Polyhedral Flows in Hybrid Automata*†

Rajeev Alur[1‡] Sampath Kannan[1§] Salvatore La Torre[2¶]

[1]University of Pennsylvania

[2]Università degli Studi di Salerno

alur,kannan @cis.upenn.edu, slatorre @unisa.it

Abstract

A *hybrid automaton* is a mathematical model for hybrid systems, which combines, in a single formalism, automaton transitions for capturing discrete updates with differential constraints for capturing continuous flows. Formal verification of hybrid automata relies on symbolic fixpoint computation procedures that manipulate sets of states. These procedures can be implemented using boolean combinations of linear constraints over system variables, equivalently, using polyhedra, for the subclass of *linear hybrid automata*. In a linear hybrid automaton, the flow at each control mode is given by a rate polytope that constrains the allowed values of the first derivatives. The key property of such a flow is that, given a state-set described by a polyhedron, the set of states that can be reached as time elapses, is also a polyhedron. We call such a flow a *polyhedral flow*. In this paper, we study if we can generalize the syntax of linear hybrid automata for describing flows without sacrificing the polyhedral property. In particular, we consider flows described by *origin-dependent rate polytopes*, in which the allowed rates depend, not only on the current control mode, but also on the specific state at which the mode was entered. We identify necessary and sufficient conditions for a class of flows described by origin-dependent rate polytopes to be polyhedral. We also propose and study additional classes of flows: *strongly polyhedral flows*, in which the set of states that can be reached *up to a given time* starting from a polyhedron is guaranteed to be a polyhedron, and *polyhedrally sliced flows*, in which the set of states that can be reached *at a given time* starting from a polyhedron is guaranteed to be a polyhedron. Finally, we discuss an application of the above classes of flows to approximate exponential behaviours.

*A preliminary version appears in *Hybrid Systems: Computation and Control, Proceedings of the Second International Workshop*, LNCS 1569, Springer Verlag, pages 5–18, 1999.

†This research was done while the third author was at the University of Pennsylvania.

‡Supported in part by ARO URI award DAAD19-01-1-0473, DARPA ITO Mobies award F33615-00-C-1707, DARPA/NASA grant NAG2-1214, and NSF awards CCR99-70925, CCR97-34115, and ITR/SY 0121431.

§Supported in part by the ARO grant DAAG55-98-1-0393 and the NSF award CCR-96-19910.

¶Supported in part by the MIUR grant project “Metodi Formali per la Sicurezza e il Tempo” (MEFISTO) and MIUR grant ex-60% 2002.

1 Introduction

A hybrid system consists of a collection of digital programs that interact with each other and with an analog environment. The formal modeling and analysis of the mixed digital-analog nature of hybrid systems requires a mathematical model that incorporates the discrete behavior of computer programs with the continuous behavior of environment variables, such as time, position, and temperature. The model of our choice is the *hybrid automaton* [ACH+95]. A hybrid automaton consists of a finite *control graph* whose vertices correspond to *control modes* and edges correspond to *control switches*. The automaton has a finite number of real-valued variables that can change discretely during the switches or continuously as the time elapses at control modes. The flow (i.e. continuous evolution) of the variables at a control mode is specified by differential equations and differential inequalities that constrain the allowed rates and by invariant predicates that constrain the allowed durations.

For analyzing hybrid systems, we build on the model-checking technology, in which a formal model of the system is checked, fully automatically, for correctness with respect to a requirement expressed in temporal logic (cf. [CE81, CK96, Hol97]). Model checking may require the exploration of the entire state space of the system. For discrete finite-state systems, this can be done enumeratively, by considering each state individually, or symbolically, by computing with constraints over boolean variables that encode state sets. Because of its ability to deal with very large state spaces, symbolic model checking has been proven an effective technique for debugging of complex hardware [BCD+92]. For hybrid systems, the state space is infinite, so an enumerative approach is impossible, but the symbolic approach can be extended to a class of hybrid automata called *linear hybrid automata* [AHH96]. A linear hybrid automaton requires that:

1. The predicates describing initialization of the system, the enabling of the control switches, the update of variables during switches, and the invariants at control modes, are boolean combinations of linear inequalities over automaton variables.

2. The flow at a control mode is given by a rate polytope that constrains the allowed values of the first derivatives of the automaton variables.

With these restrictions, it can be established that all the state-sets computed by the symbolic model checking procedures can be described by boolean combinations of linear inequalities. This theorem provides the foundations of the verifier HyTech [AHH96, HHW97], that implements the analysis procedures using packages to manipulate polyhedra. Even though termination of the model checking procedure is not guaranteed— the model checking problem for linear hybrid automata is undecidable, the method has been shown to be useful in analysis of protocols of practical interest (cf. [HW95]).

The dynamics allowed by linear hybrid automata are very restrictive, and hence, modeling in HyTech requires the user to approximate the system dynamics by identifying lower and upper bounds on the rates. Previous research has focused on identifying decidable subclasses [AD94, HKPV98, LPS00], and on general strategies for approximating the given dynamics using linear hybrid automata (cf. [HW96]).

In this paper, our objective is to identify classes of hybrid automata which are more general than linear hybrid automata, but still admit symbolic analysis using polyhedra.

Of the two restrictions imposed by linear hybrid automata, the first one seems natural and tight, given that we want only polyhedral sets. We investigate ways to generalize the second one in this paper. Given a set σ of states, let $post(\sigma)$ denote the set of states that can be reached starting from a state in σ as time elapses and variables evolve subject to the specified flow constraints. To be able to do reachability analysis using polyhedra, we want that *post* of a polyhedron must also be a polyhedron. We call such flows, which map polyhedra to polyhedra, *polyhedral flows*. For instance, the (2-dim) flow described by the equations

$$1 \leq \dot{x} \leq 2 \text{ and } \dot{x} = \dot{y},$$

describes motion diagonally upwards at a bounded speed. This flow is polyhedral and is admitted by the syntax of linear hybrid automata. On the other hand, the flow described by the equations

$$\dot{x} = x \text{ and } \dot{y} = 1,$$

describes motion along the exponential curve and is not polyhedral.

To generalize the flow constraints allowed by the syntax of linear hybrid automata, we consider flows that are described by *origin-dependent rate polytopes*, in which the allowed rates depend, not only on the current control mode, but also on the specific state at which the mode was entered. A sample 2-dimensional origin-dependent rate polytope is described by

$$\dot{x} = x_0 + 3 \text{ and } \dot{y} = y_0,$$

where (x_0, y_0) is the initial state upon entering the mode. Note that this particular flow is polyhedral. Unfortunately, not all origin-dependent rate polytopes specify polyhedral flows. We prove that flows described by the origin-dependent rate function $\dot{X} = \lambda \cdot X_0 + \Gamma$, where the vector X_0 denotes the initial values, λ is a constant, and Γ is a constant vector, are polyhedral. We also establish the tightness of our results by examples that are slight generalizations, but lead to non-polyhedral flows.

In addition to polyhedral flows, we also propose two related classes. In *strongly polyhedral* flows, for any time δ and for any polyhedron σ, the set $post_{\leq\delta}(\sigma)$ of states that can be reached at time δ or before, is a polyhedron. Such flows are relevant in the analysis of hybrid systems in which time is considered a special parameter and there is a specified upper bound on the time spent in each control mode. In *polyhedrally sliced* flows, for any time δ and for any polyhedron σ, the set $post_{\delta}(\sigma)$ of states that can be reached at time δ, is a polyhedron. Such flows are useful for symbolic simulation, or for analysis of hybrid systems in which the control switches occur at specified intervals. It turns out that, while the class of strongly polyhedral flows is strictly contained in the class of polyhedral flows, both these classes are incomparable to the class of polyhedrally sliced flows. We establish that all origin-dependent rate polytopes specify polyhedrally sliced flows and the origin-dependent rate function $\dot{X} = \lambda \cdot X_0 + \Gamma$ specifies strongly polyhedral flows (separable linear dependence).

