

Hybridization for stability analysis of switched linear systems

Pavithra Prabhakar^{*}
Kansas State University, Manhattan, KS
pprabhakar@ksu.edu

Miriam García Soto
IMDEA Software Institute, Madrid, Spain
miriam.garcia@imdea.org

ABSTRACT

In this paper, we present a hybridization method for stability analysis of switched linear hybrid system (*LHS*), that constructs a switched system with polyhedral inclusion dynamics (*PHS*) using a state-space partition that is specific to stability analysis. We use a previous result based on quantitative predicate abstraction to analyse the stability of *PHS*. We show completeness of the hybridization based verification technique for the class of asymptotically stable linear system and a subclass of switched linear systems whose dynamics are pairwise Lipschitz continuous on the state-space and uniformly converging in time. For this class of systems, we show that by increasing the granularity of the region partition, we eventually reach an abstract switched system with polyhedral inclusion dynamics that is asymptotically stable. On the practical side, we implemented our approach in the tool *Averist*, and experimentally compared our approach with a state-of-the-art tool for stability analysis of hybrid systems based on Lyapunov functions. Our experimental results illustrate that our method is less prone to numerical errors and scales better than the traditional approaches. In addition, our tool returns a counterexample in the event that it fails to prove stability, providing feedback regarding the potential reason for instability. We also examined heuristics for the choice of state-space partition during refinement.

1. INTRODUCTION

Stability is a fundamental property in the design of cyber-physical systems that ensures robustness of these systems with respect to perturbations in the initial states or inputs. Linear hybrid systems (*LHS*) are an important class of cyber-physical systems, that manifest in several embedded control systems. In addition, more often than not, the design and analysis of non-linear control systems is conducted by considering linearizations of these systems at different

^{*}Pavithra Prabhakar is partially supported by EU FP7 Marie Curie Career Integration Grant no. 631622 and NSF CAREER award no. 1552668.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](http://permissions.acm.org).

HSCC '16, April 12–14, 2016, Vienna, Austria.

© 2016 ACM. ISBN 978-1-4503-3955-1/16/04...\$15.00

DOI: <http://dx.doi.org/10.1145/2883817.2883840>

operating points. Hence, linear hybrid systems have been extensively studied in the literature; see, for instance, [10] for a survey of methods for stability analysis.

Stability analysis is a challenging problem, especially, for hybrid systems. For instance, while it is well-known that the stability of a purely continuous linear dynamical system can be inferred solely based on analyzing its eigenvalues, the same is not true when switching is allowed between these systems [4]. Hence, alternate methods based on exhibiting Lyapunov functions, which provide a certificate of stability, and their extensions to common and multiple Lyapunov functions for switched system analysis have gained prominence [9].

Computational methods for stability analysis essentially consist of a template based search for a Lyapunov function [3, 12]. Here, a template, for instance, a polynomial with coefficients as parameters, is chosen. The conditions on the Lyapunov functions, such as, non-negativity, are encoded in a constraint solving formalism. For instance, a sufficient condition for a polynomial to be non-negative is to be expressible as a sum of squares of terms. Efficient tools for semi-definite programming and sum-of-squares optimization are employed to solve for the parameters in the templates.

While the template based search has gained popularity due to the advances in convex optimization tools over the last decade, there are some fundamental difficulties with the approach which have not been adequately addressed. First, finding the right template requires user ingenuity, and the alternative of exhaustively iterating over all possible templates is highly inefficient. Next, the stability analysis tools provide no insights into the reason for instability or guide the choice of better templates, when the solvers fail to instantiate the template.

To overcome the difficulties with the template based search, an alternate approach based on abstraction-refinement was proposed by the authors in their previous work [14, 15] for stability analysis of hybrid systems with piecewise constant and polyhedral inclusion dynamics. Here, a quantitative predicate abstraction (QPA) for stability analysis was presented, which extends the standard finite state graph construction of predicate abstraction [7], by annotating the edges with weights which capture information about the distance of the state to the equilibrium point during evolution along a trajectory. This approach returned a counterexample when the abstraction failed to prove stability.

One of the main difficulties in extending QPA to linear hybrid systems, is the computation of the edges and their

weights, which require computing a reachability relation to the linear hybrid systems. This is a challenging problem, especially, when the number of edge switchings is not bounded. Hence, we propose to abstract the linear hybrid system to a polyhedral hybrid system for which the edge relation and weights can be efficiently computed (even in the presence of unbounded number of mode switches [15]).

Our main result is a novel application of the hybridization approach [1, 5] for stability analysis. Broadly, hybridization consist of splitting the state-space into a finite number of regions and approximating the dynamics in each of the regions by a simpler dynamics. Our hybridization is specialized for stability analysis, and differs from those used for safety analysis, in the class of partitions considered for hybridization. We use partitions that split the state-space into conic polyhedral sets. The linear dynamics in each of the regions is overapproximated by a polyhedral inclusion dynamics. Our partitioning is fine tuned for stability analysis, since, it guarantees a bound on the “scaling” in a region, ratio of the distance to the origin at the end of the execution to that at the beginning. Hybridization techniques for safety analysis instead seek a bound on the approximation error in the reach set [1, 5].

Our main theoretical result is the completeness of the hybridization technique for the class of asymptotically stable switched linear systems that are “uniformly converging in time” and Lipschitz continuous. For this class of systems, we show that by increasing the granularity of the region partition, we eventually reach a polyhedral switched system abstraction that is asymptotically stable.

