Acceleration of Affine Hybrid Transformations

Bernard Boigelot, Frédéric Herbreteau, Isabelle Mainz

Acceleration of Affine Hybrid Transformations*

Bernard Boigelot[1], Frédéric Herbreteau[2], and Isabelle Mainz[1]

[1] Institut Montefiore, B28, Univ. Liège, Belgium
{boigelot,mainz}@montefiore.ulg.ac.be

[2] Univ. Bordeaux & CNRS, LaBRI, UMR 5800, Talence, France
fh@labri.fr

Abstract. This work addresses the computation of the set of reachable configurations of linear hybrid automata. The approach relies on symbolic state-space exploration, using acceleration in order to speed up the computation and to make it terminate for a broad class of systems. Our contribution is an original method for accelerating the control cycles of linear hybrid automata, i.e., to compute their unbounded repeated effect. The idea consists in analyzing the data transformations that label these cycles, by reasoning about the geometrical features of the corresponding system of linear constraints. This approach is complete over Multiple Counters Systems (MCS), and is able to accelerate hybrid transformations that are out of scope of existing techniques.

1 Introduction

Hybrid automata [14] are a powerful formalism for modeling systems that combine discrete and continuous features, in particular those depending on physical processes that involve undiscretized time. Linear hybrid automata are a restricted form of hybrid automata that are amenable to automated analysis of some of their properties, while not sacrificing too much expressive power, which remains sufficient for modeling precisely enough a large range of systems.

This work addresses the general problem of analyzing reachability properties of linear hybrid automata, by computing an exact representation of their set of reachable configurations. Since this set is generally infinite, both because variables of hybrid automata are unbounded and take their value over a dense domain, this computation has to be performed symbolically, representing the manipulated sets with the help of dedicated data structures. Moreover, since linear hybrid automata are Turing complete, the computation of their reachability set cannot be guaranteed to terminate in all cases. A possible workaround would be to introduce approximations, such as widening operators [12], in order to force termination. We make a different choice and aim at an exact computation algorithm without guarantee of termination, trying to make it powerful enough for handling a relevant subclass of systems.

* This work is supported in part by the grant 2.4545.11 of the Belgian Fund for Scientific Research (F.R.S.-FNRS).

Computing the reachability set of a system can be achieved by forward symbolic state-space exploration: At each step, one propagates reachability information from the current set of reachable configurations in order to make it bigger. The procedure terminates upon reaching a fixed point. For hybrid automata, an exploration step corresponds to letting time elapse in the current control location, or to following a transition from one location to another.

This approach is not sufficient for analyzing all interesting case studies. One reason is that some linear hybrid automata have configurations that are only reached after an unbounded number of exploration steps; a typical example is the *leaking gas burner* studied in [15]. This problem is tackled by *acceleration* techniques, aimed at computing in finite time sets of configurations that are reached after following arbitrarily long control paths. For instance, accelerating a cyclic path, which corresponds to a loop in a program, amounts to computing in one step all the configurations that can be reached by iterating this cycle arbitrarily many times [2].

In order to be able to perform cycle acceleration with linear hybrid automata, one first needs a symbolic representation system that is expressive enough for the sets of values produced by unbounded loop iterations, as well as a formalism for describing the data transformations labeling control paths. The main problems are then to decide whether the effect of unbounded iterations of such a path can be computed over symbolically represented sets, and to carry out this computation.

Solutions to these problems have been proposed in earlier work: Sets of reachable data values can be expressed in the first-order logic $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$, which generalizes Presburger arithmetic to mixed integer and real variables, and for which usable data structures have been developed [7]. The transformations undergone by variables along control paths of linear hybrid automata[3] correspond to *Linear Hybrid Relations (LHR)*, the acceleration of which is studied in [5, 6].

The cycle acceleration method proposed in [5] is able to handle a broad class of LHR, in particular all *Multiple Counters Systems (MCS)* [11]. This subclass of LHR is relevant in practice since it has been established that accelerating arbitrary control paths of *timed automata* [1], reduces to the same problem over MCS. It is actually proved in [5] that acceleration of MCS makes it possible to compute symbolically the reachability set of timed automata with a guarantee of termination.

The results of [5] nevertheless suffer from two weaknesses. First, when this acceleration method is applied to purely integer transformations, which can be seen as a particular case of LHR, it is not able to handle all instances covered by an acceleration procedure that has been specifically developed for such transformations [2, 3]. Second, the method is sensitive to the coordinate system used for expressing data values. For instance, even though all MCS can be accelerated, the same property does not hold for LHR obtained after applying linear variable change operations to MCS.

[3] The results of [5, 6] actually consider the slightly smaller class of *strongly linear* hybrid automata but their extension to linear hybrid automata is immediate.

The goal of this work is to broaden substantially the scope of cycle acceleration of linear hybrid relations, by developing a new approach that does not have these weaknesses. For purely integer transformations, an obvious solution would be to detect whether the considered LHR belongs to this class, and then branch to a specific acceleration algorithm. This approach would not improve the state of the art, and we propose instead a solution that is not only able to handle all integer transformations that can be accelerated by the specialized algorithm of [3], but also combinations of such discrete transformations with simple continuous ones. After studying the properties of this solution, we then generalize it into a method that becomes powerful enough for handling all transformations extracted from MCS, as well as their transformations by arbitrary linear variable change operations.

2 Preliminaries

2.1 Algebra Basics

A *linear constraint* over variables $\boldsymbol{x} \in \mathbb{R}^n$, with $n \geq 0$, is a constraint of the form $\boldsymbol{a}.\boldsymbol{x}\#b$, with $\boldsymbol{a} \in \mathbb{Q}^n$, $b \in \mathbb{Q}$ and $\# \in \{<, \leq, =, \geq, >\}$. This constraint is *strict* if $\# \in \{<, >\}$, and *non-strict* otherwise. It is an *inequality* constraint if $\# \in \{<, \leq, \geq, >\}$, and an *equality* constraint otherwise. A constraint $\boldsymbol{a}.\boldsymbol{x}\#b$ is said to be *saturated* by a value $\boldsymbol{v} \in \mathbb{R}^n$ if this value satisfies $\boldsymbol{a}.\boldsymbol{v} = b$.