Origin-dependent rate polytopes are suitable models of hybrid systems whose dynamics depend upon some initial conditions. They can also be used to approximate the dynamics of non-linear hybrid automata, and thus, to check automatically their properties. Non-linear hybrid automata can also be approximated by linear hybrid automata as shown in [HHW98] by transformations such as the *clock translation* and the *linear phase-portrait* methods. In this paper, we propose an efficient method, based on separable origin-dependent rate polytopes, to approximate variables whose dynamics is given by an exponential function of time. We first present a way to approximate exponential trajectories, and then we show how to over-approximate, with arbitrary precision, the set of states which are reachable via exponential dynamics. We observe that our techniques can be applied to any variable x and at each mode v such that the dynamics of x at v is given by $\dot{x} = x$. This makes our approach possible also when the clock translation method cannot be used. Moreover, given a hybrid automaton H, using our approximation method the number of modes of the obtained automaton is not greater than the number of modes of H. For phase-portrait approximations, instead, a splitting of the modes is required and better approximations can be obtained by refining this splitting.

Our results are useful in two ways. First, they establish that symbolic fixpoint computation using polyhedra can be used for analysis of classes that are more general than linear hybrid automata. Second, they suggest new ways to approximate the system dynamics.

The rest of this paper is organized as follows. In Section 2 we introduce the definitions. In Section 3 we discuss three subclasses of hybrid automata characterized respectively by *polyhedral*, *strongly polyhedral*, and *polyhedrally sliced* flows. In Section 4 we introduce and study the class of hybrid automata with origin-dependent rate polytopes. In Section 5, we discuss the use of origin-dependent rate polytopes to approximate non-linear dynamics. Finally, in Section 6 we give a few conclusions.

2 Hybrid Automata

A hybrid automaton [ACH+95] is a formal model to describe reactive systems with discrete and continuous components. Formally, a hybrid automaton H consists of the following components.

- A finite directed multi-graph (V, E). The vertices are called the *control modes* while the edges are called the *control switches.*

- A finite set X of real valued variables. A *valuation* ν is a function that assigns a real value $\nu(x)$ to each variable $x \in X$. The set of all valuations is denoted Σ_X. A *state* q is a pair (v, ν) consisting of a mode v and a valuation ν. The set of all states is denoted Σ. A *region* is a subset of Σ.

- A function *init*, that assigns to each mode v, a set $init(v) \subseteq \Sigma_X$ of valuations. This describes the initialization of the system: a state (v, ν) is initial if $\nu \in init(v)$. The region containing all initial states is denoted σ^I.

- A function *flow*, that assigns to each mode v, a set $flow(v)$ of differentiable functions from $\mathbb{R}_+$ to Σ_X. This describes the way variables can evolve in a mode.

- A function *inv*, that assigns to each mode v, a set $inv(v) \subseteq \Sigma_X$ of valuations. The system can stay in mode v only as long as the state is within $inv(v)$ and a switch must be taken before the invariant gets violated.

- A function *jump*, that assigns to each switch e, a set $jump(e) \subseteq \Sigma_X \times \Sigma_X$. This describes the enabling condition for a switch, together with the discrete update of the variables as a result of the switch.

The hybrid automaton H starts in an initial state. During its execution, its state can change in one of the two ways. A *discrete* change causes the automaton to change both its control mode and the values of its variables. Otherwise, a *continuous* activity causes only the values of variables to change according to the specified flows while maintaining the invariants. The operational semantics of the hybrid automaton is captured by defining transition relations over the state space Σ:

Discrete switches: For a switch $e = (v, v')$, we write $(v, \nu) \to_e (v', \nu')$ if $(\nu, \nu') \in jump(e)$.

Continuous flows: For a mode v and a time increment $\delta \in \mathbb{R}_+$, we write $(v, \nu) \to_\delta (v, \nu')$ if there exists a function $f \in \mathit{flow}(v)$ such that $f(0) = \nu$, $f(\delta) = \nu'$, and $f(\delta') \in inv(v)$ for all $0 \leq \delta' \leq \delta$.

The transition relations $\to_e$ capture the discrete dynamics, while the transition relations $\to_\delta$ captures the continuous dynamics.

Algorithms for symbolic reachability analysis of hybrid automata manipulate regions. Let σ be a region of H. For a switch e, $post_e(\sigma)$ contains states q' such that $q \to_e q'$ for some $q \in \sigma$. Similarly, define

$$post_\infty(\sigma) \;=\; \{\, q' \mid q \to_\delta q' \text{ for some } q \in \sigma \text{ and some } \delta \in \mathbb{R}_+ \,\}.$$

Then,

$$post(\sigma) \;=\; \cup_{e \in E}\, post_e(\sigma) \;\cup\; post_\infty(\sigma)$$

denotes the successor region of σ. A state q is said to be *reachable* if $q \in post^i(\sigma^I)$ for some natural number i.

For a region σ and a time increment $\delta \in \mathbb{R}_+$,

$$post_{\leq\delta}(\sigma) \;=\; \{\, q' \mid q \to_{\delta'} q' \text{ for some } q \in \sigma \text{ and some } 0 \leq \delta' \leq \delta \,\}.$$

Thus, $post_{\leq\delta}(\sigma)$ contains all the states that can be reached up to time δ starting in σ. Moreover, we denote with $post_\delta(\sigma)$ the set of the states that can be reached at time δ starting in σ, that is

$$post_\delta(\sigma) = \;\{\, q' \mid q \to_\delta q' \text{ for some } q \in \sigma \,\}.$$

3 Polyhedral Analysis

The central problem in algorithmic formal verification of hybrid systems is to compute the set of reachable states of a given hybrid automaton. The set of all reachable states of a hybrid automaton can be computed by repeatedly applying *post* to the initial region. Model checking of more complex requirements specified in the temporal logics such as CTL [CE81] and TCTL [ACD93], involves symbolic fixpoint computation procedures with the computation of the operator *post*, and its dual *pre*, as the basic building block [AHH96].

If a hybrid automaton has n variables, then a valuation can be viewed as a point in the n-dimensional Euclidean space. A region σ of H is a (convex) *polytope*[1] if there exists a mode v and a (convex) polytope σ' in the n-dimensional space such that σ equals $\{(v, \nu) \mid \nu \in \sigma'\}$. The mode v is called the mode of the polytope σ. In the sequel, we define subclasses of hybrid automata that admit analysis using polytopes.

3.1 Polyhedral Automata

Let $H = (V, E, X, init, flow, inv, jump)$ be a hybrid automaton.

1. The automaton has *polyhedral initialization* if for each mode v, the region $init(v)$ is a polytope.

2. The automaton has *polyhedral switches* if for each polytope σ and each switch e, the region $post_e(\sigma)$ is a polytope.

3. The automaton has *polyhedral flows* if for each polytope σ, the region $post_\infty(\sigma)$ is a polytope.

The hybrid automaton H is *polyhedral* if it has polyhedral initialization, polyhedral switches, and polyhedral flows. For a polyhedral hybrid automaton H, assuming the applications of *post* operators to polytopes are effectively computable, the set of reachable states can be computed in an iterative manner.

The definition of polyhedral hybrid automata is semantic. The requirements about polyhedral initialization and polyhedral switches naturally lead to syntax:

1. For each mode v, $init(v)$ is described by a boolean combination of linear inequalities over the variables X.

2. For each switch e, $jump(e)$ is described by a boolean combination of linear inequalities over the variables $X \cup X'$, where the unprimed variables X refer to the values before the switch and the primed variables X' refer to the values after the switch.

Note that, when the *jump* function is described in the above manner, for a polytope σ and a switch e, the polytope $post_e(\sigma)$ can be computed effectively using quantifier elimination (or projection) (see, for instance, [AHH96]).

