministic behavior can be turned into meta-transitions, which conflicts with the inherently branching nature of timed steps. Another technique is developed in [BHJ03], which characterizes precisely the data transformations that label paths of timed automata, and give a sufficient *periodicity* criterion for constructing meta-transitions corresponding to the iteration of such paths.

Unfortunately, this periodicity criterion is not always satisfied in practice. This happens in particular with *timed automata*, which are a restricted subclass of linear hybrid automata. However, it is known that the data transformations labeling the paths of timed automata have a periodic behavior [CJ98], and that their reachability set can be computed within $\langle \mathbf{R}, \mathbf{Z}, +, \leq \rangle$ [CJ99]. Moreover, the shortcomings of finite-state approaches to exploring timed automata [Bou03,BY03,BLR05] provide an incentive to develop alternate solutions.

In [BHJ03], a simple reduction rule was introduced for translating linear hybrid transformations into equivalent ones that satisfy the periodicity criterion. In this paper, we generalize this idea to a much broader class of transformations, by developing several new complementary reduction rules. In particular, we show that the iteration of the multiple-counters systems studied in [CJ98] can be systematically reduced to iterating periodic linear hybrid transformations. As a secondary contribution, we provide a simpler proof of the central result of [CJ98].

For the particular case of timed automata, our algorithms make it possible to carry out symbolic state-space exploration without resorting to abstraction, and with a guarantee of termination, which solves the problems reported in [Bou03,BY03,BLR05]. Although this result was already achievable as a consequence of [CJ99], our method is simpler, and applicable to a much larger class of systems.

2 Linear Hybrid Automata

2.1 Syntax and Semantics

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

Definition 1. *A* Linear Hybrid Automaton (LHA) *[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*, or* clocks*, 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 $(\boldsymbol{l}, \boldsymbol{u}) \in \mathbf{Z}^n \times \mathbf{Z}^n$*, 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

- $Q = V \times \mathbf{R}^n$ 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 $\boldsymbol{x}' \in I(v')$ and there exists $e \in E$ such that $e = (v, v')$, $\boldsymbol{x} \in G(e)$ and $(\boldsymbol{x}, \boldsymbol{x}') \in A(e)$. Such a transition can also be denoted $(v, \boldsymbol{x}) \xrightarrow{e}_\delta (v', \boldsymbol{x}')$ 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 $t \in \mathbf{R}_{\geq 0}$ 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* of a LHA is the set of its reachable configurations.

2.2 Linear Hybrid Relations

Let $\mathcal{H} = (\boldsymbol{x}, V, E, v_0, X_0, G, A, I, R)$ be a LHA, and let $\sigma = e_1; e_2; \ldots; e_p$, with $p > 0$ and $\forall i \in [1, \ldots, p] : e_i \in E$, be a path in its control graph. Let v_1, v_2, …, $v_{p+1} \in V$ be the locations successively visited by σ.

Following σ from a configuration $(v_1, \boldsymbol{x})$ to a configuration $(v_{p+1}, \boldsymbol{x}')$, denoted $(v_1, \boldsymbol{x}) \xrightarrow{\sigma} (v_{p+1}, \boldsymbol{x}')$, amounts to performing a time step at location v_1, followed by a discrete step through e_1, then a time step at v_2, …, ending with a time step at v_{p+1}. This can only be done provided that $\boldsymbol{x}$, $\boldsymbol{x}'$, and all intermediate clock values visited along σ, satisfy some linear constraints derived from the semantics of time steps and discrete steps. Projecting out all intermediate variables from these constraints, one obtains that $(v_1, \boldsymbol{x}) \xrightarrow{\sigma} (v_{p+1}, \boldsymbol{x}')$ iff $\boldsymbol{x}$ and $\boldsymbol{x}'$ are linked by a relation θ_σ that has the following form [BHJ03].

Definition 2. *A* Linear Hybrid Relation (LHR) *is a relation*

$$\theta = \left\{ (\boldsymbol{x}, \boldsymbol{x}') \in \mathbf{R}^n \times \mathbf{R}^n \,\middle|\, P \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}' \end{bmatrix} \preceq \boldsymbol{q} \right\},$$

with $n > 0, P \in \mathbf{Z}^{m \times 2n}$*,* $\boldsymbol{q} \in \mathbf{Z}^m$*,* $\preceq \in \{<, \leq\}^m$*,and* $m > 0$.

Note that a LHR is fully characterized by its linear system $(P, \boldsymbol{q}, \preceq)$, and that P can be decomposed into $[P_1; P_2]$, with $P_1, P_2 \in \mathbf{Z}^{m \times n}$. In the sequel, we will denote such a LHR as either $(P, \boldsymbol{q}, \preceq)$, or $(P_1, P_2, \boldsymbol{q}, \preceq)$. We will also write $\theta(S)$ as a shorthand for $\{\boldsymbol{x}' \mid \exists \boldsymbol{x} \in S : (\boldsymbol{x}, \boldsymbol{x}') \in \theta\}$.

3 Acceleration

3.1 Meta-Transitions

The idea behind acceleration is to capture the effect of selected *cycles* in the control graph (V, E) of the LHA $\mathcal{H}$ being analyzed, i.e., paths that start and end in the same control location.

Let σ be such a cycle, starting from the location $v_1 \in V$. The *meta-transition* [Boi98] corresponding to σ is defined as the relation $\overset{\sigma^*}{\to}$ such that $(v, \boldsymbol{x}) \overset{\sigma^*}{\to} (v', \boldsymbol{x'})$ iff $v = v' = v_1$ and $(\boldsymbol{x}, \boldsymbol{x}') \in \theta^*_\sigma$, where $\theta^*_\sigma = \cup_{i \geq 0} \theta^i_\sigma$.