The set of points $\boldsymbol{x} \in \mathbb{R}^n$ that satisfy a given finite conjunction of equality constraints forms an *affine space*. An affine space $S \subseteq \mathbb{R}^n$ can be expressed in the form $S = A\,\mathbb{R}^m + \boldsymbol{b}$, where $0 \leq m \leq n$, $A \in \mathbb{Q}^{n \times m}$ is a matrix with rank m, and $\boldsymbol{b} \in \mathbb{Q}^n$. The value m then corresponds to the *dimension* of S. The affine space of smallest dimension that contains a given set is unique, and known as the *affine hull* of this set.

The set of solutions of a finite conjunction of linear constraints forms a *convex polyhedron*, the dimension of which is defined as the dimension of its affine hull. Within $\mathbb{R}^n$, a convex polyhedron of dimension n can be represented by a finite *canonical* conjunction of constraints, i.e., a set of constraints that is uniquely determined by the polyhedron. For each constraint in this set, there exists at least one point that saturates this constraint, and that satisfies all the other ones without saturating them. Convex polyhedra of dimension $m < n$ can be expressed as $A\,\Pi + \boldsymbol{b}$, where $A \in \mathbb{Q}^{n \times m}$, $\boldsymbol{b} \in \mathbb{Q}^n$, and $\Pi \subseteq \mathbb{R}^m$ is a polyhedron of dimension m that is represented canonically. In order to simplify notations, we sometimes denote a set $\{\boldsymbol{v}\}$ as $\boldsymbol{v}$, and write $S_1 + S_2$ to mean $\{\boldsymbol{v_1} + \boldsymbol{v_2} \mid \boldsymbol{v_1} \in S_1 \wedge \boldsymbol{v_2} \in S_2\}$.

2.2 Linear Hybrid Relations

A *Linear Hybrid Automaton (LHA)* is composed of a finite control graph extended with a given number n of variables $x_1, x_2, \ldots, x_n$ that take their values in $\mathbb{R}$. These variables can be grouped into a vector $\boldsymbol{x}$ whose domain is $\mathbb{R}^n$. We

refer the reader to [5, 6, 14] for further details and formal definitions. An example is given in Figure 2.

A *configuration* of a LHA is a pair $(\ell, \boldsymbol{v})$ where ℓ is a control location and $\boldsymbol{v}$ assigns a value to each variable. The current configuration can change in two ways. The first one *(time step)* is to let time elapse, in which case the control location remains constant, and the variable values evolve according to the *invariant* and *evolution law* of this location. Those are expressed as linear constraints over respectively the variable values, and their first time derivative. The second mechanism *(transition step)* is to follow a transition, which moves the control location and applies a discrete transformation to the variable values. This transformation is defined by linear constraints involving the initial and final values of the variables, taken across the transition.

The semantics of LHA is defined as follows. A configuration c_2 is reachable from a configuration c_1 if there exists a finite sequence of time and transition steps that leads from c_1 to c_2. A reachable configuration is one that is reachable from a designated initial set.

It has been shown in [6] that every finite control path of a LHA induces a transformation over its variables that can be characterized as follows.

Definition 1. *A* Linear Hybrid Relation (LHR) *is a relation*

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

where $P \in \mathbb{Z}^{m \times 2n}$, $\boldsymbol{q} \in \mathbb{Z}^m$, $\preceq \in \{<, \leq\}^m$, *and* $m \geq 0$.

We write $\theta = (P, \boldsymbol{q}, \preceq)$ to denote a relation of this form. Given a path in a LHA moving from a location ℓ to a location ℓ', one can compute P and $\boldsymbol{q}$ such that two values $\boldsymbol{v}, \boldsymbol{v'} \in \mathbb{R}^n$ satisfy the LHR $(P, \boldsymbol{q}, \preceq)$ iff $(\ell', \boldsymbol{v'})$ is reachable from $(\ell, \boldsymbol{v})$ by following the time and transition steps corresponding to this path.

In this work, for the sake of simplicity, we assume that all inequality constraints that appear in LHR are non-strict, i.e., that $\preceq$ stands for $\leq^m$, and that LHR are characterized by their pair $(P, \boldsymbol{q})$. All results in this paper can straightforwardly be extended to the more general setting of mixed strict and non-strict constraints.

Let θ be a LHR. Following [5], we call a constraint of this LHR *static* if it involves only either $\boldsymbol{x}$ or $\boldsymbol{x'}$. For a set $S \subseteq \mathbb{R}^n$, its *image* $\theta(S)$ by θ is given by $\{\boldsymbol{x'} \in \mathbb{R}^n \mid \exists \boldsymbol{x} \in S : (\boldsymbol{x}, \boldsymbol{x'}) \in \theta\}$. This can alternatively be expressed as $\theta(S) = (\theta \cap (S \times \mathbb{R}^n))|_{[n+1,2n]}$, where $U|_I$ denotes the *projection* of the elements of U onto the vector components belonging to I. Given two LHR θ_1 and θ_2, their *composition* $\theta_2 \circ \theta_1$ is the LHR θ such that $\theta(S) = \theta_2(\theta_1(S))$ for all sets S. Note that we have $\theta_2 \circ \theta_1 = ((\theta_1 \times \mathbb{R}^n) \cap (\mathbb{R}^n \times \theta_2))|_{[1,n] \cup [2n+1,3n]}$. Finally, for every k, the result of composing $k-1$ times a LHR θ with itself is denoted θ^k, with θ^0 corresponding to the identity relation.

2.3 Representation of Convex Polyhedra

In the following sections, we study the effect and repeated effect of LHR on sets. The image $\theta(\boldsymbol{v})$ of a point $\boldsymbol{v} \in \mathbb{R}^n$ by a LHR θ is the set of points $\boldsymbol{v'}$ such that $(\boldsymbol{v}, \boldsymbol{v'})$ satisfies the linear constraints of θ, that is, a convex polyhedron. We now study some topological properties of such polyhedra.