The requirement that the flows be polyhedral is more tricky. Since the invariant describes the boundary at which switches are triggered, it is natural to require, for each mode v, the region $inv(v)$ to be a polytope. As before, this can be ensured syntactically:

> For each mode v, $inv(v)$ is described by a boolean combination of linear inequalities over the variables X.

[1] A polytope is a region of n-dimensional space enclosed by a finite number of hyperplanes. In this paper, we overload the word polytope also to refer to a finite union of polytopes.

In a *linear hybrid automaton*, a flow at a mode v is given by a *rate polytope*, denoted $rate(v)$, an n-dimensional convex polytope that constrains the allowed values of the first derivatives with respect to time. That is, a differentiable function f belongs to $flow(v)$ iff the first derivative $\dot{f}$ of f with respect to time belongs to the convex polytope $rate(v)$ for all times $\delta \in \mathbb{R}_+$. Rate polytopes are specified as a conjunction of linear inequalities over the set $\dot{X}$ of dotted variables representing the first derivatives of the corresponding variables in X. For instance, if the variables x and y denote the position of a robot on a plane, then the rate polytope

$$\dot{x} = \dot{y}, \;\; 1/\sqrt{2} \leq \dot{x} \leq \sqrt{2}$$

specifies the dynamics in which the robot moves diagonally at a speed between 1 and 2. Flows described by rate polytopes are polyhedral and, furthermore, allow effective computation of the *post* operator [AHH96]. Note that to restrict the functions $init(v)$, $jump(e)$, and $inv(v)$ to conjunctions of linear inequalities does not affect the expressiveness of the polyhedral automaton. Thus, in the sequel we will assume that $init(v)$, $jump(e)$, and $inv(v)$ are conjunctions of linear inequalities.

3.2 Strongly Polyhedral Automata

The flows specified by linear hybrid automata satisfy a property stronger than being polyhedral: not only the set of states reached as time passes starting from a polyhedron is a polyhedron, but the set of states reached up to any given time instant starting from a polyhedron is a polyhedron. This motivates the following definition:

> A hybrid automaton H automaton has *strongly polyhedral flows* if for each polytope σ and time increment $\delta \in \mathbb{R}_+$, the region $post_{\leq\delta}(\sigma)$ is a polytope. The hybrid automaton H is *strongly polyhedral* if it has polyhedral initialization, polyhedral switches, and strongly polyhedral flows.

The strong polyhedral property is useful in the following special case. Suppose the automaton has a special timer variable t that is used only to ensure an upper bound on the time spent in each mode. That is, t is reset to 0 on every switch and, for every mode v, the invariant $inv(v)$ implies $t \leq \delta_v$, for some constant δ_v. As a consequence, if σ is a region with mode v, then we have that $post_\infty(\sigma)$ is exactly $post_{\leq\delta_v}(\sigma)$. Thus, we could remove t as a system variable and compute, at each mode v, the reachable sets using $post_{\leq\delta_v}$ instead of $post_\infty$. Clearly, if the flows are strongly polyhedral, then $post_{\leq\delta_v}$ can be computed using polyhedra, and thus we could use polyhedral analysis on an automaton with dimension reduced by one. Notice that reducing dimensions can be computationally important in the analysis of hybrid automata.

The strong polyhedral requirement implies polyhedral flows:

Remark 1 *Strongly polyhedral flows are polyhedral.*

This result follows directly from the definitions by compactness. Observe that the reverse does not hold as shown in the following example.

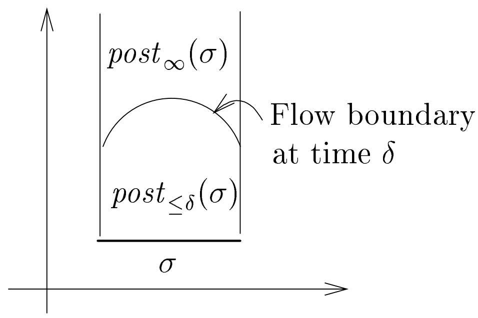

Figure 1: Polyhedral, but not strongly polyhedral, flow.

Example 1 Consider a flow given by $\dot{x} = 0$, $\dot{y} = -x^2 + 2x$ and $x = x_0$ where (x_0, y_0) is the starting point of a trajectory (i.e., trajectories are vertical lines according to the usual Cartesian representation in the flat). For any polytope σ, thus we have that $post_\infty(\sigma)$ is the union of half straight lines defined by $(x_0, y_0) + \delta\,(0, -x_0^2 + 2x_0)$ for $\delta \geq 0$ and $(x_0, y_0) \in \sigma$. We observe that for x varying in $\mathbb{R}$ the sign of $-x^2 + 2x$ changes only a finite number of times: it is negative for either $x < 0$ or $x > 2$, and positive for $0 < x < 2$. Thus, $post_\infty(\sigma)$ is the union of a finite number of adjacent polytopes and therefore is a polytope.

To see that the above flow is not strongly polyhedral, consider a segment σ with middle point in $x = 1$ and such that $-x^2 + 2x > 0$ for any of its points. As shown in Figure 1, for any δ, $post_{\leq\delta}(\sigma)$ is not a polytope. ■

3.3 Polyhedrally Sliced Automata

Finally, we consider the class for which the set of states reachable at a given time instant starting from a polyhedron is a polyhedron.

> A hybrid automaton H has *polyhedrally sliced flows* if for each polytope σ and time increment $\delta \in \mathbb{R}_+$, the region $post_\delta(\sigma)$ is a polytope. The hybrid automaton H is *polyhedrally sliced* if it has polyhedral initialization, polyhedral switches, and polyhedrally sliced flows.

In general, symbolic simulation at fixed time slices can be performed effectively using polyhedra for polyhedrally sliced automata. Analysis of polyhedrally sliced flows can be used for model checking in the following special case. Suppose the automaton has a special timer variable t such that for every mode v, the invariant $inv(v)$ implies $t \leq \delta_v$, for some constant δ_v, and every switch from mode v resets t to 0 and its guard implies $t = \delta_v$. Then, it is possible to do the symbolic analysis by computing the successor region $post_{\delta_v}(\sigma)$ of a polytope with mode v. If the flow is polyhedrally sliced, such regions can be computed using polyhedra.

Remark 2 *The class of polyhedrally sliced flows in not comparable with the class of strongly polyhedral flows and the class of polyhedral flows.*

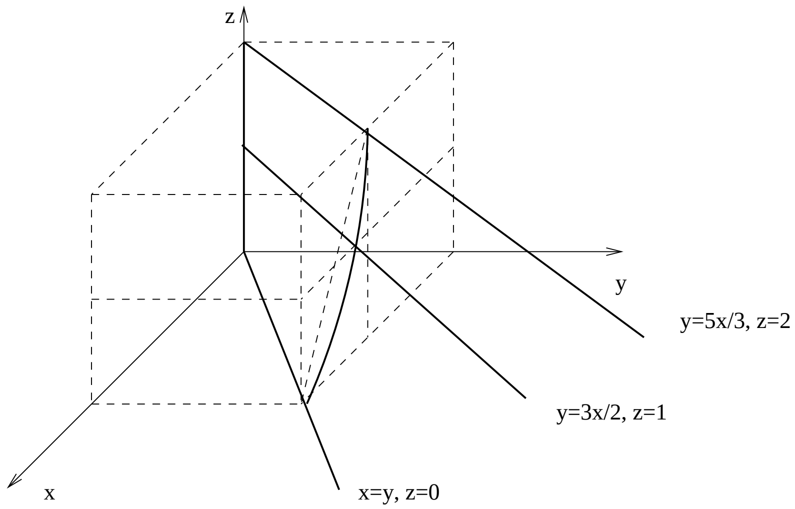

Figure 2: A polyhedrally sliced flow which is not polyhedral.

The above remark is established by the following examples.