We have implemented our approach in the tool **Averist**. We experimentally compared our approach with the state-of-the-art tool for stability analysis of hybrid systems based on Lyapunov functions, **Stabhyli**. Our experimental results illustrate that our method is less prone to numerical errors and scales better than the traditional approaches. In addition, our tool returns a potential counterexample in the event that it fails to prove stability, providing feedback regarding the potential reason for instability. This is a result of the quantitative algorithmic method that we use in the back-end for analyzing polyhedral switched systems. We also investigate some heuristics for partitioning the state-space in a more efficient manner.

2. HYBRID AUTOMATA

In this section, we define hybrid automata, which are a popular formalism for capturing the mixed discrete-continuous behaviors of cyber-physical systems. We consider switched systems that do not allow instantaneous jumps in the continuous state of the system. Below, we define a subclass of switched hybrid systems. Let $Poly(n)$ denote the set of all convex polyhedra in the n -dimensional state-space. In the sequel, by a polyhedral set, we always mean a convex polyhedral set.

Definition. A *hybrid automaton (HA)* is a tuple $\mathcal{H} = (Loc, Edges, Cont, Flow, Inv, Guard)$, where:

- Loc is a finite set of control modes or locations;
- $Edges \subseteq Loc \times Loc$ is a finite set of edges;
- $Cont = \mathbb{R}^n$, for some n , is the continuous state-space;

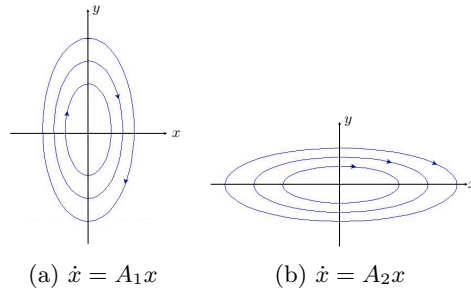


Figure 1: Linear systems

- $Flow : Loc \rightarrow (\mathbb{R}^n \mapsto 2^{\mathbb{R}^n})$ is the vector field function;
- $Inv : Loc \rightarrow Poly(n)$ is the invariant function; and
- $Guard : Edges \rightarrow Poly(n)$ is the guard function.

We call n , the dimension of \mathcal{H} .

The hybrid automaton evolves continuously by following the vector field function at each location. It executes starting in a mode q and a continuous state x . For every location $q \in Loc$ the state follows the vector field $Flow(q)$ while being inside the invariant $Inv(q)$. The continuous state switches from one location to other when it satisfies the guard between these locations. The vector field function is typically represented as a differential inclusion $\dot{x} \in F(x)$ or a linear differential equation $\dot{x} = Ax$, where $F(x)$ represents a polyhedron and A is a real matrix.

We say that a hybrid automaton is *linear* if in every mode q the behaviour of the system is determined by a linear dynamical system of the form $\dot{x} = A_q x$. Hence, for every $q \in Loc$ and $x \in Inv(q)$, the flow in the hybrid automaton is given by $Flow(q)(x) = \{A_q x\}$. And we say that a hybrid automaton is *polyhedral*, when, for every mode q , the dynamics is defined by a polyhedral inclusion of the form $\dot{x} \in P_q$ where P_q is a polyhedron. Here, the flow is specified as $Flow(q)(x) = P_q$ for every $q \in Loc$ and $x \in Inv(q)$.

Let \mathcal{I} denote the set of all closed intervals over \mathbb{R} . Given an interval $I \in \mathcal{I}$, let $first(I)$ and $last(I)$ denote the greatest lower bound and the least upper bound of I . Let $F|_X$ denote the restriction of the function F to the set X as its domain.

Definition. An *execution* of a HA $\mathcal{H} = (Loc, Edges, Cont, Flow, Inv, Guard)$ is a function $\sigma : I \rightarrow \mathbb{R}^n$, where $I = [0, T]$ or $I = [0, \infty)$, such that there exists a finite or infinite sequence $(q_0, I_0), (q_1, I_1), (q_2, I_2) \dots$ satisfying for all $i \geq 0$:

- $(q_i, I_i) \in Loc \times \mathcal{I}$, such that $\cup_i I_i = I$ and $last(I_i) = first(I_{i+1})$;
- $\sigma|_{I_i}$ is a solution of $\dot{x} \in Flow(q_i)(x)$, that is, $\frac{d}{dt} \sigma|_{I_i}(t) \in Flow(q_i)(\sigma|_{I_i}(t))$ and satisfies $\sigma|_{I_i}(t) \in Inv(q_i)$ for all $t \in I_i$ and $\sigma(last(I_i)) \in Guard(q_i, q_{i+1})$.

Let $Exec(\mathcal{H})$ denote the set of all executions of \mathcal{H} .

The evolution of a linear dynamical system is determined by the equation $\dot{x} = Ax$ where A is a real square matrix. Consider two linear dynamical systems of the form $\dot{x} = A_1 x$ and $\dot{x} = A_2 x$, where the matrices are given as follows

$$A_1 = \begin{pmatrix} 0 & 1 \\ -4 & 0 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0 & 1 \\ -0.1 & 0 \end{pmatrix}$$

The phase portraits showing the solutions for each of these systems are depicted in Figure 1. A linear dynamical system defines the evolution of a linear hybrid automaton at each location. A linear hybrid automaton model is depicted in Figure 2. Observe this system consists of four modes, q_1, q_2, q_3 and q_4 , with associated invariants R_1, R_2, R_3 and R_4 , which correspond to the first, second, third and fourth quadrants of the two dimensional plane, respectively. The dynamics A_1 and A_2 are as given in Figure 6. Hence, the system evolves in the first and third quadrants following a solution of the dynamical system $\dot{x} = A_1x$ and in the other two quadrants following one of $\dot{x} = A_2x$. The switching condition is along the common boundaries, and is specified by the guard annotations on the edges. A sample execution of the system is given in Figure 3.