Following the discussion in Section 2.1, we consider w.l.o.g. a convex polyhedron $\Pi \subseteq \mathbb{R}^n$ of dimension n, defined by its canonical set of inequality constraints. As explained in [4, 13], such a polyhedron induces a finite equivalence relation $\sim_\Pi$ on the points of $\mathbb{R}^n$: One has $\boldsymbol{v} \sim_\Pi \boldsymbol{v'}$ iff these two points saturate identical subsets of constraints of Π. The equivalence classes of $\sim_\Pi$ correspond to the *geometrical components* of Π. For each geometrical component C, its affine hull $\mathrm{aff}(C)$ matches the constraints of Π saturated by C, and its *dimension* is defined as the one of this affine hull. The geometrical components of Π are linked together by an *incidence* partial order $\prec_\Pi$: One has $C_1 \prec_\Pi C_2$ iff $\mathrm{aff}(C_1) \subset \mathrm{aff}(C_2)$, i.e., iff the constraints saturated in C_1 are a superset of those saturated in C_2.

Those properties lead to a data structure for representing symbolically convex polyhedra: A *Convex Polyhedron Decision Diagram (CPDD)* representing a polyhedron Π is a directed acyclic graph in which:

- The nodes correspond to the geometrical components of Π, and are labeled by the constraints of Π that they saturate, written as equalities (in other words, by the affine hull of their geometrical component).
- If Π admits a unique minimal component with respect to the incidence order $\prec_\Pi$, then the node q_0 associated to this component is marked as initial. Otherwise, the initial node q_0 is an additional special node in which all constraints are considered to be saturated (yielding an empty affine hull).
- The edges follow the incidence relation, removing those that are redundant by transitivity. An edge from q_1 to q_2 is labeled by the constraints that are saturated in q_1 and not in q_2, written as strict inequalities.

An example of CPDD is given in Figure 1. This data structure actually provides a simple procedure for locating the geometrical component of Π to which a given point $\boldsymbol{v} \in \mathbb{R}^n$ belongs: Starting from the initial node, one follows edges labeled by inequality constraints that are satisfied by $\boldsymbol{v}$. The procedure ends upon reaching a node labeled by equality constraints satisfied by $\boldsymbol{v}$, which then represents the component to which $\boldsymbol{v}$ belongs. If several paths can be followed from a given node, one of them can be chosen arbitrarily without the need for backtracking.

This procedure illustrates an essential property of convex polyhedra: The points contained in a geometrical component are exactly those that saturate the constraints associated to this component, and that do not saturate the other constraints. This property will be exploited in order to establish a key result in Section 4.

It is worth mentioning that CPDD nodes do not correspond to all possible combinations of saturated linear constraints, but only to those that are associated to geometrical components. For instance, the CPDD depicted in Figure 1

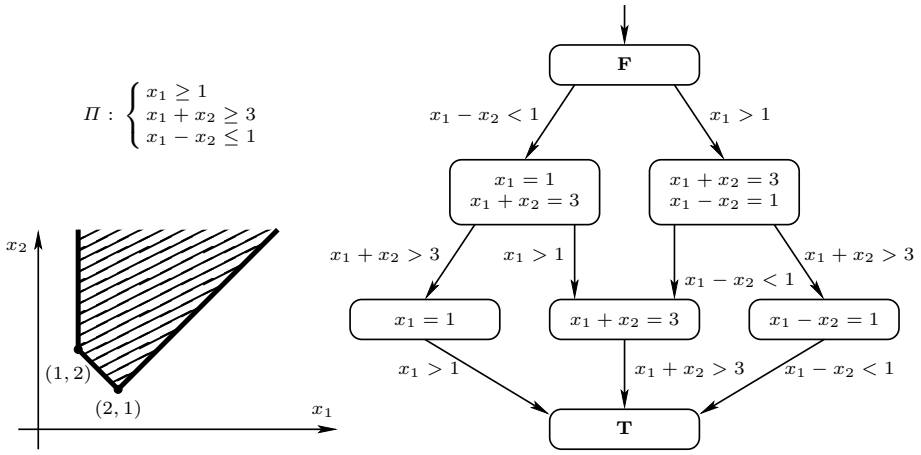

Fig. 1. Example of Convex Polyhedron Decision Diagram.

does not have a node corresponding to the set of constraints $\{x_1 \geq 1, x_1 - x_2 \leq 1\}$, since these constraints cannot be saturated while simultaneously satisfying $x_1 + x_2 \geq 3$.

Algorithms are available for building and manipulating polyhedra represented by CPDD, in particular for computing their canonical form (which is unique up to isomorphism), as well as their intersection and projection. This data structure has been generalized to non-convex polyhedra in [4, 13].

2.4 Cycle Acceleration

The *cycle acceleration* problem consists in checking, within a symbolic representation system, whether the image of any representable set by unbounded iterations of a given data transformation is representable as well. In such a case, this transformation is said to be *iterable* [2]. One also needs an algorithm for computing symbolically the image of represented sets by iterable transformations. This decision does not have to be precise: a sufficient criterion can be used provided that it handles practically relevant transformations.

In the next section, we recall two iterability criteria, one developed for linear transformations over integer variables and one for linear hybrid relations, and show that they can be combined into a criterion that has a broader scope.

3 Affine Hybrid Transformations

3.1 Discrete and Hybrid Periodic Transformations

Over the domain $\mathbb{Z}^n$, it has been established that transformations of the form $\boldsymbol{x} \mapsto A\boldsymbol{x} + \boldsymbol{b}$, with $A \in \mathbb{Z}^{n \times n}$ and $\boldsymbol{b} \in \mathbb{Z}^n$, are iterable within Presburger

arithmetic, i.e., the first-order theory $\langle \mathbb{Z}, +, < \rangle$, iff there exists $p \in \mathbb{N}_{>0}$ such that $A^{2p} = A^p$. This criterion can be decided using only integer arithmetic, and a suitable value of p can be computed whenever one exists [2, 3].