Intuitively, following a meta-transition once leads in one step to all the configurations that could be reached by iterating its underlying cycle arbitrarily many times. Adding meta-transitions to the transition relation of a system thus preserves its semantics, but speeds up, or *accelerates*, state-space exploration, making it possible to explore in finite time some infinite region automata (though not all of them).

The practical use of meta-transitions requires a decision procedure for checking whether the closure θ^* of a given transformation θ can effectively be constructed and computed over sets of data values that are symbolically representable. We describe in the next section a symbolic representation system suited for analyzing linear hybrid automata.

3.2 Real Vector Automata

In order to explore symbolically the state-space of a linear hybrid automaton, one needs a data structure for representing the sets of configurations that have to be handled. Since hybrid automata have a finite number of control locations, it is actually sufficient to represent symbolically sets of clock values, that is, subsets of $\mathbf{R}^n$.

When only time steps and discrete steps are performed, the sets to be represented are characterized by conjunctions of linear constraints, and thus correspond to convex polyhedra. However, following meta-transitions may produce sets that cannot be expressed as finite unions of polyhedra. For instance, think of a cycle that has the effect of adding a constant set of values S_0 to the current clock value. The meta-transition associated to this cycle would transform a set S of clock values into the set $\cup_{i \in \mathbf{N}} S + iS_0$. We say that sets of this form have a *periodic structure*.

Real Vector Automata (RVA) [BBR97] have been introduced as effective data structures for representing convex and non convex polyhedra, as well as sets with a periodic structure. A RVA is, essentially, a finite-state automaton recognizing the infinite-word encodings of real vectors in some base $r > 1$. It is shown in [BJW05] that RVA are able to represent all the sets that are definable in $\langle \mathbf{R}, \mathbf{Z}, +, \leq \rangle$, i.e., the first-order additive theory of integer and real numbers. It is known that these sets essentially correspond to those that have a periodic structure [Wei99]. Efficient algorithms have been developed for constructing and manipulating RVA, that do not rely on the costly mechanisms usually associated to infinite-word automata. An implementation of RVA is available in the framework of the LASH toolset [LASH].

3.3 Acceleration of Linear Hybrid Relations

Let σ be a cycle of a LHA $\mathcal{H}$, and let θ_σ be its associated LHR. This cycle can be turned into a meta-transition if unbounded iterations of θ_σ preserve the representable nature of sets. This property is formalized by the following definition.

Definition 3. *The LHR θ_σ is* iterable *iff, for each set $S \subseteq \mathbf{R}^n$ definable in $\langle \mathbf{R}, \mathbf{Z}, +, \leq \rangle$, the set $\theta_\sigma^*(S)$ is definable in the same theory.*

The following sufficient criterion for iterability is given in [BHJ03].

Theorem 1. *Let $\theta_\sigma = (P, \boldsymbol{q}, \preceq)$. If the system $(P, \boldsymbol{q}, \preceq)$ over $\boldsymbol{x}$ and $\boldsymbol{x'}$ is only composed of constraints of the form $\boldsymbol{p}.\boldsymbol{x}\#q$, $\boldsymbol{p}.\boldsymbol{x}'\#q$, and $\boldsymbol{p}.(\boldsymbol{x'} - \boldsymbol{x})\#q$, with $\boldsymbol{p} \in \mathbf{Z}^n$, $q \in \mathbf{Z}$, and $\# \in \{<, \leq\}$, then θ_σ is iterable.*

LHR that satisfy the hypotheses of Theorem 1 are said to be *periodic*. Intuitively, if a LHR θ_σ is a conjunction of constraints of the form $\boldsymbol{p}.(\boldsymbol{x'} - \boldsymbol{x})\#q$, then its effect consists in adding a constant convex polyhedron Π to the current clock value, i.e., $\theta_\sigma(S) = S + \Pi$ for all sets $S \subseteq \mathbf{R}^n$. One thus has, for each $k \geq 0$, $\theta_\sigma^k(S) = S + k\Pi$, hence θ_σ^k is a conjunction of constraints of the form $\boldsymbol{p}.(\boldsymbol{x'} - \boldsymbol{x})\#kq$. One can then compute θ_σ^* within $\langle \mathbf{R}, \mathbf{Z}, +, \leq \rangle$ by quantifying k over the natural integers. For any set $S \subseteq \mathbf{R}^n$, we thus have $\theta_\sigma^*(S) = \cup_{i \in \mathbf{N}} S + i\Pi$, which has a periodic structure.

Constraints of the form $\boldsymbol{p}.\boldsymbol{x}\#q$ or $\boldsymbol{p}.\boldsymbol{x}'\#q$ are handled using a convexity argument. When a system of periodic constraints is iterated k times from a clock value $\boldsymbol{x}$ to a value $\boldsymbol{x'}$, one can always place the intermediate values $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-1}$ produced by the successive iterations onto the straight line joining $\boldsymbol{x}$ and $\boldsymbol{x}'$. Since the constraints are convex, it is thus sufficient to enforce them on $\boldsymbol{x}_1$ and $\boldsymbol{x}_{k-1}$, which can be done by a simple construction [BHJ03].

4 Reduction Rules

The iterability criterion expressed by Theorem 1 is not sufficient for identifying all LHR θ such that $\theta^*(S)$ has a periodic structure for any $S \in \mathbf{R}^n$. This restriction is problematic in practice, since simple case studies such as the classical *leaking gas burner* [ACH$^+$95] cannot be handled.

In this section, we develop *reduction rules* aimed at broadening substantially the scope of Theorem 1. The approach consists in considering LHR θ that are not periodic, such as those given in Figure 1, and then try to express their iterated effect in terms of that of a periodic LHR θ'. Precisely, we say that θ is *reducible to* θ' if $\theta^k(S)$ can be expressed in terms of $(\theta')^k(S)$ within $\langle \mathbf{R}, \mathbf{Z}, +, \leq \rangle$, for any $S \subseteq \mathbf{R}^n$ and $k > 0$. Concretely, that θ reduces to θ' entails that an algorithm for computing $(\theta')^*$ can straightforwardly be transformed into one computing θ^*.