A polyhedral hybrid automaton is shown in Figure 4, analogous to the linear one but with dynamics determined by the differential inclusion $\dot{x} \in P_i$ for each mode q_i . A sample execution of the polyhedral system is depicted in Figure 5.

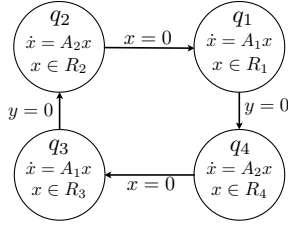


Figure 2: Linear hybrid automaton \mathcal{H}

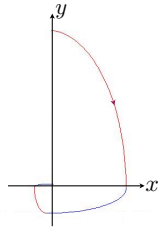


Figure 3: A sample execution of \mathcal{H}

3. STABILITY

Stability is a fundamental property in control system design. We consider two classical notions of stability in control theory, namely, Lyapunov and asymptotic stability. An *equilibrium point* is a state of the system such that an execution starting from it remains there. Origin $\bar{0}$ will be an equilibrium point in the systems we consider, and we define stability with respect to the origin. A system is said to be *Lyapunov stable* if small perturbations in the initial state lead to small deviations in the executions with respect to the equilibrium point. Further, it is said to be *asymptotically stable* if the executions starting in a small neighborhood of the equilibrium point converge to the point. In the following $B_r(\bar{0})$ refers to a ball of radius r around $\bar{0}$.

Definition. A hybrid system \mathcal{H} is said to be *Lyapunov stable* if for every $\epsilon > 0$, $\exists \delta > 0$ such that for every $\sigma \in Exec(\mathcal{H})$ $[(\sigma(0) \in B_\delta(\bar{0})) \Rightarrow (\forall t \in dom(\sigma), \sigma(t) \in B_\epsilon(\bar{0}))]$.

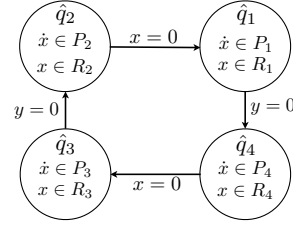


Figure 4: Polyhedral hybrid automaton $\hat{\mathcal{H}}$

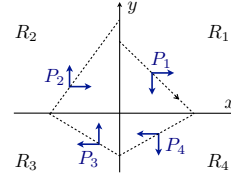
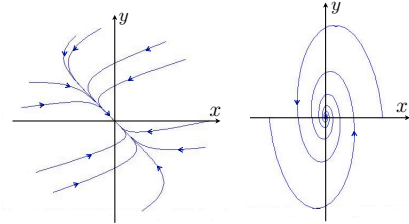


Figure 5: A sample execution of $\hat{\mathcal{H}}$



(a) $\dot{x} = A_3x$ (b) $\dot{x} = A_4x$

Figure 6: Sample Executions

Definition. A hybrid system \mathcal{H} is said to be *asymptotically stable* if it is Lyapunov stable and

$$\exists \gamma > 0 \forall \sigma \in Exec(\mathcal{H}) [(\sigma(0) \in B_\gamma(\bar{0})) \Rightarrow Conv(\sigma, \bar{0})].$$

$$Conv(\sigma, \bar{0}) := \forall \epsilon > 0 \exists T \geq 0 \forall t \geq T \sigma(t) \in B_\epsilon(\bar{0}).$$

In fact, for the purpose of stability analysis, we can assume that the invariants and guards of the linear hybrid system are closed under positive scaling. A polyhedral set P is *closed under positive scaling* if for every $x \in P$, $\alpha x \in P$ for all $\alpha \geq 0$. Note that the Lyapunov and asymptotic stability analysis with respect to origin depend on the behavior of the system in a small neighborhood of the equilibrium point. Hence, the stability of every hybrid system is equivalent to the stability of a hybrid system constructed by removing constraints which do not pass through $\bar{0}$ (or equivalently, replacing all the invariants and guards by their positive scaling closure.)

Observe that the executions of linear systems depicted in Figure 1 determine both systems to be Lyapunov stable. The systems shown in Figure 6 correspond to asymptotically stable systems, while if the executions were pointing out they would be unstable.

4. HYBRIDIZATION FOR STABILITY ANALYSIS OF LINEAR HYBRID AUTOMATA

In this section, we formally present the hybridization procedure for constructing a polyhedral hybrid automaton from a linear hybrid automaton, and provide soundness and completeness results.

4.1 Hybridization construction

The basic construct in the hybridization procedure is the abstraction of the linear vector field in a region by a polyhedron. This is done by the function $CPoly$. Given $F : \mathbb{R}^n \rightarrow 2^{\mathbb{R}^n}$ and $P \subseteq \mathbb{R}^n$, let $CPoly(F, P) = \cup_{x \in P} F(x)$. Next, we present the formal definition of hybridizing a system \mathcal{H} with respect to a partition \mathcal{R} .