Transformations θ that satisfy this criterion have an *ultimately periodic* behavior: For every $\boldsymbol{v} \in \mathbb{Z}^n$, the sequence $\theta^p(\boldsymbol{v})$, $\theta^{2p}(\boldsymbol{v})$, $\theta^{3p}(\boldsymbol{v})$, ... is such that $\theta^{(k+1)p}(\boldsymbol{v}) = \theta^{kp}(\boldsymbol{v}) + \boldsymbol{\delta}$ for all $k > 0$, where $\boldsymbol{\delta} \in \mathbb{Z}^n$ is a constant increment vector. It is also known that adding a linear guard $P\boldsymbol{x} \leq \boldsymbol{q}$, with $P \in \mathbb{Z}^{m \times n}$, $\boldsymbol{q} \in \mathbb{Z}^m$ and $m \geq 0$, to an iterable transformation produces one that is iterable as well.

Hybrid transformations can also show a periodic behavior. It has been proved in [6] that LHR θ over $\mathbb{R}^n$ in which all constraints have the form $\boldsymbol{p}.(\boldsymbol{x'} - \boldsymbol{x}) \leq q$, with $\boldsymbol{p} \in \mathbb{Z}^n$ and $q \in \mathbb{Z}$ have this property: For every $\boldsymbol{v} \in \mathbb{R}^n$, the sequence $\theta(\boldsymbol{v})$, $\theta^2(\boldsymbol{v})$, $\theta^3(\boldsymbol{v})$, ... is such that $\theta^{k+1}(\boldsymbol{v}) = \theta^k(\boldsymbol{v}) + \Delta$ for all $k > 0$, where $\Delta \subseteq \mathbb{R}^n$ is an increment that now takes the form of a constant convex polyhedron.

A natural idea is therefore to study hybrid transformations that have a periodic behavior, but with a period that may be greater than one. The following definition generalizes linear integer transformations to the hybrid case.

Definition 2. *An* Affine Hybrid Transformation (AHT) *is a LHR $\theta \subseteq \mathbb{R}^n \times \mathbb{R}^n$ such that for every $\boldsymbol{x} \in \mathbb{R}^n$,*

$$\theta(\boldsymbol{x}) = A\boldsymbol{x} + \Pi,$$

where $A \in \mathbb{Q}^{n \times n}$, and $\Pi \subseteq \mathbb{R}^n$ is a convex polyhedron.

The iterability criterion obtained for linear integer transformations straightforwardly extends to AHT.

Theorem 3. *Let θ be an AHT $\boldsymbol{x} \mapsto A\boldsymbol{x} + \Pi$, with $A \in \mathbb{Q}^{n \times n}$. If A is such that $A^{2p} = A^p$ for some $p \in \mathbb{N}_{>0}$, then θ is iterable within $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$. Moreover, adding static constraints to an iterable AHT that satisfies this property produces a LHR that is iterable as well.*

Proof sketch. For every $\boldsymbol{v} \in \mathbb{R}^n$ and $k > 1$, one has $\theta^{kp}(\boldsymbol{v}) = A^{kp}\boldsymbol{v} + \sum_{i=0}^{kp-1} A^i \Pi$. If $A^{2p} = A^p$, this simplifies into $\theta^{kp}(\boldsymbol{v}) = A^p\boldsymbol{v} + \sum_{i=0}^{2p-1} A^i \Pi + (k-2) \sum_{i=p}^{2p-1} A^i \Pi$. Using the mechanisms introduced in [6], this leads to a formula of $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ defining $\theta^k(\boldsymbol{v})$ for all $k \geq 0$ in terms of $\boldsymbol{v}$ and k. □

In order to be able to exploit the acceleration of AHT during symbolic state-space exploration of linear hybrid automata, two problems need to be solved:

- Given a LHR expressed as a conjunction of linear constraints, deciding whether it is equivalent to an AHT and, in the positive case, computing the corresponding matrix A.
- Deciding whether a matrix $A \in \mathbb{Q}^{n \times n}$ is such that $A^{2p} = A^p$ for some $p \in \mathbb{N}_{>0}$, and computing such a value p.

The former problem is addressed in Section 3.2. The latter can be solved by adapting a result from [2, 3]:

Theorem 4. *A matrix $A \in \mathbb{Q}^{n\times n}$ is such that $A^{2p} = A^p$ for some $p \in \mathbb{N}_{>0}$ if and only if A^p is diagonalizable and has eigenvalues that belong to $\{0, 1\}$. There exists an algorithm for deciding this criterion and computing a suitable value of p, using only integer arithmetic.*

Proof sketch. This result is established in [2, 3] for matrices with integer components, the idea being to check whether they admit a characteristic polynomial that can be decomposed into a product of cyclotomic polynomials. The method proposed in [2, 3] for performing this operation also applies to rational matrices. □

3.2 Detecting Affine Hybrid Transformations

We now address the problems of deciding whether a LHR is affine, that is, whether $\theta = (P, \boldsymbol{q})$ is equivalent to some AHT $\boldsymbol{x} \mapsto A\boldsymbol{x} + \Pi$, and of computing the corresponding matrix A and convex polyhedron Π.

When θ is affine, the image of a set $S \subseteq \mathbb{R}^n$ is obtained by first applying to each point in S a transformation $\boldsymbol{x} \mapsto A\boldsymbol{x}$, where $A \in \mathbb{Q}^{n\times n}$ is identical for each point, and then adding a constant convex polyhedron Π to the result.

Let us assume that this polyhedron has at least one vertex, i.e., a geometrical component of dimension 0. We can actually make this assumption without loss of generality, since it follows from [5] that if an affine transformation θ does not satisfy this property, then its acceleration can be reduced to that of a LHR of smaller dimension.

The image by θ of an arbitrary point $\boldsymbol{x} \in \mathbb{R}^n$ is the polyhedron $A\boldsymbol{x}+\Pi$, which corresponds to Π translated by the vector $A\boldsymbol{x}$. Consider a particular vertex $\boldsymbol{v_i}$ of this polyhedron, in other words, a point that is the only one to saturate some given subset of its constraints. The vertex $\boldsymbol{v_i}$ is the translation of a vertex $\boldsymbol{b_i}$ of Π by the vector $A\boldsymbol{x}$, that is, $\boldsymbol{v_i} = A\boldsymbol{x} + \boldsymbol{b_i}$. The same reasoning applied to other vertices will yield the same matrix A.