We first generalize in Section 4.1 a reduction rule introduced in [BHJ03], and then propose new rules in Sections 4.2 and 4.3.

4.1 Subspace Reduction

Consider the non-periodic LHR θ_1 in Figure 1. This relation transforms $\mathbf{R}^2$ into a set of smaller dimension, namely $E_1 = \theta_1(\mathbf{R}^2) = \{(x, x-1) \mid x \in \mathbf{R}\}$. Hence,

$\theta_1 \equiv \begin{cases} x'_1 + x'_2 = 2x_1 + 1 \\ x'_1 - x'_2 = 1 \end{cases}$	$\theta_2 \equiv \begin{cases} x'_1 + x'_2 = 2x_1 + 1 \\ x_1 - x_2 = 1 \end{cases}$	$\theta_3 \equiv \begin{cases} x'_1 + x'_2 = x_1 + x_2 + 1 \\ x'_1 - x'_2 \geq x_1 + x_2 \end{cases}$
$\theta_4 \equiv \begin{cases} x'_1 + x'_2 \leq 2x_1 + 1 \\ x'_1 + x'_2 \geq x_1 + x_2 \\ x'_1 + x'_2 \geq x_1 - x_2 \end{cases}$	$\theta_5 \equiv \begin{cases} x'_1 + x'_2 \leq 2x_1 + 1 \\ x'_1 + x'_2 \geq x_1 + x_2 \\ x'_1 + x'_2 \geq x_1 - x_2 \\ x'_1 - x'_2 \leq 2 \end{cases}$	$\theta_6 \equiv \begin{cases} x'_1 + x'_2 \leq 2x_1 + 1 \\ x'_1 + x'_2 \geq x_1 + x_2 \\ x'_1 + x'_2 \geq x_1 - x_2 \\ x_1 - x_2 \leq 2 \end{cases}$
$\theta_7 \equiv \begin{cases} x'_1 \geq x_2 + 1 \\ x'_2 \geq x_1 + 2 \end{cases}$	$\theta_8 \equiv \begin{cases} x'_1 \geq x_2 + 1 \\ x'_2 \geq x_1 + 2 \\ x'_3 + x'_4 \leq x_3 + x_4 + 3 \end{cases}$	$\theta_9 \equiv \begin{cases} x'_1 + x'_2 \geq x_1 - x_2 + 1 \\ x'_1 - x'_2 \geq x_1 + x_2 + 2 \\ x'_1 + x'_3 \leq x_1 + x_3 + 3 \end{cases}$

Fig. 1. Examples of non-periodic LHR.

for any set $S \subseteq \mathbf{R}^2$, the images $\theta_1(S)$, $\theta_1^2(S)$, $\theta_1^3(S)$, $\ldots$, are all subsets of E_1. It is thus sufficient to study the iterations of θ_1 in this subspace, in which it turns out to be periodic.

Formally, restricting a LHR $\theta \subseteq \mathbf{R}^n \times \mathbf{R}^n$ to a subspace E such that $\dim(E) = m$, with $m < n$, is done by a variable change operation. Let $\boldsymbol{u}_0 \in \mathbf{Z}^n$ be an arbitrary element of E, let $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m \in \mathbf{Z}^n$ be a vector basis of $E - \boldsymbol{u}_0$, and let $U \in \mathbf{Z}^{n \times m} = [\boldsymbol{u}_1; \ldots; \boldsymbol{u}_m]$. We introduce new variables $y_1, y_2, \ldots, y_m$ and $y'_1, y'_2, \ldots, y'_m$, such that $(x_1, \ldots, x_n) = U(y_1, \ldots, y_m) + \boldsymbol{u}_0$ and $(x'_1, \ldots, x'_n) = U(y'_1, \ldots, y'_m) + \boldsymbol{u}_0$. Adding these constraints to the underlying system of θ, and then projecting out the variables x_i and x'_i for all $i \in [1, \ldots, n]$, one obtains a transformation $\theta' \subseteq \mathbf{R}^m \times \mathbf{R}^m$ that has the same iterative behavior as θ. Indeed, for any $S \subseteq \mathbf{R}^n$ and $k > 0$, we have $\theta^k(S) = U(\theta')^{k-1}(S') + \boldsymbol{u}_0$, where $S' \subseteq \mathbf{R}^m$ is the solution of $\theta(S) = US' + \boldsymbol{u}_0$. It is worth emphasizing that computing S' from S and $\theta^k(S)$ from $(\theta')^{k-1}(S')$ can be done within $\langle \mathbf{R}, \mathbf{Z}, +, \leq \rangle$. We thus have the following rule.

Reduction Rule 1 *If a LHR* $\theta \subseteq \mathbf{R}^n \times \mathbf{R}^n$ *is such that* $\dim(\theta(\mathbf{R}^n)) = m$ *with* $m < n$*, then* θ *is reducible to a computable LHR* $\theta' \subseteq \mathbf{R}^m \times \mathbf{R}^m$*.*

In the case of our example θ_1, we have $\dim(E_1) = 1$, which prompts the definition of new variables y_1, y'_1 such that $x_1 = y_1$, $x_2 = y_1 - 1$, $x'_1 = y'_1$, and $x'_2 = y'_1 - 1$. The LHR θ_1 then translates into $\theta'_1 \equiv y'_1 = y_1 + 1$, which is periodic.

Rule 1 admits a dual form. The LHR θ_2 in Figure 1 is such that $\theta_2(\mathbf{R}^2) = \mathbf{R}^2$, hence Rule 1 does not apply. Notice however that θ_2 produces a nonempty result only when it is applied to values that belong to $E_2 = \theta_2^{-1}(\mathbf{R}^2) = \{(x, x-1) \mid x \in \mathbf{R}\}$. In order to study the iterations of θ_2, one can therefore ignore the values that are outside of this subspace. This leads to the following rule.