A partition \mathcal{R} of the state space \mathbb{R}^n is a finite set of polyhedra $\{R_1, \dots, R_k\}$ such that $\mathbb{R}^n = \cup_{i=1}^k R_i$ and for every pair of different indices i, j , $R_i \cap R_j$ has affine dimension smaller than n . Two polyhedra $P, Q \subseteq \mathbb{R}^n$ are adjacent in case of $P \cap Q$ has affine dimension $n - 1$, and we denote as $P \text{adj } Q$.

Definition.[Hybridization] Given an n -dimensional linear hybrid system $\mathcal{H} = (Loc, Edges, Cont, Flow, Inv, Guard)$ and a polyhedral partition $\mathcal{R} = \{R_1, \dots, R_k\}$, we define the hybridization of \mathcal{H} with respect to \mathcal{R} , as $PolyH(\mathcal{H}, \mathcal{R}) = (\widehat{Loc}, \widehat{Edges}, Cont, \widehat{Flow}, \widehat{Inv}, \widehat{Guard})$, where:

- $\widehat{Loc} = \{(q, R) : q \in Loc, R \in \mathcal{R} \text{ and } R \subseteq Inv(q)\}$
- $\widehat{Edges} = \{((q_i, R_i), (q_j, R_j)) \in \widehat{Loc} \times \widehat{Loc} : q_i = q_j \text{ or } (q_i, q_j) \in Edges \text{ and } R_i \text{adj } R_j\}$
- $\widehat{Flow}((q, R))(x) = CPoly(Flow(q), R)$.
- $\widehat{Inv}(q, R) = Inv(q) \cap R$.
- $\widehat{Guard}((q_i, R_i), (q_j, R_j)) = \mathbb{R}^n$ if $q_i = q_j$ and $Guard((q_i, q_j))$, otherwise.

Note that for every $q \in Loc$ and $R \in \mathcal{R}$, $CPoly(Flow(q), R)$ is a polyhedral set. Hence, the polyhedral differential inclusion $\dot{x} \in CPoly(Flow(q), R)$ over-approximates the dynamical system $\dot{x} \in Flow(q)(x)$ in the polyhedral region R . Then, it follows that $PolyH(\mathcal{H}, \mathcal{R})$ is a polyhedral hybrid automaton.

Remark. We will assume from now on that the partition \mathcal{R} used in the hybridization only consists of conic polyhedral sets, that is, those that are positive scaling closed. Our hybridization is similar to that in the literature [17, 6], where rectangular/polyhedral abstractions are considered, except that we consider a partition of the state-space specific to stability analysis.

4.1.1 Illustration

We explain in detail the hybridization process for a particular 2-dimensional linear hybrid system example. The dynamical behaviour of the example is determined by the linear switched system explained in Section 2.

To perform hybridization, we choose a polyhedral partition of the continuous state-space. Consider the polyhedral partition $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$, formed by the planar quadrants, where R_i corresponds to the i -th one. This choice

determines a polyhedral hybrid system with the same number of modes as in the linear automaton, with the same associated invariants, and the same edges labelled with the same guards. The polyhedral dynamics in the four modes are given by $\dot{x} \in P_1$, $\dot{x} \in P_2$, $\dot{x} \in P_3$ and $\dot{x} \in P_4$, respectively, where $P_1 = CPoly(A_1, R_1)$, $P_2 = CPoly(A_2, R_2)$, $P_3 = CPoly(A_1, R_3)$ and $P_4 = CPoly(A_2, R_4)$. For instance, in the first quadrant all the flow vectors of the linear hybrid system are between $(1, 0)$ and $(0, -1)$, therefore P_1 is determined by the constraints $x > 0$ and $y < 0$, as we observe in Figure 5. The polyhedral hybrid system is shown in Figure 4. However, this polyhedral hybrid automaton is not stable. To obtain a stable one, we need to use a finer partition.

4.2 Stability preservation theorem

Next, we show that our hybridization procedure is sound, in that, the stability of the polyhedral hybrid system output by the hybridization procedure implies the stability of the linear hybrid system. This is formalized below.

Theorem 1. *Given a linear hybrid automaton \mathcal{H} and a polyhedral partition \mathcal{R} , the following hold:*

1. *$PolyH(\mathcal{H}, \mathcal{R})$ is Lyapunov stable implies \mathcal{H} is Lyapunov stable.*
2. *$PolyH(\mathcal{H}, \mathcal{R})$ is asymptotically stable implies \mathcal{H} is asymptotically stable.*

PROOF. This follows from the fact that $Exec(\mathcal{H}) \subseteq Exec(PolyH(\mathcal{H}, \mathcal{R}))$, which in turn follows from the fact that the polyhedral differential inclusion $\dot{x} \in CPoly(F, R)$ over-approximates the dynamical system $\dot{x} = F(x)$ in the polyhedral region R . \square

Remark. It has been observed in [13] that simulations do not preserve stability and the stronger notion of uniformly continuous simulations is required. Note that in Theorem 1, the simulation relation between the two systems is the identity relation which trivially satisfies the additional constraints of uniform continuity imposed by [13].

4.3 Completeness results for hybridization

In this section, we identify a subclass of switched linear hybrid systems for which the hybridization procedure is complete, that is, for every asymptotically stable system in this class, by iteratively choosing finer and finer partitions for the hybridization, we will arrive at an abstract polyhedral hybrid system that is asymptotically stable.

Before identifying the subclass of systems referred above, we provide a detailed proof for the class of linear dynamical systems. We then sketch how this proof can be extended for the class that we define. Let \mathcal{H}_A denote the linear hybrid automaton (with one mode) corresponding to the dynamical system $\dot{x} = Ax$.