Example 2 Consider the line segment $x = y$ in the plane $z = 0$. Suppose the dynamics is such that the point (x_0, y_0, z_0) on the line moves to the position $x = (\delta + 1)x_0$, $y = (2\delta + 1)y_0$ and $z = \delta + z_0$, at time δ. Then, the line traces a curved surface as time progresses (e.g. at time 1, it is the line $y = 3x/2$ in the plane $z = 1$, and at time 2, it is the line $y = 5x/3$ in the plane $z = 2$, see Figure 2). The corresponding flow is polyhedrally sliced but it is not polyhedral. ■

Example 3 Consider the flow F_1 given by $\dot{x} = 2$ and $\dot{y} = 2\,\tau$ for $\tau \in [1, 2]$ and the flow F_2 given by $\dot{x} = \gamma$ and $\dot{y} = \gamma^2$ for $\gamma \in [1, 2]$. In the rest of this example, for a given flow F we denote the sets $post_{\leq\delta}(\sigma)$ and $post_\delta(\sigma)$ respectively by $post^F_{\leq\delta}(\sigma)$ and $post^F_\delta(\sigma)$. It is simple to verify that the flow F_1 is both strongly polyhedral and polyhedrally sliced, while the flow F_2 is neither strongly polyhedral or polyhedrally sliced. Consider the flow $F = F_1 \cup F_2$. By definition, $post^F_{\leq\delta}(\sigma) = post^{F_1}_{\leq\delta}(\sigma) \cup post^{F_2}_{\leq\delta}(\sigma)$ and $post^F_\delta(\sigma) = post^{F_1}_\delta(\sigma) \cup post^{F_2}_\delta(\sigma)$. We claim that F is strongly polyhedral but it is not polyhedrally sliced.

To see that F is strongly polyhedral we show that for any $\sigma \subseteq \mathbb{R}^2$ and any $\delta \in \mathbb{R}_+$, $post^{F_2}_{\leq\delta}(\sigma) \subseteq post^{F_1}_{\leq\delta}(\sigma)$ and thus $post^F_{\leq\delta}(\sigma) = post^{F_1}_{\leq\delta}(\sigma)$. Let $(x, y) \in post^{F_2}_{\leq\delta}(\sigma)$. The case $(x, y) \in \sigma$ is trivial. Suppose $(x, y) \notin \sigma$. Then there exist γ, δ', x' and y' such that $\gamma \in [1, 2]$, $0 < \delta' \leq \delta$, $(x', y') \in \sigma$, $x = x' + \gamma\,\delta'$ and $y = y' + \gamma^2\,\delta'$. Now, let δ'' be such that $\delta'' = \frac{\gamma}{2}\delta'$. From $\gamma \in [1, 2]$, we have that $0 < \delta'' \leq \delta' \leq \delta$. Since $x' + \gamma\,\delta' = x' + 2\,\delta''$ and $y' + \gamma^2\,\delta' = y' + 2\,\gamma\,\delta''$, we have that $(x, y) \in post^{F_1}_{\leq\delta}(\sigma)$. To complete the proof of the above claim, consider the segment $l = [1, 2] \times \{1\}$. The sets $post^{F_1}_1(l)$ and $post^{F_2}_1(l)$ are respectively the rectangle and the curved surface in Figure 3. Clearly, their union, i.e. $post^F_1(l)$, is not a polytope, and thus F is not polyhedrally sliced.

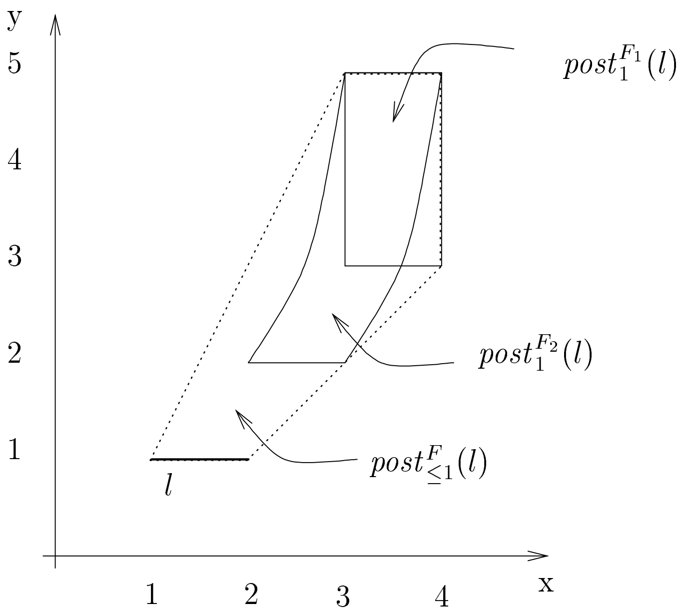

Figure 3: A strongly polyhedral, but not polyhedrally sliced, flow.

4 Rates Dependent on Initial State

Recall that in a linear hybrid automaton with n variables, a flow at a mode is given by the n-dimensional rate polytope that constrains the allowed values of the first derivatives. Our objective is to understand if there are polyhedral flows that are more general than those specified by rate polytopes. The dynamics of a linear hybrid automaton can be understood as follows. After switching to a mode v, the system chooses a rate belonging to the convex polytope *rate*(v) and then evolves at this fixed rate[2]. In our generalization, the rate polytope depends, not only on the mode, but also on the state upon entering the mode. More precisely, the flow is specified by a function which maps a mode v to a $2n$-dimensional convex polytope $rate(v) \subseteq \mathbb{R}_+^n \times \mathbb{R}_+^n$. After switching to a mode v, if the valuation is ν_0, then the system chooses a rate r such that (ν_0, r) belongs to the convex polytope *rate*(v) and then evolves at this fixed rate. That is, a differentiable function f belongs to *flow*(v), for a mode v, iff $(f(0), \dot{f}(\delta)) \in$ *rate*(v) for all times $\delta \in \mathbb{R}_+$. We call such flows *origin-dependent rate polytopes.*

We will use X_0 to denote the values of the corresponding variables in X at the beginning of a flow. Then, origin-dependent rate polytopes will be described by constraints involving the dotted variables $\dot{X}$ and the variables X_0. A sample 2-dimensional origin-dependent rate polytope is described by

$$\dot{x} \;=\; x_0 + 2 \text{ and } \dot{y} \;=\; y_0.$$

Figure 4 illustrates how different points evolve according to this flow for a sample region.

[2]This is because when the rate polytope is convex: there is a trajectory f starting in a valuation ν and ending in a valuation ν' with $\dot{f}$ belonging to the rate polytope at all times iff the rate of the straight-line trajectory from ν to ν' is in the rate polytope.

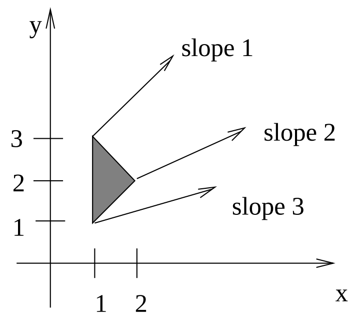

Figure 4: Illustration of origin-dependent rates: $\dot{x} = x_0 + 2$ and $\dot{y} = y_0$.

4.1 General Properties of Origin-Dependent Rate Polytopes

An interesting feature of the origin-dependent rate polytopes is that they are polyhedrally sliced flows, as stated in the following theorem.