Reduction Rule 2 *If a LHR* $\theta \subseteq \mathbf{R}^n \times \mathbf{R}^n$ *is such that* $\dim(\theta^{-1}(\mathbf{R}^n)) = m$ *with* $m < n$*, then* θ *is reducible to a computable LHR* $\theta' \subseteq \mathbf{R}^m \times \mathbf{R}^m$*.*

Technically, the reduction of θ is performed in the following way. Let $E = \theta^{-1}(\mathbf{R}^n)$. We first transform θ into $\theta'' = \theta \wedge (\boldsymbol{x}' \in E)$, so as to systematically discard output values that do not belong to E. Then, we define a variable change $(x_1, \ldots, x_n) = U(y_1, \ldots, y_m) + \boldsymbol{u}_0$, with $U \in \mathbf{Z}^{n\times m}$ and $\boldsymbol{u}_0 \in \mathbf{Z}^n$, from $\mathbf{R}^n$ to E. Applying this variable change to θ'' yields a LHR $\theta' \in \mathbf{R}^m \times \mathbf{R}^m$. For each $S \subseteq \mathbf{R}^n$ and $k > 0$, we then have $\theta^k(S) = \theta(U(\theta')^{k-1}(S') + \boldsymbol{u}_0)$, where S' is the solution of $S \cap E = US' + \boldsymbol{u}_0$.

4.2 Rank Reduction

Rules 1 and 2 are not able to capture all sources of periodicity. For instance, they cannot be applied to the LHR θ_3 in Figure 1, since $\theta_3(\mathbf{R}^2) = \theta_3^{-1}(\mathbf{R}^2) = \mathbf{R}^2$.

However, remark that applying θ_3 to two vectors (a_1, a_2) and (b_1, b_2) such that $a_1 + a_2 = b_1 + b_2$ produces identical output values. Therefore, the iterations of θ_3 can be studied with respect to a single variable $y_1 = x_1 + x_2$. Like in the previous case, this variable change transforms θ_3 into a LHR of smaller dimension.

Formally, consider a LHR $\theta \in \mathbf{R}^n \times \mathbf{R}^n$. The system of constraints $(P_1, P_2, \boldsymbol{q}, \preceq)$ of θ can be rewritten as $P_2\boldsymbol{x'} \preceq -P_1\boldsymbol{x} + \boldsymbol{q}$. Let $p = \rho(P_1)$ be the rank of P_1. If $p < n$, then the possible values of $P_1\boldsymbol{x}$, and hence also of $\boldsymbol{x'}$, can be described in terms of only p independent variables. We express P_1 as a product $P_1 = P_1'U$, with $P_1' \in \mathbf{Z}^{m\times p}$ and $U \in \mathbf{Z}^{p\times n}$, and introduce new variables $y_1, \ldots, y_p, y_1', \ldots, y_p'$ such that $(y_1, \ldots, y_p) = U(x_1, \ldots, x_n)$ and $(y_1', \ldots, y_p') = U(x_1', \ldots, x_n')$. Adding these constraints to $(P_1, P_2, \boldsymbol{q}, \preceq)$, and then projecting out $x_1, \ldots, x_n$ and $x_1', \ldots, x_n'$, yields a LHR $\theta' \in \mathbf{R}^p \times \mathbf{R}^p$. For each $S \subseteq \mathbf{R}^n$ and $k > 0$, we have $\theta^k(S) = \theta''((\theta')^{k-1}(US))$, where $\theta'' \in \mathbf{R}^p \times \mathbf{R}^n \equiv P_2\boldsymbol{x'} \preceq -P_1'\boldsymbol{y} + \boldsymbol{q}$. Thus, θ is reducible to θ', which leads to the following rule.

Reduction Rule 3 *If a LHR $\theta \subseteq \mathbf{R}^n \times \mathbf{R}^n = (P_1, P_2, \boldsymbol{q}, \preceq)$ is such that $\rho(P_1) = p$ with $p < n$, then θ is reducible to a computable LHR $\theta' \subseteq \mathbf{R}^p \times \mathbf{R}^p$.*

A similar reduction can also be applied if the rank of P_2 is less than the number of variables. Consider the LHR θ_4 in Figure 1. If the vector (a_1, a_2) belongs (resp. does not belong) to $\theta_4(S)$ for some $S \subseteq \mathbf{R}^2$, then for all $(b_1, b_2) \in \mathbf{R}^2$ such that $b_1 + b_2 = a_1 + a_2$, we have $(b_1, b_2) \in S$ (resp. $(b_1, b_2) \notin S$). Thus, the behavior of θ_4 can be studied with respect to a single variable defined as $y_1 = x_1 + x_2$.

More generally, let $\theta \in \mathbf{R}^n \times \mathbf{R}^n$ be a LHR, and let $(P_1, P_2, \boldsymbol{q}, \preceq)$ be its underlying system of constraints. If $p = \rho(P_2)$ is such that $p < n$, then we decompose P_2 into $P_2 = P_2'U$, with $P_2' \in \mathbf{Z}^{m\times p}$ and $U \in \mathbf{Z}^{p\times n}$, and introduce new variables $y_1, \ldots, y_p, y_1', \ldots, y_p'$ such that $(y_1, \ldots, y_p) = U(x_1, \ldots, x_n)$ and $(y_1', \ldots, y_p') = U(x_1', \ldots, x_n')$. This variable change transforms θ into a LHR $\theta' \in \mathbf{R}^p \times \mathbf{R}^p$. For each $S \subseteq \mathbf{R}^n$ and $k > 0$, we have $\theta^k(S) = \theta''((\theta')^{k-1}(U\theta(S)))$, where $\theta'' \in \mathbf{R}^p \times \mathbf{R}^n \equiv U\boldsymbol{x'} = \boldsymbol{y}$. We therefore have the following rule.