Theorem 2. *Let \mathcal{H}_A be a linear hybrid automaton corresponding to the dynamical system $\dot{x} = Ax$ which is asymptotically stable. Then, there exists a polyhedral partition \mathcal{R} such that the polyhedral hybrid system $PolyH(\mathcal{H}_A, \mathcal{R})$ is asymptotically stable with respect to $\bar{0}$.*

The hybridization techniques for safety analysis ensure a bounded error in the executions of the original and approximate systems over a bounded time. They rely on the

fact that error between the concrete and the abstract vector fields is bounded. However, this is no more true in our setting. More precisely, the Hausdorff distance between Ax (the value of the concrete vector field) and $P = \{Ax \mid x \in R\}$, the abstract vector field in a region R (positively closed) is infinity. Hence, our proof of completeness is bit more involved. We construct an intermediate system which is equivalent to the abstract system. In this new system, though we cannot bound the distance between the vector fields of the concrete system and that of the intermediate system at state x by some ϵ , we can bound it by $\epsilon\|x\|$.

Observe that the scaling in a region R is determined by the vector field restricted to the state-space with norm 1. Let $CUPoly(A, R) = \{Ax \mid x \in R, \|x\| = 1\}$.

Definition. We define the scaled hybridization of \mathcal{H} with respect to \mathcal{R} , as $SPolyH(\mathcal{H}, \mathcal{R}) = (\widehat{Loc}, \widehat{Edges}, \widehat{Cont}, \widehat{Flow}, \widehat{Inv}, \widehat{Guard})$, where all the elements in the tuple are the same as for the hybrid system $PolyH(\mathcal{H}, \mathcal{R})$ but for the vector field function, which for every $(q, R) \in \widehat{Loc}$, $\widehat{Flow}((q, R))(x) = CUPoly(Flow(q), R)\|x\|$.

The following lemmas comprise the proof for Theorem 2.

Lemma 1. $SPolyH(\mathcal{H}, \mathcal{R})$ is asymptotically stable if and only if $PolyH(\mathcal{H}, \mathcal{R})$ is asymptotically stable.

Lemma 2. Let A be an $n \times n$ matrix and let \mathcal{H}_A be the hybrid system associated to the matrix. Let \mathcal{H}_A be asymptotically stable with respect to $\bar{0}$. Then, there exists a partition \mathcal{R} such that the polyhedral system $SPolyH(\mathcal{H}_A, \mathcal{R})$ is asymptotically stable with respect to $\bar{0}$.

Definition. Given a polyhedral set P and a scalar a , let $Pa = \{xa : x \in P\}$. The norm of a polyhedral set P is $\|P\| = \max\{\|p\| \mid p \in P\}$.

The Lemma 2 depends on the following two propositions.

Proposition 1. Consider a polyhedral partition $\mathcal{R} = \{R_1, \dots, R_k\}$ and a finite set of bounded polyhedral sets $\{P_1, \dots, P_k\}$. Consider two differential inclusions,

$$\dot{x} \in \{Ax\} \quad (1)$$

$$\dot{x} \in F(x) \quad (2)$$

where $F(x) = P_i\|x\|$ for $x \in R_i$ and for every $0 \leq i \leq k$. Let $d_H(Ax, F(x)) \leq \epsilon\|x\|$ for every x . Let x and y be solutions of 1 and 2 respectively, such that $x(0) = y(0)$. Then,

$$\|x(t) - y(t)\| \leq \frac{\epsilon\|y(0)\|}{m - \|A\|} (e^{mt} - e^{\|A\|t})$$

for every $t \geq 0$ with $m = \max_{0 \leq i \leq k} \|P_i\|$.

PROOF. Consider x and y be solutions of 1 and 2 respectively, such that $x(0) = y(0)$. Define the error function $z(t) = \|x(t) - y(t)\|$ for $t \in [0, \infty)$. For infinity norm we know that $\frac{d}{dt}\|f(t)\| \leq \|\frac{d}{dt}f(t)\|$. Then, $\dot{z} \leq \|\dot{x} - \dot{y}\|$, which is equal, for some $y' \in F(y)$, to $\|Ax - y'\| = \|Ax - Ay + Ay - y'\| \leq \|Ax - Ay\| + \|Ay - y'\|$, and since $\|Ay - y'\| \leq \epsilon\|y'\|$, We obtain the differential system

$$\begin{cases} \dot{z}(t) \leq \|A\|z(t) + \epsilon\|y'\| \\ \dot{y} \in F(y) \end{cases}$$

Let us first solve the inclusion in the system. Consider $y \in R_i$. Then we have $\dot{y} \in P_i\|y\|$, therefore $\|\dot{y}\| \in \|P_i\|y\| = \|P_i\|\|y\|$. Define $m = \max_{0 \leq i \leq k} \|P_i\|$. Therefore, $\|\dot{y}\| \leq m\|y\|$. Denote $\|y(t)\| = r(t)$. Since $\frac{d}{dt}\|y\| \leq \|\dot{y}\|$, we get the inequality $\dot{r}(t) \leq mr(t)$. Since infinity norm is differentiable everywhere except at the origin, and there is no time such that we get zero norm, we can apply Grönwall's lemma, which states that the inequality solution is bounded by the correspondent differential equation. We get the solution $r(t) \leq r(0)e^{mt}$. Plug the solution to the first equation in the system, $\dot{z}(t) \leq \|A\|z(t) + \epsilon e^{mt}r(0)$ where $z(t)$ is differentiable everywhere but in the origin and by applying also the Grönwall's lemma and solving the differential equation, we obtain

$$z(t) \leq e^{\|A\|t}z(0) + \frac{\epsilon r(0)}{m - \|A\|} (e^{mt} - e^{\|A\|t}) \quad (3)$$