Recall that the constraints defining θ are expressed over the variables $\boldsymbol{x}$ and $\boldsymbol{x'}$, the value of which is respectively considered before and after applying the transformation. A transformation of the form $\boldsymbol{x} \mapsto A\boldsymbol{x} + \boldsymbol{b}_i$ thus corresponds to the saturated form $\boldsymbol{x'} = A\boldsymbol{x} + \boldsymbol{b}_i$ of some constraints of θ. Since this set of saturated constraints is satisfiable, an important observation is that it must correspond to a geometrical component of the convex polyhedron $\Theta \subseteq \mathbb{R}^{2n}$ defined by the constraints of θ. In other words, there must exist in this polyhedron a geometrical component C_i that has an affine hull equal to $\boldsymbol{x'} = A\boldsymbol{x} + \boldsymbol{b}_i$.

Since we have considered the vertices of Π, which are its geometrical components of smallest dimension, the components C_i with this property must correspond to the minimal non-empty components of Θ. We thus have the following result.

Theorem 5. *There exists a procedure for deciding whether a LHR $\theta \subseteq \mathbb{R}^n \times \mathbb{R}^n$ is an Affine Hybrid Transformation.*

Proof sketch. A simple strategy for deciding whether θ is affine consists in inspecting the minimal non-empty geometrical components in a symbolic representation of Θ. The following procedure can be used:

1. Build a CPDD representing Θ.
2. Select one of its minimal non-empty components.
3. Extract a matrix A from the affine hull of this component.
4. Compute $\Pi = \theta(\mathbf{0})$.
5. Check whether θ is equivalent to $\boldsymbol{x} \mapsto A\boldsymbol{x} + \Pi$, by comparing Θ with the polyhedra induced by the corresponding sets of constraints.

If the polyhedron Π satisfies our initial hypothesis of having at least one vertex, then performing Step 3 simply amounts to checking that the considered affine hull is defined by constraints of the form $\boldsymbol{x'} = A\boldsymbol{x} + \boldsymbol{b}_i$, and then syntactically extracting A from these constraints. Otherwise, if Π does not have vertices, this operation can still be performed but after first applying to the affine hull constraints the *rank* and *subspace* reductions of [5]. Those correspond intuitively to applying a linear coordinate transformation that results in constraints expressed in terms of the smallest possible number of independent variables. More precisely, the rank reductions amount to performing the following operations. First, the set of constraints is rewritten in the form $P_2\boldsymbol{x'} = P_1\boldsymbol{x} + \boldsymbol{q}$, where $P_1, P_2 \in \mathbb{Z}^{m\times n}$, $\boldsymbol{q} \in \mathbb{Z}^m$ and $m \geq 0$. If the rank r of P_1 is less than n, then a linear variable change operation is applied in order to express the transformation in terms of only r distinct variables. The same procedure is also carried out if the rank of P_2 is less than n. In addition, subspace reductions are applied when the set of constraints $P_2\boldsymbol{x'} = P_1\boldsymbol{x} + \boldsymbol{q}$ implies static constraints on either $\boldsymbol{x}$ or $\boldsymbol{x'}$. The reduction consists in performing a linear variable change operation onto the largest number of distinct variables that are not statically constrained. □

This procedure is illustrated in Section 5.1. In practice, since it follows from Theorem 3 that static constraints do not hamper iterability, a good strategy is to remove them before checking whether a LHR is affine. Finally, note that the acceleration method for AHT discussed in this section is able to successfully process all linear integer transformations that are handled by [2, 3].

4 Generalized Affine Transformations

4.1 Principles

Affine hybrid transformations θ have the property that we can compute from their set of constraints a value $p \in \mathbb{N}_{>0}$ such that θ^p has an ultimately periodic behavior. In other words, iterating θ reduces to iterating θ^p, which is feasible within additive arithmetic. We call such a value p a *period* of θ.

In Section 3, we have shown that such a period p can be obtained by inspecting matrices extracted from the minimal geometrical components of the polyhedron $\Theta \subseteq \mathbb{R}^{2n}$ induced by the constraints of θ. If θ is affine, then these matrices happen to be identical for all components, which represents the fact

that they are similarly affected by θ^p, in the sense that they share the same periodic behavior.

This sufficient condition for iterability is not at all necessary: If the geometrical components of Θ correspond to matrices A_1, A_2, ... that are not identical, but yield values p_1, p_2, ... such that $A_i^{2p_i} = A_i^{p_i}$ for all i, then all those components share an ultimately periodic behavior of period $p = \mathrm{lcm}(p_1, p_2, \ldots)$. A possible acceleration procedure thus consists in computing such a value p by inspecting the geometrical components of Θ, computing p as the least common multiple of their detected periodicities p_i, and then checking whether θ^p reduces to a periodic transformation that is iterable within $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$. This inspection does not necessarily have to be carried out for all geometrical components: The iterability of θ^p can be checked whenever a candidate value for p has been obtained. If the analysis of a geometrical component fails to produce a periodicity p_i, the procedure can nevertheless continue with the other components.

This approach shares similarities with the solution proposed in [5] for accelerating *Multiple Counters Systems (MCS)* [11], which are a subclass of LHR in which all constraints are of the form $z_i \# z_j + c$, with $z_i, z_j \in \{x_1, \ldots, x_n, x'_1, \ldots, x'_n\}$, $\# \in \{<, \leq, =, \geq, >\}$, and $c \in \mathbb{Z}$. This solution proceeds by building directed weighted graphs that represent the set of constraints of a MCS θ, and then measuring the weights p_1, p_2, ... of the simple cycles in these graphs. The value $p = \mathrm{lcm}(p_1, p_2, \ldots)$ provides a (non necessarily optimal) candidate for the periodicity of θ. It is shown in [5] that this technique is able to accelerate every MCS.

In Section 4.2, we establish a connection between the acceleration technique presented in this paper and the one proposed for MCS in [5], by showing that the periodicities that are captured by the graph analysis method can also be detected by the inspection of geometrical components. As a consequence, our technique is complete over MCS. Compared with the method of [5], it has the important advantage of being closed under linear variable change operations, since those do not affect the properties of geometrical components of polyhedra. Furthermore, our approach is not limited to handling MCS, unlike the acceleration method developed in [11].