Reduction Rule 4 *If a LHR $\theta \subseteq \mathbf{R}^n \times \mathbf{R}^n = (P_1, P_2, \boldsymbol{q}, \preceq)$ is such that $\rho(P_2) = p$ with $p < n$, then θ is reducible to a computable LHR $\theta' \subseteq \mathbf{R}^p \times \mathbf{R}^p$.*

4.3 Static Reduction

None of the reduction rules obtained so far can handle the LHR θ_5 in Figure 1. One nevertheless observes that the linear system of θ_5 contains a constraint $x'_1 - x'_2 \leq 2$ that is solely expressed over the output variables. Requiring that the value produced at the end of an iteration of θ_5 satisfies this constraint is actually equivalent to imposing $x_1 - x_2 \leq 2$ on the input value of the next iteration. Hence, the LHR θ_6 in Figure 1, on which Rule 4 can be applied, has essentially the same iterative behavior as θ_5.

Formally, let θ be a LHR. We call a constraint of θ *static* if it involves only either $x_1, \ldots, x_n$, or $x'_1, \ldots, x'_n$, and *dynamic* otherwise. A static constraint is said to be *explicit* if it is not implied by the dynamic constraints of θ. We denote by $\overline{\theta}$ the conjunction of explicit static constraints in θ. The LHR obtained from θ by rewriting over $x_1, \ldots, x_n$ (resp $x'_1, \ldots, x'_n$) all constraints in $\overline{\theta}$ is denoted θ_x (resp. $\theta_{x'}$).

For all $S \subseteq \mathbf{R}^n$ and $k > 1$, we have $\theta^k(S) = (\theta \wedge \overline{\theta_x})((\theta_x)^{k-2}(\theta(S)))$ and $\theta^k(S) = \theta((\theta_{x'})^{k-2}(\theta(S) \wedge \overline{\theta_{x'}}))$. We thus have the following rule.

Reduction Rule 5 *Every LHR θ is reducible to θ_x and $\theta_{x'}$.*

In practice, Rule 5 is only useful when it can be followed by another reduction. A simple guideline consists of reducing systematically LHR θ to θ_x before attempting to apply Rules 2 and 4, and to $\theta_{x'}$ before Rules 1 and 3.

5 Multiple Counters Systems

The combination of Rules 1 to 5 suffices for many applications. However, these rules are unable to handle relations such as θ_7 in Figure 1. This LHR is actually an instance of a *Multiple Counters System (MCS).* It is known [CJ98] that all such relations θ are iterable within $\langle \mathbf{R}, \mathbf{Z}, +, \leq \rangle$.

In this section, we give a simpler proof of that result, and use it to derive an expression of $\theta^k(S)$ in terms of $k \in \mathbf{N}$ and $S \subseteq \mathbf{R}^n$. We then show that the acceleration of MCS can be reduced to that of periodic LHR. MCS are formally defined as follows.

Definition 4. *A* Multiple Counters Systems (MCS) *[CJ98] is a relation $\theta(\boldsymbol{x}, \boldsymbol{x}') \subseteq \mathbf{R}^n \times \mathbf{R}^n$, defined by a finite conjunction of constraints of the form $z_1 \# z_2 + c$, where $z_1, z_2 \in \{x_1, \ldots, x_n, x'_1, \ldots, x'_n\}$, $\# \in \{<, \leq, \geq, >\}$, and $c \in \mathbf{Z}$.*

5.1 Acceleration of MCS

Let $\theta \subseteq \mathbf{R}^n \times \mathbf{R}^n$ be a MCS. We assume w.l.o.g. that the explicit static constraints of θ are expressed over both $x_1, x_2, \ldots, x_n$ and $x'_1, x'_2, \ldots, x'_n$.

Our goal is to construct within $\langle \mathbf{R}, \mathbf{Z}, +, \leq \rangle$ an expression for $\theta^k(\boldsymbol{x}^0, \boldsymbol{x}^k)$ in terms of the variables $\boldsymbol{x}^0$, $\boldsymbol{x}^k$ and k. For a fixed value of k, such an expression can be obtained by projecting $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^{k-1}$ out of $\theta(\boldsymbol{x}^0, \boldsymbol{x}^1) \wedge \theta(\boldsymbol{x}^1, \boldsymbol{x}^2) \wedge \cdots \wedge \theta(\boldsymbol{x}^{k-1}, \boldsymbol{x}^k)$, which can be done by Fourier-Motzkin elimination.

In this operation, the dynamic constraints of $\theta^k(\boldsymbol{x}^0, \boldsymbol{x}^k)$ are all obtained as combinations of constraints in $\theta(\boldsymbol{x}^0, \boldsymbol{x}^1)$, $\theta(\boldsymbol{x}^1, \boldsymbol{x}^2)$, $\ldots$, $\theta(\boldsymbol{x}^{k-1}, \boldsymbol{x}^k)$. Each dynamic constraint of θ^k results from combining a sequence of constraints of θ that links a variable x_i^0 to a variable x_j^k, with $i, j \in [1, \ldots, n]$, hence it takes the form $x_j^k \# x_i^0 + c$, where $\# \in \{<, \leq, \geq, >\}$ and $c \in \mathbf{Z}$. Likewise, the static constraints of θ^k correspond to sequences of constraints of θ that link either x_i^0 to x_j^0, or x_i^k to x_j^k, for some $i, j \in [1, \ldots, n]$.