Since $z(0) = 0$, the inequality is just

$$z(t) \leq \frac{\epsilon r(0)}{m - \|A\|} (e^{mt} - e^{\|A\|t}) \quad (4)$$

□

Proposition 2. Let A be an $n \times n$ matrix and \mathcal{H}_A be the hybrid system associated to the matrix. Consider $\epsilon > 0$. Then, there exists a partition $\mathcal{R} = \{R_1, \dots, R_k\}$ such that $d_H(Ax, P_i\|x\|) \leq \epsilon\|x\|$ for every $x \in R_i$ where $P_i = CUPoly(A, R_i)$.

PROOF. Ax is a continuous function and the state-space with norm 1 is compact. Hence, Ax restricted to the state-space of norm 1 is a uniformly continuous function. Therefore, given any ϵ , there exists a δ , such that for every x, y with $\|x - y\| \leq \delta$, $\|Ax - Ay\| \leq \epsilon$. We can choose a polyhedral partition such that each of the regions restricted to norm 1 is contained in a δ ball. Consider a region R_i , we want to prove that $d_H(Ax, P_i\|x\|) \leq \epsilon\|x\|$. We know $d_H(Ax, P_i\|x\|) = \|Ax - Ay\|$ for some $y \in R_i$ such that $\|y\| = 1$. Then, $\|Ax - Ay\| = \|Ax - Ay'\|$ where $y' \in R_i$ and $\|y'\| = \|x\|$, so we can rewrite as $\|(A\frac{x}{\|x\|} - A\frac{y'}{\|y'\|})\|x\|$. Since $\|\frac{x}{\|x\|} - \frac{y'}{\|y'\|}\| \leq \delta$ by the partition construction, we get $\|A\frac{x}{\|x\|} - A\frac{y'}{\|y'\|}\| \leq \epsilon$ and finally, by substituting, $d_H(Ax, P_i\|x\|) \leq \epsilon\|x\|$. □

Theorem 1 states that our hybridization based analysis approach is sound in that if the polyhedral abstraction is stable, then so is the linear hybrid system. Here, we show that our analysis method is complete for linear dynamical systems with respect to asymptotic stability, that is, given a linear dynamical system which is asymptotically stable, one can construct a polyhedral abstraction which is stable.

Proposition 3. Let $\dot{x} = Ax$ be a dynamical system in \mathbb{R}^n which is asymptotically stable. Then, for every $\gamma > 0$ there exists a value $T_\gamma \geq 0$ such that for every solution $x(t)$ with $\|x(0)\| = \gamma$, $\|x(t)\| \leq \frac{\gamma}{2}$ holds for every $t \geq T_\gamma$.

PROOF. Let $\dot{x} = Ax$ be asymptotically stable. Then A can be written as PJP^{-1} , where P is a real invertible matrix and J is the real Jordan matrix [8]. A solution $x(t)$ of the system can be expressed as $e^{At}x(0) = e^{PAP^{-1}t}x(0)$, which is equal to $Pe^{Jt}P^{-1}x(0)$ by definition of matrix exponential. Because of norm matrix properties, we get $\|x(t)\| \leq \|P\|\|e^{Jt}\|\|P^{-1}\|\gamma$. The asymptotic stability implies that

A is Hurwitz, and therefore the matrix e^{Jt} is also Hurwitz, which means that every term in it is of the form $e^{\lambda t}p(t)$ where $\lambda < 0$ and $p(t)$ is a polynomial function. Then, every term in $Pe^{Jt}P^{-1}x(0)$ is equal to $e^{\lambda t}p'(t)$ where $p'(t)$ is a different polynomial and the values will be dominated by the exponentials. Hence, for every row, $x_i(t)$, there exists $T_\gamma^i > 0$ such that $\|x_i(t)\| \leq \frac{\gamma}{2}$ for every $t \geq T_\gamma^i$. So it is enough to choose $T_\gamma = \max_{0 \leq i < n} (T_\gamma^i)$. \square

Proof of Lemma 2.

Suppose \mathcal{H}_A to be asymptotically stable. Fix $\gamma \geq 0$. We know by Proposition 3 that there exists $T_\gamma \geq 0$ such that for every $\sigma \in Exec(\mathcal{H}_A)$ with $\|\sigma(0)\| = \gamma$, $\|\sigma(t)\| \leq \frac{\gamma}{2}$ for every $t \geq T_\gamma$. Fix $\epsilon = \frac{\gamma(m - \|A\|)}{4\|\sigma'(0)\|(e^{mT_\gamma} - e^{\|A\|T_\gamma})}$. By Proposition 2 we know there exists a partition \mathcal{R} such that $d_H(Ax, CUPoly(A, \mathcal{R})) \leq \epsilon\|x\|$. Consider $\sigma \in Exec(\mathcal{H}_A)$ and $\sigma' \in Exec(SPolyH(\mathcal{H}_A, \mathcal{R}))$ with $\sigma(0) = \sigma'(0)$. Then, by Proposition 1 $\|\sigma(t) - \sigma'(t)\| \leq \frac{\gamma}{4}$. Since we also know that $\|\sigma(T_\gamma)\| \leq \frac{\gamma}{2}$, we get $\sigma'(T_\gamma) \leq \frac{3\gamma}{4}$. It is possible to iterate such construction for values of γ smaller and smaller, obtaining $\sigma' \in SPolyH(\mathcal{H}_A, \mathcal{R})$ closer and closer to zero for some time values.