After a candidate periodicity value p has been obtained by inspecting the geometrical components of Θ, it remains to check whether the transformation θ^p has a periodic behavior that can be captured within $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$. This problem is addressed in Section 4.3.

4.2 Multiple Counters Systems

Let us briefly describe the method introduced in [5] for computing the periodicity of a MCS θ. As discussed in Section 2.2, for the sake of clarity, we consider that all inequality constraints are non-strict.

The first step is to build a finite directed graph G_θ, in which the nodes correspond to the variables $x_1, x_2, \ldots, x_n$, and the edges $(x_i, (c, d), x_j)$ are labeled with a *cost* $c \in \mathbb{Z}$ and a *depth* $d \in \{-1, 0, 1\}$. This graph represents the constraints of θ:

– A constraint $x_j \leq x_i + c$ or $x'_j \leq x'_i + c$ is represented by an edge $(x_i, (c, 0), x_j)$.
– A constraint $x'_j \leq x_i + c$ is represented by an edge $(x_i, (c, 1), x_j)$.
– A constraint $x'_j \geq x_i + c$ is represented by an edge $(x_j, (-c, -1), x_i)$.

The paths of G_θ correspond to combinations of constraints of θ. The cost and depth of such a path σ are defined as the sum of the individual cost and depth of the edges that compose it. For every $k > 0$, a path σ of depth k in G_θ represents a constraint $x'_j \leq x_i + c$ of the transformation θ^k, where x_i and x_j are respectively the origin and destination nodes of σ, c is the cost of σ, and the intermediate depths reached at each node visited by σ remain in the interval $[0, k]$. In the same way, the paths of G_θ of depth $-k$ or 0 also correspond to constraints of θ^k.

The main result of [5] is to show that, in order to obtain all constraints of θ^k, it is sufficient to consider the paths of G_θ of suitable depth that contain only unbounded occurrences of a single simple cycle. A periodicity p of θ, i.e., a value such that θ^p reduces to a periodic transformation, is then obtained by computing the least common multiple of the depths of the simple cycles of G_θ. This periodicity may not be the smallest one for θ, but this is not problematic.

We are now going to establish that such a periodicity p can also be computed by the procedure outlined in Section 4.1. This property is a consequence of the following result.

Theorem 6. *Let $k > 0$, and σ be a simple cycle of G_θ of depth $\pm k$ and cost c, representing a constraint $x'_i \leq x_i + c$ or $x'_i \geq x_i - c$ of θ^k. If this constraint can be saturated*[4] *by values of $\boldsymbol{x}$ and $\boldsymbol{x'}$ that satisfy $(\boldsymbol{x}, \boldsymbol{x'}) \in \theta^k$, then there exists a geometrical component of Θ producing a matrix $A \in \mathbb{Q}^{n \times n}$ such that $A^{2k} = A^k$.*

Proof sketch. Let S be the set of constraints of θ that are represented by the edges of G_θ composing σ. Since the constraint represented by σ can be saturated, there exist values $\boldsymbol{v}, \boldsymbol{v'} \in \mathbb{R}^n$ that can respectively be assigned to $\boldsymbol{x}$ and $\boldsymbol{x'}$ in order to saturate all constraints in S.

The values $\boldsymbol{v}$ and $\boldsymbol{v'}$ may also saturate other constraints of θ. Let S' denote the set of constraints of θ that are necessarily saturated when S is saturated, i.e., that are saturated by every $\boldsymbol{v}$ and $\boldsymbol{v'}$ that saturate S. The set S' contains only constraints that are either saturated for all $\boldsymbol{v}, \boldsymbol{v'} \in \mathbb{R}^n$, or correspond to one or several simple cycles of G_θ. In the latter case, it can be established that each of these cycles shares the same depth $\pm k$ as σ.

One can thus find values $\boldsymbol{v}$ and $\boldsymbol{v'}$ that saturate all constraints in $S \cup S'$, and do not saturate the other constraints of θ. From the discussion in Section 2.3, it follows that the point $(\boldsymbol{v}, \boldsymbol{v'}) \in \mathbb{R}^{2n}$ belongs to a geometrical component of Θ with an affine hull that exactly corresponds to the solutions of $S \cup S'$.

The matrix A produced by this component using the procedure described in Section 3.2 has the following property. Let $X \subseteq \{x_1, \ldots, x_n\}$ denote the set of all variables visited by the simple cycles of G_θ that correspond to the constraints

[4] This saturation requirement intuitively expresses the property that the constraint is essential, i.e., that it is not implied by other constraints of θ^k.

in $S \cup S'$. Recall that these simple cycles are all of depth $\pm k$. It follows that the transformation $\boldsymbol{x} \mapsto A^k \boldsymbol{x}$ preserves the values of the variables in X and assigns the value $\mathbf{0}$ to the other variables. One thus has $A^{2k} = A^k$. □

In [5], a candidate value for the periodicity p of θ is obtained by computing the least common multiple of the depths p_i of all simple cycles in G_θ. Theorem 6 shows that each such value p_i will also be computed by the procedure discussed in Section 4.1, provided that the underlying cycle represents a constraint that is not redundant. The reciprocal property does not hold: Some geometrical components of Θ may correspond to a set of saturated constraints of θ that does not form a cycle. The inspection of such components may produce matrices A that do not yield a periodicity p_i, or yield a spurious one. This is not problematic, since a transformation θ such that θ^p has a periodic behavior is also periodic when it is raised to a power equal to an integer multiple of p.

4.3 Checking Periodicity

We now investigate the possibility of validating a candidate periodicity $p \in \mathbb{N}_{>0}$ for a LHR θ, i.e., checking whether θ^p has a periodic behavior that can be accelerated. Note that, for every $j \in [0, p-1]$ and $k \geq 0$, one has $\theta^{j+kp} = (\theta^p)^k \circ \theta^j$, hence accelerating θ reduces to accelerating θ^p.