Theorem 1 *Let H be a hybrid automaton with origin-dependent rate polytopes at a mode v. Then, H has polyhedrally sliced flows at v.*

Proof. Since $post_\delta(\sigma_1 \cup \sigma_2) = post_\delta(\sigma_1) \cup post_\delta(\sigma_2)$, it is enough to prove the theorem for convex polytopes. Then, let σ be a convex polytope with mode v and $\delta \in \mathbb{R}_+$ be a time increment. From the convexity of σ and $rate(v)$, it follows that $post_\delta(\sigma)$ is convex. To show that $post_\delta(\sigma)$ is a polytope, then it is enough to prove that it has a finite number of vertices. Consider $(p, r_p) \in rate(v)$ such that (p, r_p) is not a vertex of $rate(v)$. We claim that $q = p + r_p\delta$ is not a vertex in $post_\delta(\sigma)$. Since (p, r_p) is not a vertex of $rate(v)$, there exist $(a, r_a), (b, r_b) \in rate(v)$, and c, with $0 < c < 1$, such that $(p, r_p) = c\,(a, r_a) + (1 - c)(b, r_b)$. Let $s, t \in post_\delta(\sigma)$ be obtained respectively by allowing a and b to evolve for time δ along rate vectors r_a and r_b. Then $q = cs + (1 - c)t$ proving that q is not a vertex. Thus, the number of vertices in $post_\delta(\sigma)$ is bounded by the number of vertices in $rate(v)$ which is a convex polytope and therefore has a finite number of vertices. ■

Observe that for a flow function to be polyhedral, a necessary condition is that, for every state q, $post_\infty(q)$ is a polytope. This condition is not satisfied by flows described by differential equations in which the first derivative depends on the state in a continuous manner, but holds for origin-dependent rate polytopes. On the other hand, origin-dependent rate polytopes do not guarantee polyhedral flows. In fact, the flow described in the Example 2 is an origin-dependent rate polytope but is not a polyhedral flow. (If different variables exhibit different types of evolution at the same mode, then the resulting flows are not polyhedral.) Finally, note that if for a mode v, the rate polytope $rate(v)$ is independent of the entering point p, then the automaton has polyhedral flows at v.

4.2 Separable Linear Dependence

Now we consider the case when the rate of change of a variable x_i depends linearly only on the initial value of x_i. That is, for each variable x_i, there exist constants λ_i and γ_i such that an origin-dependent

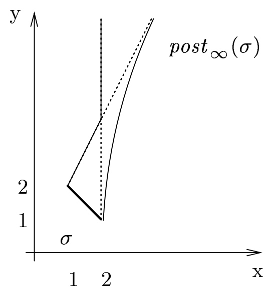

Figure 5: A non-polyhedral origin-dependent rate polytope with separable variables.

rate polytope is specified by $\dot{x_i} = x_i(0) \cdot \lambda_i + \gamma_i$. Thus, different components evolve independently, and the flow function at a mode is completely specified by the two vectors Λ and Γ. We call such an origin-dependent rate function *separable*.

Does a separable rate function guarantee polyhedral flows? The answer is positive only if we add a further requirement, namely, that all the components of the vector Λ are the same. For instance, the 3-dimensional origin-dependent rate polytope described by $\dot{x} = x_0 + 2$, $\dot{y} = y_0$, and $\dot{z} = z_0 - 5$ is polyhedral.

We state a technical lemma that allows us to prove the main theorem about separable rate functions.

Lemma 1 *For $n \geq 2$, consider a flow given by an n-dimensional separable rate function. Let $p_1, p_2 \in \sigma$, and $q_1, q_2 \in post_\infty(\sigma)$ be such that q_1 and q_2 are distinct and $q_i = post_{\delta_i}(p_i)$ for $i = 1, 2$ where $\delta_i \in \mathbb{R}_+$ and $p_i \in \sigma$. If the flow is polyhedral then points p_1, p_2, q_1 and q_2 belong to the same plane (2-dimensional space) of $\mathbb{R}^n$.*

Proof. For $n = 2$ the result is trivial. Now, let $n > 2$. Suppose that p_1, p_2, q_1, and q_2 are not in the same plane and let σ' be the segment joining p_1 and p_2. We claim that, for any $p'_1, p'_2 \in \sigma'$, $p'_1 \neq p'_2$, and for any $q'_i \in post_\infty(p'_i)$, $i = 1, 2$, it holds that p'_1, p'_2, q'_1, and q'_2 are not co-planar. To establish this, we observe that, starting from a point p in σ', the associated rate is a convex linear combination of the rates from p_1 and p_2. Thus it is easy to prove that the segment joining p'_1 to q'_1 lies on the same plane as the segment joining p'_2 to q'_2 if and only if the segment joining p_1 to q_1 lies on the same plane as the segment joining p_2 to q_2. As a consequence there are infinitely many maximal[3] convex polytopes contained in $post_\infty(\sigma')$ which are pairwise disjoint. Thus $post_\infty(\sigma')$ cannot be expressed as the union of finitely many polyhedra and, hence, the flow is not polyhedral. ∎

Example 2 gives an origin-dependent rate polytope whose successor region is a curved surface. It is worth noting that, in general, the converse of the above lemma does not hold. A counterexample is given by the flow described by $\dot{x} = 2 - x_0$ and $\dot{y} = y_0$. It is an origin-dependent rate polytope with separable linear dependence, and obviously, for any σ, all the points of $post_\infty(\sigma)$ are co-planar. But, if we consider the region σ given by the segment joining the points $(1, 2)$ and $(2, 1)$, we have that $post_\infty(\sigma)$ is not a polyhedron (see Figure 5).

[3]The term maximal is used here in the sense that the union of two or more such polytopes is not a convex polytope.

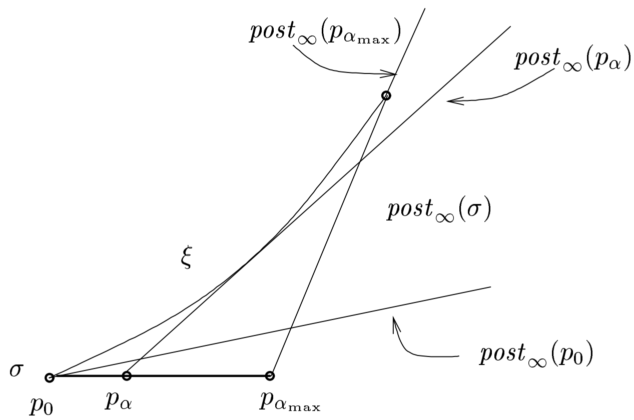

Figure 6: The time successor region of the segment $\sigma = \overline{p_0 p_{\alpha_{\max}}}$.

Theorem 2 *Suppose the flow function at a mode v is described by the origin-dependent separable rate function specified by the vectors Λ and Γ. The automaton has strongly polyhedral flows at the mode v if $\lambda_i = \lambda_j$ for all variables x_i and x_j. Conversely, if this condition is not satisfied, there exists a polytope σ such that $post_\infty(\sigma)$ is not polyhedral.*

Proof. We consider first the converse direction of the theorem for 2-dimensional separable origin-dependent rate functions, then for n-dimensional ones and finally we prove the forward direction.

Consider a 2-dimensional separable function $\dot{x}_i = x_i(0)\,\lambda_i + \gamma_i$. We want to show that if $\lambda_1 \neq \lambda_2$ then there exists a polytope σ such that $post_\infty(\sigma)$ is not a polytope. Let p_0 be a general point (y_1, y_2) of the plane. Consider a segment defined by points $p_\alpha = (y_1, y_2) + \alpha\,(1, 0)$ for $\alpha \in [0, \alpha_{\max}]$. We claim that for each choice of $\lambda_1, \lambda_2, \gamma_1, \gamma_2$ such that $\lambda_1 \neq \lambda_2$, there exist $y_1, y_2, \alpha_{\max}$ such that either (1) all the angular coefficients of the half straight lines $post_\infty(p_\alpha)$ are positive and grow for increasing values of α, or (2) they are all negative and decreasing for decreasing values of α. We leave it to the reader to verify this claim. Either one of properties (1) and (2) ensures that a portion of the boundary of the region $post_\infty(\sigma)$ is a curved line and thus $post_\infty(\sigma)$ is not a polytope. As an example of this, consider the case given in Figure 6 where $y_1, y_2, \alpha_{\max}$ are such that the angular coefficients of $post_\infty(p_\alpha)$ are positive and grow for increasing values of α. When property (2) holds, we have a symmetric case.