Definition. A linear hybrid system \mathcal{H} is Lipschitz continuous if there exists a constant L such that for every $x, y \in Cont_{\mathcal{H}}$ with $x \in Inv(q_1)$ and $y \in Inv(q_2)$ the following holds $\|A_{q_1}x - A_{q_2}y\| \leq L\|x - y\|$. We say that \mathcal{H} is uniformly converging if for every $\gamma > 0$, there exists $T_\gamma \geq 0$ such that for every execution σ of \mathcal{H} , $\|\sigma(0)\| = \gamma$ implies $\|\sigma(t)\| \leq \gamma/2$ for all $t \geq T_\gamma$.

Theorem 3. *Let \mathcal{H} be a linear hybrid system that is Lipschitz continuous and uniformly converging. Then, there exists a polyhedral partition \mathcal{R} such that the polyhedral hybrid system $PolyH(\mathcal{H}, \mathcal{R})$ is asymptotically stable with respect to 0.*

PROOF. Here we sketch the proof of the theorem, which is similar to that of the proof of Theorem 2. Proposition 1 can be extended to linear switched systems which are Lipschitz continuous. The error between the solution of \mathcal{H} and some approximation whose vector field is at most ϵ far from \mathcal{H} can be analyzed as follows. $\dot{z} \leq \|\dot{x} - \dot{y}\| = \|A_i x - y'\| = \|A_i x - A_j y + A_j y - y'\| \leq \|A_i x - A_j y\| + \|A_j y - y'\|$ for some $y' \in F_j(y)$. Then, we obtain a new differential system

$$\begin{cases} \dot{z}(t) \leq Lz(t) + \epsilon\|y'\| \\ \dot{y} \in F_j(y) \end{cases}$$

The rest of the proof is similar to that of Proposition 1. Proposition 2 requires to choose a partition such that the difference in the vector fields between \mathcal{H} and the polyhedral inclusion induced by it is bounded by ϵ . This can be done by choosing a partition which ensures the ϵ bound for every A_i in \mathcal{H} . Proposition 3 is in fact captured using the uniformly converging condition in the theorem. \square

5. HEURISTICS FOR NON UNIFORM PREDICATE CONSTRUCTION

Next, we explore certain heuristics to construct the polyhedral partition for the hybridization procedure. A naive

algorithm is to uniformly partition the state-space and gradually increase the granularity of the partitions. This ignores the dynamics of the system in the selection of the partition. Instead, we propose a heuristic to refine the partition, such that the size of the resulting polyhedral inclusions are similar.

Here, we explain the heuristics for defining a non uniform predicate construction. Consider a polyhedral switched system $PolyH(\mathcal{H}, \mathcal{R})$, obtained by hybridization of the linear switched system \mathcal{H} with an initial set of predicates. We want to partition the state space in order to obtain a more accurate polyhedral hybrid system. For constructing a finer partition, we choose a location, (q, R) , in $PolyH(\mathcal{H}, R)$ and the polyhedral inclusion associated to such location, $P = CPoly(Flow(q), R)$. The target is to split region R . To split this region, we first compute the diameter of P , d_P , which corresponds to the maximum of $\|x - y\|$ while $x, y \in P$ and $\|x\| = \|y\| = 1$. The solution to the optimization problem returns the diameter value d_P and the two points x_P, y_P . We construct a vector v_P starting at x_P and finishing at y_P . This vector determines the direction in the dynamical polyhedron P where the longest error on approximating the linear dynamics arises. To overcome this problem, we construct a hyperplane which splits the dynamical polyhedron and it is perpendicular to that direction. The hyperplane is the set $\{x \in \mathbb{R}^n : v_P \cdot x = 0\}$. Observe that this hyperplane is not partitioning the state space but the polyhedral dynamics. Then, the hyperplane is transformed into a hyperplane in region R which will result in the polyhedral dynamics splitting. This hyperplane is the set $\{x \in R : v_P Ax = 0\}$, where A is the matrix for the linear dynamical system defining the vector field of location q in the initial linear hybrid system.

We need to split just regions that generate a coarse polyhedral inclusion dynamics. For choosing such regions we compare the diameter of each dynamical polyhedra with a diameter bound. This diameter bound is computed by considering every location (q, R) in $PolyH(\mathcal{H}, R)$ and the polyhedra $CUPoly(Flow(q), R)$, and it is defined as

$$D = \max_{(q, R) \in Loc} d_{CUPoly(Flow(q), R)}$$

The regions chosen are those with polyhedral inclusion dynamics whose polyhedral diameter is greater or equal to D or to some percentage of D . The higher the percentage, the less predicates will be added for partitioning the state space.