Let θ' be the LHR defined by the *periodic* constraints of θ^p, i.e., those of the form $\boldsymbol{p}.(\boldsymbol{x'} - \boldsymbol{x}) \# q$, with $\boldsymbol{p} \in \mathbb{Z}^n$, $q \in \mathbb{Z}$, and $\# \in \{\leq, =, \geq\}$. Following [6], one can obtain a formula of $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ representing the relation $\boldsymbol{x'} \in (\theta')^k(\boldsymbol{x})$ for all $k \geq 0$ in terms of the variables $\boldsymbol{x}$, $\boldsymbol{x'}$, and k. The problem is thus to check whether the acceleration of θ (or, equivalently, θ^p) can be reduced to the acceleration of θ'.

We first consider the case of a MCS θ for which we have obtained a period p by applying either the method introduced in Section 4.1, or the one given in [5]. For any $k \geq 0$, we know that the constraints of θ^{kp} are represented by paths of depth 0, k or $-k$ in the graph G_{θ^p}. It has been shown in [5] that it is sufficient to consider the paths of this graph that are either acyclic, or contain repetitions of only a single cycle of length 1. Such cycles correspond to periodic constraints, which are captured in θ'.

The transformation θ^p therefore satisfies two properties. The first one states that there exists $m > 0$ such that $m \leq n$, and every composition of m constraints of θ^p that results in a constraint of θ^{mp} necessarily includes at least one periodic constraint from θ'. Formally, this condition can be expressed as

$$\theta^{mp} = \bigcap_{i+j=m-1} \left[\theta^{ip} \circ \theta' \circ \theta^{jp}\right]. \tag{1}$$

The second property states that, in compositions of constraints of θ^p, periodic constraints do not need to be repeated at more than one place. Formally, we have

$$\forall i < m : \left[\theta' \circ \theta^{ip} \circ \theta'\right] \supseteq \left[(\theta')^2 \circ \theta^{ip}\right] \cap \left[\theta^{ip} \circ (\theta')^2\right]. \tag{2}$$

In the case of MCS, Conditions 1 and 2 are always satisfied. For more general LHR θ, they can be used as a sufficient criterion for validating a candidate value p for the periodicity of θ. This is illustrated in Section 5.1 below. In practical applications, these conditions can be decided by operations over CPDD representations of the transformations, as discussed in Section 2.3.

The last step is to show that a LHR θ that satisfies Conditions 1 and 2 can be accelerated. These conditions imply that for all $k \geq m$, we have

$$\theta^{kp} = \bigcap_{i+j=m-1} \left[\theta^{ip} \circ (\theta')^{k-i-j} \circ \theta^{jp}\right].$$

Since θ' can be accelerated, this expression can be turned into a formula of $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$ representing the relation $\boldsymbol{x'} \in (\theta')^k(\boldsymbol{x})$ in terms of $\boldsymbol{x}$, $\boldsymbol{x'}$, and k.

5 Examples

5.1 Periodic LHR

Let us illustrate the approach proposed in this paper on the LHR $\theta \subseteq \mathbb{R}^2 \times \mathbb{R}^2$ defined by the set of constraints

$$\theta : \begin{cases} x_2' + x_1 \leq -1 \\ x_2' - x_1' + x_2 \leq -1 \\ 2x_2' - x_1' + x_1 + x_2 \geq -4. \end{cases}$$

First step: extracting a candidate periodic matrix A from θ. The convex polyhedron $\Theta \subseteq \mathbb{R}^4$ induced by these constraints admits three minimal non-empty geometrical components, with the corresponding affine hulls

$$\alpha_1 : \begin{cases} x_2' + x_1 = -1 \\ x_2' - x_1' + x_2 = -1, \end{cases}$$

$$\alpha_2 : \begin{cases} x_2' + x_1 = -1 \\ 2x_2' - x_1' + x_1 + x_2 = -4, \end{cases}$$

and

$$\alpha_3 : \begin{cases} x_2' - x_1' + x_2 = -1 \\ 2x_2' - x_1' + x_1 + x_2 = -4. \end{cases}$$

The affine hull α_1 can equivalently be represented by the following constraints, from which we deduce the matrix A below:

$$\begin{cases} x_1' = -x_1 + x_2 \\ x_2' = -x_1 - 1 \end{cases} \qquad A = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}$$

Note that the affine hulls α_2 and α_3 produce the same matrix A, which hints at the property that θ is affine.

Using the algorithm mentioned in Theorem 4, one obtains that this matrix is such that $A^6 = A^3$ (actually, it satisfies the stronger property $A^3 = I_2$, where I_2 denotes the identity matrix of size 2), which gives a candidate periodicity $p = 3$.

Second step: checking whether θ is affine. Following the procedure given in Section 3.2, we can compute a polyhedron Π such that θ is equivalent to $\boldsymbol{x} \mapsto A\boldsymbol{x} + \Pi$. This yields:

$$\Pi : \begin{cases} x_2' \leq -1 \\ x_2' - x_1' \leq -1 \\ 2x_2' - x_1' \geq -4. \end{cases}$$

From Theorem 3, we deduce that θ is iterable within $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$.

Alternative second step: checking the candidate period. Alternatively, we may avoid computing Π and directly use the technique of Section 4.3 for checking that the candidate periodicity $p = 3$ is valid. We obtain that θ^3 is of the form:

$$\begin{cases} -4 \leq x_1' - x_1 \leq 4 \\ -4 \leq x_2' - x_2 \leq 4 \\ -4 \leq x_1' - x_2' - x_1 + x_2 \leq 4 \\ x_1' + x_2' - x_1 - x_2 \leq 6 \\ x_1' - 2x_2' - x_1 + 2x_2 \leq 6 \\ 2x_1' - x_2' - 2x_1 + x_2 \geq -6, \end{cases}$$

which is periodic since all its constraints are expressed over $x_1' - x_1$ and $x_2' - x_2$. For all $k > 1$, one thus has:

$$\theta^{3k} : \begin{cases} -4k \leq x_1' - x_1 \leq 4k \\ -4k \leq x_2' - x_2 \leq 4k \\ -4k \leq x_1' - x_2' - x_1 + x_2 \leq 4k \\ x_1' + x_2' - x_1 - x_2 \leq 6k \\ x_1' - 2x_2' - x_1 + 2x_2 \leq 6k \\ 2x_1' - x_2' - 2x_1 + x_2 \geq -6k. \end{cases}$$

The reflexive and transitive closure of θ^{3k} can be obtained by quantification over k. As a result, θ is iterable within $\langle \mathbb{R}, \mathbb{Z}, +, \leq \rangle$.