Consider now an n-dimensional origin-dependent separable rate function described by the vectors Λ and Γ with $n > 2$. Let p and q be two distinct points of $post_\infty(\sigma)$ such that, $p = post_{\delta_1}(a)$ and $q = post_{\delta_2}(b)$ where $\delta_i \in \mathbb{R}_+$, for $i = 1, 2$, and $a, b \in \sigma$. Without loss of generality assume that $\delta_1 \leq \delta_2$. Let p_i denote the i^{th} component of p and similarly for the other points, we have the following relations:

$$\begin{aligned} p_i &= a_i + (\lambda_i a_i + \gamma_i)\delta_1, \\ q_i &= b_i + (\lambda_i b_i + \gamma_i)\delta_2, \end{aligned}$$

The four points a, b, p, q are co-planar if and only if $p - q$ is linearly dependent on $p - a$ and $p - b$. We compute the i^{th} components of these difference vectors.

$$
\begin{aligned}
(p-q)_i &= (a_i - b_i) + (\lambda_i a_i + \gamma_i)\delta_1 - (\lambda_i b_i + \gamma_i)\delta_2 \\
(p-a)_i &= (\lambda_i a_i + \gamma_i)\delta_1 \\
(p-b)_i &= (a_i - b_i) + (\lambda_i a_i + \gamma_i)\delta_1
\end{aligned}
$$

Since we need to prove that there exist a polytope σ and four non-co-planar points $a, b, p, q \in post_\infty(\sigma)$ defined as above. By setting $\delta_1 = \delta_2 = 1$ we get the equations:

$$
\begin{aligned}
(p-q)_i &= (a_i - b_i)(1 + \lambda_i) \\
(p-a)_i &= \lambda_i a_i + \gamma_i \\
(p-b)_i &= a_i - b_i + \lambda_i a_i + \gamma_i
\end{aligned}
$$

For co-planarity of the original points we must have constants c_1 and c_2 such that for all i,

$$
c_1(a_i - b_i)(1 + \lambda_i) + c_2(\lambda_i a_i + \gamma_i) = a_i - b_i + a_i\lambda_i + \gamma_i
$$

By hypothesis there are i, j such that $\lambda_i \neq \lambda_j$. Consider first the case that $\lambda_i \neq -1$, and $\lambda_j \neq -1$. We can choose a and b such that $a_1 = b_1 \neq 0$ and $a_1 \neq \frac{\gamma_1}{\lambda_1}$. From the above equations, we have that $c_2 = 1$. Substituting this value for c_2 we get the contradiction that c_1 must be equal to both $\frac{1}{1+\lambda_i}$ and $\frac{1}{1+\lambda_j}$. In the other case, that is when $\lambda_i \neq -1$ and $\lambda_j = -1$, we have that the above equation reduces to

$$
c_2(\gamma_j - a_j) = \gamma_j - b_j
$$

If we choose $a_j = \gamma_j \neq b_j$, we have that for any c_2 the above equality cannot be satisfied, and thus the considered points are not in the same plane. By Lemma 1, we can thus conclude that there exists a convex polytope σ which includes the points a and b and such that $post_\infty(\sigma)$ is not polyhedral.

Consider now the forward direction of the theorem. Since $post_{\leq\delta}(\sigma_1 \cup \sigma_2) = post_{\leq\delta}(\sigma_1) \cup post_{\leq\delta}(\sigma_2)$, we can restrict the proof of the theorem to convex polytopes. Let σ be a convex polytope denoted by the equation $AX \leq B$, where A is a constant matrix, X is the vector of variables, and B is a constant vector. Suppose the dynamics is given by $\dot{X} = \lambda X_0 + \Gamma$. Then,

$$
post_{\leq\delta}(\sigma) \;=\; \{\, Y \mid \exists\delta', X.\; 0 \leq \delta' \leq \delta \;\wedge\; Y = X + \delta'(\lambda X + \Gamma) \;\wedge\; AX \leq B \,\}
$$

Eliminating X, we have that $post_{\leq\delta}(\sigma)$ is

$$
\left\{\; Y \mid \exists\delta'.\; 0 \leq \delta' \leq \delta \;\wedge\; A\left[\frac{Y - \delta'\Gamma}{1 + \lambda\delta'}\right] \leq B \;\wedge\; \delta' \neq -\frac{1}{\lambda} \;\right\} \cup \left\{\; Y \mid Y = -\frac{1}{\lambda}\,\Gamma \;\wedge \lambda \leq -\frac{1}{\delta} \;\right\}
$$

Hence, $post_{\leq\delta}(\sigma)$ can be written as

$$
\left(\exists\delta'.\; 0 \leq \delta' \leq \delta \;\wedge\; AY - A\delta'\Gamma \leq (1 + \lambda\delta')B \;\wedge\; \delta' \neq -\frac{1}{\lambda}\right) \;\vee\; \left(Y = -\frac{1}{\lambda}\Gamma \;\wedge \lambda \leq -\frac{1}{\delta}\right)
$$

This is a boolean combination of linear expressions and the constraints on Y can be found by eliminating the variable δ. Then $post_{\leq\delta}(\sigma)$ is a polytope when all the λ_i are equal. ■

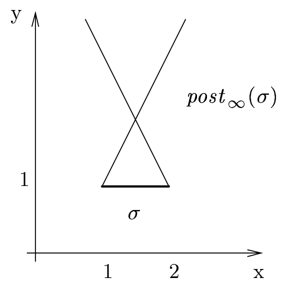

Figure 7: An origin-dependent rate polytope with separable variables, not expressible by a rate polytope.

Note that the proof of the forward direction of the above theorem is effective in the sense that gives an algorithm to compute the set $post_{\leq\delta}(\sigma)$. This algorithm can be easily modified to compute $post_\infty(\delta)$, in the restricted case of separable origin-dependent rate polytopes characterized by having all the λ-coefficients equal, and $post_\delta(\sigma)$ for any separable origin-dependent rate polytope. In the first case we use quantifier elimination and the formula

$$\exists\delta.\ \left(\left(\delta \geq 0\ \wedge\ AY - A\delta\Gamma \leq (1+\lambda\delta)B\ \wedge\ \delta \neq -\frac{1}{\lambda}\right)\ \vee Y = -\frac{1}{\lambda}\Gamma\ \right)$$

In the other case, from Theorem 1 the computation of $post_\delta(\delta)$ is possible for all flows given by $\dot{X} = X_0 \cdot \Lambda + \Gamma$, and can be done via the formula

$$post_\delta(\sigma) = \exists X.\ Y = X + \delta(X \cdot \Lambda + \Gamma)\ \wedge\ AX \leq B$$

It is worth noting that the origin-dependent rate polytopes with separable linear dependence enlarges the class of hybrid automata which can be effectively analyzed. Consider the flow given by

$$\dot{x} \ = \ -x_0 + \frac{3}{2},\ \ \dot{y} \ = \ -y_0 + 2$$

and the line segment σ joining the two points $(1,1)$ and $(1,2)$ in the 2-dimensional space (see Figure 7). This dynamics cannot be captured by a rate polytope and, hence, there is no equivalent linear hybrid automaton.

5 Approximation of non-linear hybrid automata

In this section we discuss the approximation of non-linear dynamics by origin-dependent rate polytopes. We observe that origin-dependent rate polytopes are suitable models of hybrid systems whose dynamics depend upon some initial conditions. Also, origin-dependent rate polytopes turn out to be very useful to approximate dynamics described by exponential functions of time. We present here a simple way to obtain flows that resemble those generated by exponential functions, and also show how to over-approximate, with arbitrary precision, the set of states reachable by exponential flows.