This leads to a simple way of characterizing the constraints of θ^k. Inspired by [Rev93,CJ98], we build two directed graphs $G^\theta_<$ and $G^\theta_>$ in the following way. The vertices of these graphs correspond to the variables $x_1, \ldots, x_n$. The edges of $G^\theta_<$ (resp. $G^\theta_>$) are labeled by tuples $(\#, c, d)$, where $\# \in \{<, \leq\}$ (resp. $\# \in \{>, \geq\}$) is a *strictness marker*, $c \in \mathbf{Z}$ is a *cost*, and $d \in \{-1, 0, 1\}$ is a *depth*. The edges of $G^\theta_<$ and $G^\theta_>$ are created as follows:

- For each constraint $x_j \preceq x_i + c$ in θ, with $\preceq \in \{<, \leq\}$, we add the edge $(x_i, (\preceq, c, 0), x_j)$ to $G^\theta_<$, and the edge $(x_j, (\succeq, -c, 0), x_i)$ to $G^\theta_>$;
- For each constraint $x'_j \preceq x_i + c$ in θ, we add the edge $(x_i, (\preceq, c, 1), x_j)$ to $G^\theta_<$, and the edge $(x_j, (\succeq, -c, -1), x_i)$ to $G^\theta_>$;
- For each constraint $x'_j \succeq x_i + c$ in θ, we add the edge $(x_j, (\preceq, -c, -1), x_i)$ to $G^\theta_<$, and the edge $(x_i, (\succeq, c, 1), x_j)$ to $G^\theta_>$.

For each path π in $G^\theta_<$ or $G^\theta_>$, we define its *strictness* $s(\pi)$ as the strongest marker labeling the transitions followed by π, and its *cost* $c(\pi)$ and *depth* $d(\pi)$ as the sums of the individual cost and depth of all these transitions. The *absolute depth* of π is $|d(\pi)|$. The minimum (resp. maximum) depth $d_-(\pi)$ (resp. $d_+(\pi)$) of π is defined as the smallest (resp. largest) depth among all prefixes of π. Intuitively, a path π of $G^\theta_<$ or $G^\theta_>$ linking a variable x_i to a variable x_j represents constraints $x_j^{d'} \# x_i^d + c$, where $\#$ corresponds to the strictness of π, $d' - d$ to its depth, and c to its cost. The minimum and maximum depth of π then bound the superscripts of the intermediate variables that are visited by π, and that are thus projected out when the constraints of θ are combined.

Proposition 1. *Each dynamic constraint $x_j^k \# x_i^0 + c$ of $\theta^k(\boldsymbol{x}^0, \boldsymbol{x}^k)$, with $\# \in \{<, \leq, \geq, >\}$ and $c \in \mathbf{Z}$, corresponds to a path π from x_i to x_j in either $G^\theta_<$ or $G^\theta_>$, such that $s(\pi) = \#$, $c(\pi) = c$, $d(\pi) = k$, $d_-(\pi) = 0$, and $d_+(\pi) = k$. Similarly, static constraints $x_j^0 \# x_i^0 + c$ correspond to paths π from x_i to x_j such that $s(\pi) = \#$, $c(\pi) = c$, $d(\pi) = 0$, $d_-(\pi) = 0$, and $d_+(\pi) \leq k$. Finally, static constraints $x_j^k \# x_i^k + c$ correspond to paths π from x_i to x_j such that $s(\pi) = \#$, $c(\pi) = c$, $d(\pi) = 0$, $d_-(\pi) \geq -k$, and $d_+(\pi) = 0$.*

The problem of computing the constraints of $\theta^k(\boldsymbol{x}^0, \boldsymbol{x}^k)$ has thus been reduced to that of characterizing, in terms of k, the costs of the paths of depths 0 and k that link two given variables in $G^\theta_<$ and $G^\theta_>$, without exceeding some minimum and maximum depths. Note that, in the case of multiple paths, it is sufficient to consider the strongest constraints, which correspond to the paths with the minimal cost in $G^{\theta}_{<}$ and with the maximal cost in $G^{\theta}_{>}$.

We now show that this characterization can be carried out with a bounded construction. Let π be a path of $G^{\theta}_{<}$ (the graph $G^{\theta}_{>}$ is handled symmetrically).

- *If π contains occurrences of a simple loop σ, i.e., $\pi = \pi_1 \sigma^{k_1} \pi_2$ with $k_1 > 0$, such that $d(\sigma) = 0$ and $c(\sigma) \geq 0$.* Then the path $\pi' = \pi_1 \pi_2$ has the same depth as π, but a smaller or equal cost.
- *If π contains occurrences of a simple loop σ such that $d(\sigma) = 0$ and $c(\sigma) < 0$.* Then π represents an unsatisfiable constraint.
- *If π contains occurrences of two simple loops σ_1 and σ_2, i.e., we have $\pi = \pi_1 \sigma_1^{k_1} \pi_2 \sigma_2^{k_2} \pi_3$ or $\pi = \pi_1 \sigma_2^{k_2} \pi_2 \sigma_1^{k_1} \pi_3$, such that either $d(\sigma_1) > 0$ and $d(\sigma_2) > 0$, or $d(\sigma_1) < 0$ and $d(\sigma_2) < 0$, and $c(\sigma_1)/|d(\sigma_1)| \leq c(\sigma_2)/|d(\sigma_2)|$.* Then, removing $|d(\sigma_1)|/g$ occurrences of σ_2 and adding $|d(\sigma_2)|/g$ occurrences of σ_1, with $g = \gcd(|d(\sigma_1)|, |d(\sigma_2)|)$, transforms π into a path with the same depth, but a smaller or equal cost.
- *If π contains occurrences of two simple loops σ_1 and σ_2 such that $d(\sigma_1) < 0$, $d(\sigma_2) > 0$, and $c(\sigma_1)/d(\sigma_1) \geq c(\sigma_2)/d(\sigma_2)$.* Then removing $-d(\sigma_1)/g$ occurrences of σ_2 and $d(\sigma_2)/g$ occurrences of σ_1 from π, where $g = \gcd(-d(\sigma_1), d(\sigma_2))$, yields a path that has the same depth as π, but a smaller or equal cost.
- *If π contains occurrences of two simple loops σ_1 and σ_2 such that $d(\sigma_1) < 0$, $d(\sigma_2) > 0$, and $c(\sigma_1)/d(\sigma_1) < c(\sigma_2)/d(\sigma_2)$.* For any $l > 0$, adding $-l.d(\sigma_1)/g$ occurrences of σ_2 and $l.d(\sigma_2)/g$ occurrences of σ_1, with $g = \gcd(-d(\sigma_1), d(\sigma_2))$, transforms π into a path that has the same depth, but a smaller or equal cost. In this case, we thus obtain the strongest constraint by selecting the largest value of l for which the minimum and maximum path depths are not exceeded. The path π is then split into $\pi = \pi_1 \pi_2$, such that each subpath π_1 or π_2 contains only iterations of either σ_1 or σ_2, and the split point maximizes the depth of the subpath π_1 or π_2 that contains the iterations of σ_2. Since the constraints represented by π are implied by those corresponding to π_1 and π_2, these two paths can now be considered individually.