5.2 Linear Hybrid Automaton

As a second example, consider the linear hybrid automaton H in Figure 2. The effect of the cycle in H, starting from the leftmost location and preceding each transition by the passage of time, is described by the LHR θ_H below. The variable x has been eliminated using the reductions of [5] since, after the first iteration, the cycle starts and ends with $x = 0$.

$$\theta_H = \begin{cases} y + t - y' + t' \leq 1 \\ -2y + z - t + 2y' - z' - t' \leq -1 \\ y - y' \leq -10 \end{cases}$$

The convex polyhedron $\Theta_H \subseteq \mathbb{R}^6$ induced by θ_H has one minimal non-empty geometrical component, obtained by saturating all the constraints of θ_H. Its

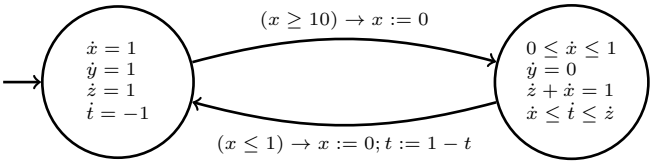

Fig. 2. Linear Hybrid Automaton H.

affine hull is described by the following constraints, from which we derive the matrix A_H.

$$\begin{cases} y' = y + 10 \\ z' = z + 10 \\ t' = -t + 11 \end{cases} \qquad A_H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Using the algorithm mentioned in Theorem 4, we get a candidate period $p = 2$ since $A_H^2 = I_3$ (the identity matrix of dimension 3). Following the approach of Section 4.3 confirms that θ_H^2 is periodic. Hence, $(\theta_H^2)^*$ can be computed using the techniques of [6]. One then obtains $\theta_H^* = (\theta_H^2)^* \circ (\theta_H \cup Id)$. Note that the computation of θ_H^* was out of scope of the techniques of [5, 6], which cannot handle periodicities greater than one.

6 Conclusions

This paper introduces an original method for accelerating the data transformations that label control cycles of linear hybrid automata. Given such a transformation θ, the idea consists in constructing a convex polyhedron from its linear constraints, and then inspecting the geometrical components of this polyhedron in order to compute a value p such that θ^p is periodic.

This method is able to accelerate all transformations that can be handled by the specialized algorithms developed in [3, 5, 6, 11], in particular *Multiple Counters Systems*, to which the reachability analysis of timed automata can be reduced. Compared with those solutions, our method has the advantage of being closed under linear changes of coordinates, which naturally do not affect the geometrical features of polyhedra. Our acceleration algorithm can also potentially be applied to the *octagonal* transformations studied in [8–10], and an open question is to establish whether it provides full coverage of such transformations.

We did not analyze the practical cost of our acceleration procedure, which actually depends on the implementation details of the symbolic data structure used for manipulating polyhedra, and on the heuristics employed for selecting the geometrical components to be inspected. In all our case studies, considering the minimal non-empty components for which a non-trivial matrix A can be extracted turned out to be sufficient, but we do not know whether this property holds in all cases.

Acknowledgment

The authors wish to thank Nicolas Legrand for his contribution to the study of the CPDD data structure.

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science 126(2), 183–235 (1994)
2. Boigelot, B.: Symbolic Methods for Exploring Infinite State Spaces. Ph.D. thesis, Université de Liège (1998)
3. Boigelot, B.: On iterating linear transformations over recognizable sets of integers. Theoretical Computer Science 309(1–3), 413–468 (2003)
4. Boigelot, B., Brusten, J., Degbomont, J.F.: Automata-based symbolic representations of polyhedra. In: Proc. LATA'12. Lecture Notes in Computer Science, vol. 7183, pp. 3–20. Springer (2012)
5. Boigelot, B., Herbreteau, F.: The power of hybrid acceleration. In: Proc. CAV'06. Lecture Notes in Computer Science, vol. 4144, pp. 438–451. Springer (2006)
6. Boigelot, B., Herbreteau, F., Jodogne, S.: Hybrid acceleration using real vector automata. In: Proc. CAV'03. Lecture Notes in Computer Science, vol. 2725, pp. 193–205. Springer (2003)
7. Boigelot, B., Jodogne, S., Wolper, P.: An effective decision procedure for linear arithmetic over the integers and reals. ACM Transactions on Computational Logic 6(3), 614–633 (2005)
8. Bozga, M., Gîrlea, C., Iosif, R.: Iterating octagons. In: Proc. TACAS'09. Lecture Notes in Computer Science, vol. 5505, pp. 337–351. Springer (2009)
9. Bozga, M., Iosif, R., Konecný, F.: Fast acceleration of ultimately periodic relations. In: Proc. CAV'10. Lecture Notes in Computer Science, vol. 6174, pp. 227–242. Springer (2010)
10. Bozga, M., Iosif, R., Konecný, F.: Safety problems are NP-complete for flat integer programs with octagonal loops. In: Proc. VMCAI'14. Lecture Notes in Computer Science, vol. 8318, pp. 242–261. Springer (2014)
11. Comon, H., Jurski, Y.: Multiple counters automata, safety analysis and Presburger arithmetic. In: Proc. CAV'98. Lecture Notes in Computer Science, vol. 1427, pp. 268–279. Springer (1998)
12. Cousot, P., Cousot, R.: Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Proc. POPL'77. pp. 238–252. ACM Press (1977)
13. Degbomont, J.F.: Implicit Real-Vector Automata. Ph.D. thesis, Université de Liège (2013)
14. Henzinger, T.A.: The theory of hybrid automata. In: Proc. LICS'96. pp. 278–292. IEEE Computer Society Press (1996)
15. Zhou, C., Hoare, C. A. R., Ravn, A. P.: A calculus of durations. Information Processing Letters 40(5), 269–276 (1991)