Given a constant $T > 0$, consider the exponential-like function $y(t,T)$ defined as the piecewise linear function whose vertices are the points (kT, e^{kT}), for $k \in \mathbb{N}$. To approximate an exponential flow, we

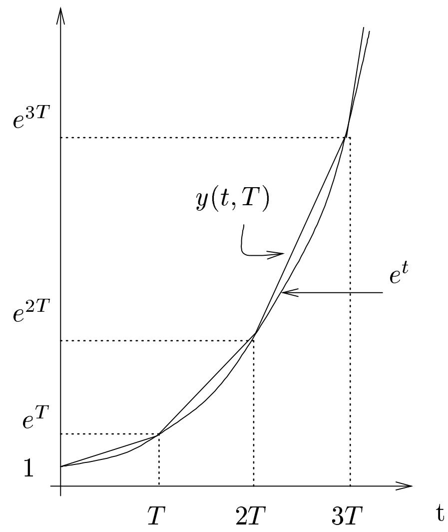

Figure 8: Approximation of the function e^t with a piecewise linear function.

model $y(t,T)$ by origin-dependent rate polytopes (see Figure 8). Thus, we approximate by the segments $e^{kT} + t\,\frac{e^{(k+1)T} - e^{kT}}{T}$ the function e^t on the intervals $[kT,(k+1)T]$. We observe that the error $|e^t - y(t,T)|$ corresponding to this approximation is diverging with t, and on each interval $[kT,(k+1)T]$, an upper bound on the error is $(e^{(k+1)T} - e^{kT})$. Clearly, reducing the constant T also reduces the error and we obtain better approximations.

Consider a mode v and a variable x such that, for some $k \in \mathbb{N}$ and $T > 0$, x is reinitialized to e^{kT} on all the transitions to v. It is possible to model the flow given by $y(t,T)$ at v by the variable x using the formalism of the origin-dependent rate polytopes. For this purpose, we associate with v the flow constraint [4] $\dot{x} = x_0\,\frac{e^T-1}{T}$, and a clock variable t (i.e., a variable with flow constraint $\dot{t} = 1$). Moreover, we add a self-loop transition on v which is triggered whenever t reaches T. To enforce this, we add a conjunct $0 \le t < T$ to the guards of all the other transitions from v, and the self-loop transition is guarded by $t = T$ and resets t (see Figure 9). When v is entered for the first time, x is equal to e^{kT}. After that the self-loop transition is taken for the first time, x evaluates $e^{kT} + T\,e^{kT}\,\frac{e^T-1}{T}$, that is $e^{(k+1)T}$. By induction on the number of times the self-loop transition is taken before taking another transition, it is possible to prove that x evolves according to $y(t,T)$ starting from the value e^{kT}.

The assumption about the initial value of x can be relaxed, that is, we can allow any initial value for x. To capture the flow $y(t,T)$ the above construction can be modified in the following way. In mode v, we introduce a new variable y which approximates e^t by $y(t,T)$, and use x to store the value at the time v is entered, say x_0. Since in the original automaton the value of x at time t after entering v is $x_0 + e^t - 1$, we set y to 1 on the transitions to v, and on all the transitions from v, replace x by $x+y-1$ in the guards, and if x is not reinitialized, we set x to $x+y-1$. The flows in v for x and y are given

[4]Notice that e^T may not be rational even if T is rational, thus we need to approximate it with a rational value to use in the flow constraints associated with v. We will not take into consideration this aspect here since it introduces some technicalities that do not bear on the approach we present in this section.

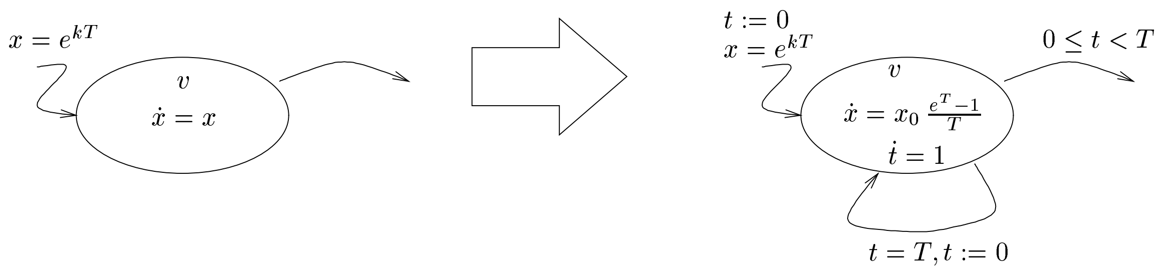

Figure 9: Origin-dependent rate polytopes to model $y(t, T)$.

respectively by $\dot{x} = 0$ and $\dot{y} = y_0 \frac{e^T-1}{T}$. This construction can be applied iteratively to approximate by origin-dependent rate polytopes all the flow constraints of type $\dot{x} = x$. We observe that if we need to approximate several variables, we need to add only a variable for all of them since we can use the same constant T for all of them. Moreover, we only need a clock variable, since in the construction it is used locally to each mode. We can thus state the following result.

Lemma 2 *For an n-dimensional hybrid automaton H, let X be the set of flow constraints of H of type $\dot{x} = x$. It is possible to construct a hybrid automaton H' that models the flow constraints in X by origin-dependent rate polytopes approximating e^t by $y(t, T)$. Moreover, the size of H' is $O(|H|)$ and the dimension of H' is $n + 2$.*

In general, the above construction does not preserve properties of the starting hybrid automaton, but still shows an easy way to obtain approximated exponential behaviours using origin-dependent rate polytopes. A simple modification of the above construction will allow us to over-approximate with an arbitrary precision the exponential dynamics of hybrid automata, and thus the resulting model will be suitable to check safety properties. In fact, if H' over-approximate H and the "bad region" is not reachable in H', then it is not reachable also in H. Vice-versa, if the bad region is reachable in H' we cannot establish anything for H, but we can repeat the control on a finer approximation of H. To obtain an over-approximation of e^t we use triangles whose vertices are (kT, e^{kT}), $((k+1)T, e^{(k+1)T})$, and the intersection between the straight line tangent to e^t at the value e^{kT} and the vertical line through the t value $(k + 1)T$ (Figure 10). Clearly, for $t \in [kt, (k + 1)T]$, the points (t, e^t) are all contained in the above triangle, moreover reducing the constant T we can get better over-approximations.

We can use origin-dependent rate polytopes to model flows captured by these triangles. We modify the construction given to model $y(t, T)$. First we assign to y in v a rate in the interval with endpoints y_0 and $y_0 \frac{e^T-1}{T}$, that is, we use the flow constraint $\dot{y} \in [y_0, y_0 \frac{e^T-1}{T}]$. To keep the over-approximation close to the exponential curve, we reinitialize y to $y_0(e^T - 1)$ on the added self-loop transition on v [5]. In Figure 11, we show this construction on a the mode v.

We show now that the approximation method we have presented in this section is appropriate to check safety properties of a given hybrid automaton. Given $\bar{a}, \bar{b} \in \mathbb{R}^n$ we define the distance between $\bar{a}$

[5] Notice that this is not captured directly by the standard syntax of linear jump constraints. It is possible to prove that allowing this kind of origin-dependent reinitialization, we still have polyhedral jumps.

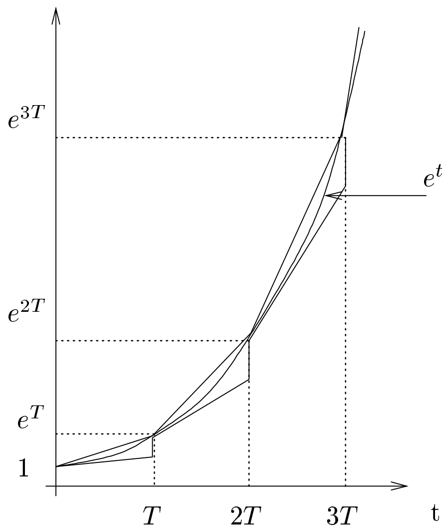

Figure 10: Over-approximation of e^t by triangles.