Let $l_<$ (resp. $l_>$) be the least common multiple of the absolute depths of the simple cycles in $G^{\theta}_{<}$ (resp. $G^{\theta}_{>}$). Applying repeatedly the above transformations, the paths of $G^{\theta}_{<}$ are eventually replaced by ones in which all occurrences of simple loops but one have an absolute depth less than $l_<$. We thus have a simple algorithm for iterating θ:

1. Compute $l = \mathrm{lcm}(l_<, l_>)$. Since l is fixed, θ is reducible to θ^l;
2. In order to obtain the constraints of $(\theta^l)^k(\boldsymbol{x}^0, \boldsymbol{x}^k)$, it is sufficient to consider the paths of $G^{\theta^l}_{<}$ and $G^{\theta^l}_{>}$ with a depth $d \in \{0, k\}$, that are either acyclic or of the form $\pi_1 \sigma^{d - d(\pi_1) - d(\pi_2)} \pi_2$, where π_1 and π_2 are acyclic, and σ is a simple cycle of absolute depth 1. These paths can be inspected in bounded time. For each of them, one must also ensure that the minimum and maximum depth constraints imposed by Proposition 1 are satisfied.

Recall that σ represents a constraint of the form $x'_i \# x_i + c$, hence σ^{k-d_0}, with $d_0 = d(\pi_1) + d(\pi_2)$, corresponds to $x'_i \# x_i + (k - d_0)c$. The dynamic constraints of $(\theta^l)^k(\boldsymbol{x}^0, \boldsymbol{x}^k)$ are thus obtained in the form $k \geq q \Rightarrow x^k_j \# x^0_i + k\delta + \gamma$, with $\delta, \gamma \in \mathbf{Z}$, and $q \in \mathbf{N}$.

5.2 MCS and Periodic LHR

The results of the previous section give an interesting insight into the iterative behavior of MCS. For any MCS θ, we now know that there exists $l > 0$ such that $(\theta^l)^k$ can be decomposed into $(\theta^l)^k = \theta_0 \cup \bigcup_{1 \leq i \leq p} \theta_i \circ (\theta')^{k-k_i} \circ \theta'_i$, where $p \geq 0$, θ' is a periodic LHR, $\theta_0, \theta_1, \ldots, \theta_p, \theta'_1, \ldots, \theta'_p$ are LHR, and $k_1, \ldots, k_p \in \mathbf{N}$. We therefore have the following result.

Reduction Rule 6 *Every MCS θ is reducible to a periodic LHR θ'.*

In practice, we iterate a given LHR θ by first applying all possible reductions rules from 1 to 5, and then checking whether the resulting system forms a periodic LHR, a MCS, or a conjunction of both. In the last situation, we can iterate θ provided that the periodic LHR θ_P and the remaining MCS θ_M that compose θ are defined over distinct subsets of variables. Indeed, assuming w.l.o.g. that the variables of θ_P precede these of θ_M, we have $\theta^k(S) = \theta^k_P(S_P) \times \theta^k_M(S_M)$, for all $S \subseteq \mathbf{R}^n$ and $k > 0$, where S_P and S_M are the projections of S over the variables of θ_P and θ_M. The LHR θ_8 in Figure 1 provides an example of relation that can be handled in this way.

This approach has the shortcoming that the class of LHR that can be iterated is not closed under linear transformations, e.g., the LHR θ_9 in Figure 1, which is functionally equivalent to θ_8, cannot be handled. A method that lifts this restriction will be investigated in another paper.

6 Application to Timed Automata

Definition 5. *A* Timed Automaton (TA) *is a LHA $(\boldsymbol{x}, V, E, v_0, X_0, G, A, I, R)$ such that*

- *its initial region X_0, guard $G(e)$ of each edge $e \in E$, and invariant $I(v)$ of each location $v \in V$ are conjunctions of constraints of the form $x_i \# c$ and $x_i - x_j \# c$, with $\# \in \{<, \leq, =, \geq, >\}$ and $c \in \mathbf{Z}$;*
- *the assignment $A(e)$ of each edge $e \in E$ has the form $\bigwedge_{i=0}^{n} x'_i = d_i x_i$, with $\forall i \in [1, \ldots, n] : d_i \in \{0, 1\}$. In other words, a transition can either reset a clock, or leave it unchanged;*
- *the activity $R(v)$ of each location $v \in V$ equals $((1, \ldots, 1), (1, \ldots, 1))$, i.e., all clocks increase uniformly with time.*

It is shown in [Fri98,CJ98] that the LHR that label arbitrary paths of timed automata can be turned into MCS by a simple variable change operation. Let π be such a path. The idea consists in defining one new global clock t that is never reset, and that will serve as a reference for relating the values of other variables. Then, for each clock x_i of the TA that is reset along π, or that is evaluated without having been reset, one introduces a new variable t_i such that $x_i = t - t_i$. Let $t^0, t_1^0, t_2^0, \ldots$ denote the initial values, and $t', t_1', t_2', \ldots$ the final values, of $t, t_1, t_2, \ldots$ with respect to π. Intuitively, each t_i' gives the *date*, expressed with respect to the reference timeframe, at which the corresponding clock x_i has been last reset. Expressed over $t^0, t_1^0, t_2^0, \ldots$ and $t', t_1', t_2', \ldots$, the LHR induced by π takes the form of a MCS [Fri98,CJ98].

Together with Rule 6, this result gives an effective algorithm for turning any cycle of a TA into a meta-transition. We now recall a property established in [CJ99].

Theorem 2. *For any TA, there exists a finite choice of meta-transitions for which symbolic state-space exploration terminates.*

Algorithms are given in [Boi98,BLFP03] for discovering automatically such a choice of meta-transitions whenever one exists. It is thus possible to guarantee that exploring symbolically the state-space of timed automata with hybrid acceleration terminates.

7 Conclusions

The contribution of this paper is to show that, for a large class of linear hybrid relations θ, computing θ^* reduces to iterating the periodic relations considered in [BHJ03]. This broadens substantially the applicability of hybrid acceleration, and provides a powerful framework for reasoning about linear hybrid automata.

A secondary contribution is to provide a simpler proof of the acceleration result for multiple counters systems given in [CJ98]. We have established that the iterative behavior of such systems reduces to that of periodic relations, which brings their acceleration algorithm closer to an actual implementation.

For the particular case of timed automata, an exact state-space exploration algorithm was already known [CJ99]. Compared to this method, the advantage of our approach is to be applicable to a much larger class of systems (although obviously without a general guarantee of termination). For large systems, we also expect our technique to scale up much more nicely than [CJ99]. Indeed, iterating a periodic hybrid relation in $\langle \mathbf{R}, \mathbf{Z}, +, \leq \rangle$ is fundamentally very close to iterating a linear transformation within $\langle \mathbf{Z}, +, \leq \rangle$, as done by the NDD package of LASH [LASH]. Although this conjecture remains to be substantiated with actual experiments, we believe that adding timed constraints to the case studies performed with LASH will not significantly complicate their analysis.

Acknowledgments

We would like to thank the reviewers of this paper for their insightful comments on the MCS acceleration algorithm described in Section 5.1.

References

[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, 1995.

[BBR97] B. Boigelot, L. Bronne, and S. Rassart. An improved reachability analysis method for strongly linear hybrid systems. In *Proceedings of the 9th Int. Conf. on Computer-Aided Verification (CAV'97)*, volume 1254 of *Lecture Notes in Computer Science*, pages 167–177. Springer-Verlag, 1997.

[BHJ03] B. Boigelot, F. Herbreteau, and S. Jodogne. Hybrid acceleration using real vector automata. In *Proc. of the 15th Int. Conf. on Computer-Aided Verification (CAV'03)*, volume 2725 of *Lecture Notes in Computer Science*, pages 193–205. Springer-Verlag, 2003.

[BJW05] B. Boigelot, S. Jodogne, and P. Wolper. An effective decision procedure for linear arithmetic with integer and real variables. *ACM Transactions on Computational Logic (TOCL)*, 6(3):614–633, 2005.

[BLFP03] S. Bardin, J. Leroux, A. Finkel, and L. Petrucci. FAST: Fast accelereation of symbolic transition systems. In *Proc. 15th Int. Conf. on Computer-Aided Verification*, volume 2725 of *Lect. Notes in Comp. Sc.*, pages 118–121, 2003.

[BLR05] P. Bouyer, F. Laroussinie, and P.-A. Reynier. Diagonal constraints in timed automata: Forward analysis of timed systems. In *Proc. of the 3rd Int. Conf. on Formal Modelling and Analysis of Timed Systems*, volume 3829 of *Lecture Notes in Computer Science*, pages 112–126. Springer-Verlag, 2005.

[Boi98] B. Boigelot. *Symbolic Methods for Exploring Infinite State Spaces.* PhD thesis, Université de Liège, 1998.

[Bou03] P. Bouyer. Untameable timed automata! In *Proc. of 20th Ann. Symp. Theoretical Aspects of Computer Science (STACS'03)*, Lecture Notes in Computer Science, Berlin, Germany, 2003. Springer-Verlag.

[BY03] J. Bengtsson and W. Yi. On clock difference constraints and termination in reachability analysis of timed automata. In *Proc. of 5th Int. Conf. on Formal Engineering Methods (ICFEM'03)*, volume 2885 of *Lecture Notes in Computer Science*, pages 491–503, 2003.

[CJ98] H. Comon and Y. Jurski. Multiple counters automata, safety analysis and Presburger arithmetic. In *Proc. of 10th Int. Conf. on Computer-Aided Verification (CAV'98)*, volume 1427 of *Lecture Notes in Computer Science*, pages 268–279. Springer-Verlag, 1998.

[CJ99] H. Comon and Y. Jurski. Timed automata and the theory of real numbers. In *Proc. 10th Int. Conf. Concurrency Theory (CONCUR'99)*, volume 1664 of *Lecture Notes in Computer Science*, pages 242–257. Springer-Verlag, 1999.

[Fri98] L. Fribourg. A closed-form evaluation for extended timed automata. Research Report LSV-98-2, LSV, March 1998.

[Hen96] T. A. Henzinger. The theory of hybrid automata. In *Proc. 11th Annual Symp. on Logic in Computer Science (LICS)*, pages 278–292. IEEE Computer Society Press, 1996.

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

[Rev93] P. Z. Revesz. A closed-form evaluation for datalog queries with integer (gap)-order constraints. *Theor. Comp. Sc.*, 116(1&2):117–149, 1993.

[Wei99] V. Weispfenning. Mixed real-integer linear quantifier elimination. In *ISSAC: Proceedings of the ACM SIGSAM Int. Symp. on Symbolic and Algebraic Computation*, pages 129–136, Vancouver, 1999. ACM Press.