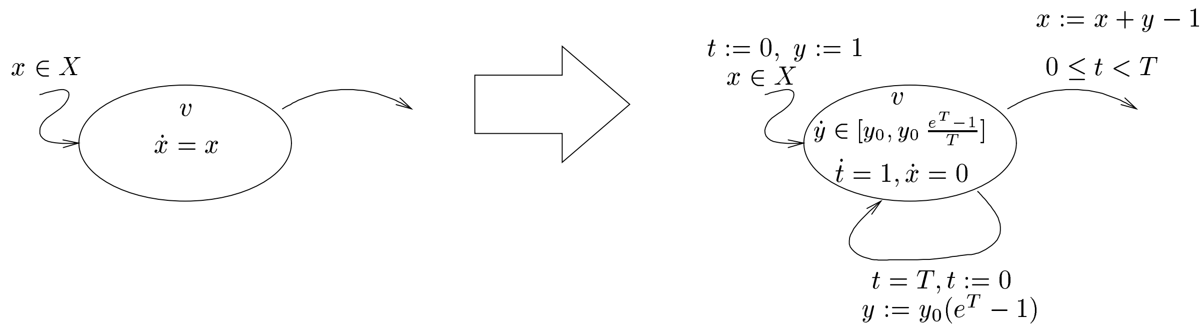

Figure 11: Origin-dependent rate polytopes for over-approximating e^t.

and $\bar{b}$ as the maximum of the differences $|a_i - b_i|$, for $i = 1, \ldots, n$. Given a hybrid automaton H and $\epsilon > 0$, an ϵ-relaxation of H is a hybrid automaton obtained from H by replacing any initial, invariant, jump, and flow condition ϕ with a condition ϕ' such that if $\bar{a}$ satisfies ϕ then the distance between $\bar{a}$ and $\bar{b}$ is less than ϵ, for some $\bar{b}$ satisfying ϕ'. For a hybrid automaton H, let X be the set of flow constraints $\dot{x} = x$ assigned to modes v of H such that x is upper-bounded in v. Denote by $\gamma(H, X, T)$ the hybrid automaton obtained by over-approximating the flow constraints in X via triangles with approximation constant T. We say that $\gamma(H, X, T)$ is *asymptotically-complete*, if for any $\epsilon > 0$ there exists a $T > 0$ and an ϵ-relaxation H' of H such that H' can capture the behaviours of $\gamma(H, X, T)$ [6]. This last property guarantees that if a hybrid automaton H violates a safety property ϕ, then there exists a $T > 0$ such that the automaton $\gamma(H, X, T)$ violates ϕ. Thus the following statement holds:

Let H be a hybrid automaton with polyhedral initialization and polyhedral switches such that

[6] In other words, H' *time simulates* $\gamma(H, X, T)$. See [HHW98] for a formal definition of time simulation of hybrid automata.

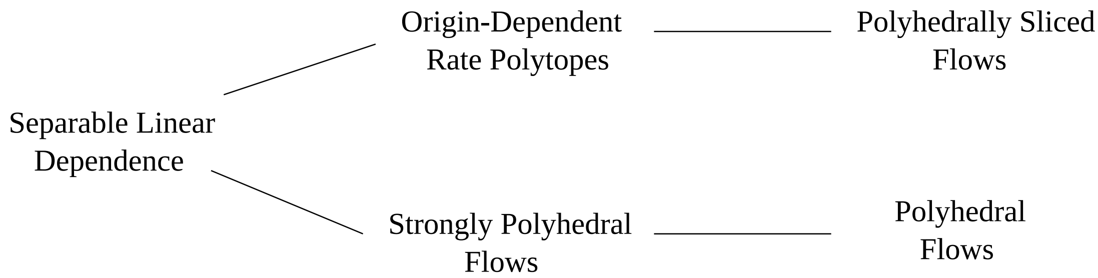

Figure 12: Relationships among the classes of polyhedral, strongly polyhedral, polyhedrally sliced flows and flows determined by origin-dependent rate polytopes and separable linear dependence.

the flows of any variable x at any mode v are given by either:

- *origin-dependent rate polytopes,*
- *rate polytopes or*
- $\dot{x} = x$.

Then, the polyhedral analysis on H is possible.

6 Conclusions

We have studied ways to generalize the syntax of linear hybrid automata without sacrificing the polyhedral property of the resulting flows. The specific generalizations we look at are motivated by an on-going study aimed at application of formal modeling and verification to problems in multi-robot coordination [AEK$^+$99]. In particular, we consider origin-dependent rate polytopes in which the dynamics depend, not only on the control mode, but also on the state upon entry to the mode. Not all origin-dependent rate polytopes determine polyhedral flows. This motivates the study of separable linear dependence in origin-dependent rate polytopes. Besides polyhedral flows, we have also considered strongly polyhedral flows and polyhedrally sliced flows. The relationships among the various classes are summarized in Figure 12.

Origin-dependent rate polytopes offer a different way to approximate complex dynamics. When the flow specified by an origin-dependent rate polytope is not polyhedral, $post_\infty(\sigma)$ can be approximated by the convex-hull of the $post_\infty$ of the corner-points of σ. Such an approximation seems appropriate for origin-dependent rate polytopes for two reasons. First, the $post_\infty$ of a corner point is guaranteed to be polyhedral and easily computable. Second, the convex-hull of the $post_\infty$ of the corner-points of σ is guaranteed to include $post_\infty(\sigma)$, a property not true of arbitrary dynamics. We have also discussed a method to approximate exponential dynamics by origin-dependent rate polytopes with linear separable dependence. For this class of systems our approximation method is very natural compared to other approximation techniques and still allows us to perform analysis by manipulating polyhedra.

References

[ACD93] R. Alur, C. Courcoubetis, and D.L. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2 – 34, 1993.

[ACH+95] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3 – 34, 1995.

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

[AEK+99] R. Alur, J. Esposito, M. Kim, V. Kumar, and I. Lee. Formal modeling and analysis of hybrid systems: A case study in multirobot coordination. In *Proceedings of the Formal Methods '99*, LNCS 1708, pages 212–232. Springer, 1999.

[AHH96] R. Alur, T.A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. *IEEE Transactions on Software Engineering*, 22(3):181 – 201, 1996.

[BCD+92] J.R. Burch, E.M. Clarke, D.L. Dill, L.J. Hwang, and K.L. McMillan. Symbolic model checking: 10^{20} states and beyond. *Information and Computation*, 98(2):142 – 170, 1992.

[CE81] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proc. of Workshop on Logic of Programs*, LNCS 131, pages 52 – 71. Springer-Verlag, 1981.

[CK96] E.M. Clarke and R.P. Kurshan. Computer-aided verification. *IEEE Spectrum*, 33(6):61 – 67, 1996.

[HHW97] T.A. Henzinger, P. Ho, and H. Wong-Toi. HyTech: a model checker for hybrid systems. *Software Tools for Technology Transfer*, 1, 1997.

[HHW98] T.A. Henzinger, P.H. Ho, and H. Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 43:540 – 554, 1998.

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

[Hol97] G.J. Holzmann. The model checker spin. *IEEE Transanctions on Software Engineering*, 23(5):279 – 295, 1997.

[HW95] P.H. Ho and H. Wong-Toi. Automated analysis of an audio control protocol. In *Proc. of the Seventh Conference on Computer-Aided Verification*, LNCS 939, pages 381 – 394. Springer-Verlag, 1995.

[HW96] T.A. Henzinger and H. Wong-Toi. Linear phase-portrait approximations of nonlinear hybrid systems. In *Hybrid Systems III: Verification and Control*, LNCS 1066, pages 377 – 388. Springer-Verlag, 1996.

[LPS00] G. Lafferriere, G.J. Pappas, and S. Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems*, 13(1):1–21, 2000.