Example 1. We illustrate the hybridization process for a 2-dimensional linear hybrid system \mathcal{H} . The dynamical behaviour of the system is determined by two linear dynamical systems, $\dot{x} = A_1 x$ and $\dot{x} = A_2 x$, defined by the following matrices:

$$A_1 = \begin{pmatrix} 0 & 1 \\ -4 & 2 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0 & 1 \\ -3 & 2 \end{pmatrix}$$

We evaluate the difference between uniform and non uniform predicate construction methods. In the case of the uniform method, 64 predicates are needed to obtain stability answer for the polyhedral hybrid system $PolyH(\mathcal{H})$, while with the non uniform method, stability is verified by considering 40 predicates for the polyhedral hybrid system construction. We observe that the non uniform method provides a less

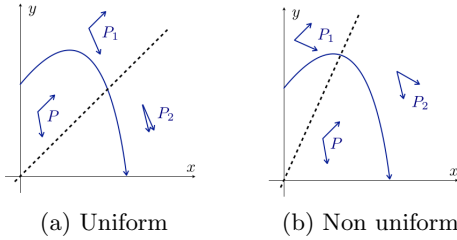


Figure 7: Predicate construction

number of predicates for verifying stability. The fewer quantity of predicates in comparison with the uniform predicate construction is because the choice for new predicates in the non uniform methods implies an homogeneous division of the dynamical polyhedra.

Let us consider the linear dynamical system defined by the matrix A_1 and restrict it to the polyhedral region $R = \{(x, y) \in \mathbb{R} : x \geq 0, y \geq 0\}$ which coincides with the first quadrant. We observe in Figure 7 a sample execution in R of the linear system $\dot{x} = A_1x$, and the dynamical polyhedron defined for such region and linear dynamics, denoted as P . For the case of uniform predicate construction we observe in Figure 7(a) the addition of the hyperplane $\{(x, y) \in \mathbb{R} : x - y = 0\}$, which results in a new partition with two regions, $\mathcal{R} \cap \{(x, y) \in \mathbb{R} : x - y \leq 0\}$ and $\mathcal{R} \cap \{(x, y) \in \mathbb{R} : x - y \geq 0\}$. For the first region we get a new dynamical polyhedron P_1 with diameter 2 while in the second region the new dynamical polyhedron P_2 has diameter 0.05. In the case of applying the heuristics explained above, we obtain the hyperplane $\{(x, y) \in \mathbb{R} : 2x - y = 0\}$ which partition R into two new subregions in an analogous manner to the uniform case. These subregions determine two new dynamical polyhedra, P_1 and P_2 with 1 as diameter value for both as shown in Figure 7(b). Observe that the case of non uniform partition results in a more homogeneous distribution of the error between linear and polyhedral dynamics.

6. EXPERIMENTAL RESULTS

In this section, we provide details regarding the implementation of the presented hybridization method, its integration into an abstraction based stability analysis approach for polyhedral hybrid systems, experimental set up, experimental evaluation and comparison with another stability verification tool based on Lyapunov function proofs.

The hybridization procedure is implemented in **Averist** [15]. **Averist** is a stability verification tool based on quantitative predicate abstraction. It has been implemented in Python and uses Parma Polyhedra Library (PPL) to deal with polyhedral operations and NetworkX Python package to manage graphs and their analysis. It performs stability analysis of polyhedral switched systems by constructing weighted graph abstractions, and returns a counterexample, a cycle with product of weights > 1 , in case it cannot establish stability.

The hybridization technique has been implemented in Python 2.7 and it uses the Parma Polyhedra Library (PPL) [2] for computing $CPoly(A, R)$ and the NetworkX Python package to represent the underlying graphs in linear and polyhedral hybrid automata. The experiments have been performed on

a virtual machine with OS Ubuntu 14.04, processor equivalent to 4 virtual CPUs and 4.68 GB of memory, installed on Mac OS X 10.10 with processor 2.8 GHz Intel Core i5 and 8GB 1600 MHz DDR3 memory.

These experiments are performed, first, to evaluate the scalability of **Averist**, in terms of state-space dimension, number of modes in the input hybrid automaton and predicates involved in hybridization, and second, to compare our general approach with a stability verification tool based on searching for Lyapunov functions.

6.1 Averists evaluation

We have constructed two kinds of input linear hybrid systems, those with arbitrary switching represented as AS_i and those with state based switching represented as SSi_j . The AS_i examples are chosen from [16] (Example 2) and the SSj_i examples are constructed from AS_i examples by restricting the switchings to happen at the boundaries of a partition with j regions. In particular, the set of executions of $SS8_i$ is a subset of the executions of $SS4_i$ which is a subset of the executions of AS_i . The 4 and 8 refer to the number of locations in the hybrid system.

The initial linear predicates used in partitioning the state-space for hybridization are taken to be the linear expressions extracted from the input system definition, the ones delimiting the invariants and guards. **Averist** runs iteratively with increasing number of linear predicates. We have implemented a function to choose the subsequent predicates, the predicates split the state-space into uniform sized chunks. For instance, for a two dimensional system, in addition to the constraints in the hybrid system, after the first iteration the predicates $x = 0$ and $y = 0$ are added, after the second iteration $x = y$ and $x = -y$ are added and so on.

A summary of experimental results is presented in Table 1. Every experiment has been run for a maximum time of 25 minutes as the time out. The first 6 columns consist of the **Averist** experiments. In addition to the dimension (number of continuous variables) and the experiment name, we report the number of iterations of the hybridization procedure that were run until stability was proven or the tool timed out, the number of linear expressions and the regions in the corresponding partition in the final hybridization, and the total time taken to infer Lyapunov stability. The total time includes the time taken by the underlying quantitative predicate abstraction technique to analyse the polyhedral hybrid system as well.

The experimental results in Table 1 demonstrate the feasibility of our approach for stability analysis of linear hybrid systems. Note that our tool was able to prove stability on almost all the 2D, 3D and 4D examples that we considered. The time for analysis increases reasonably slowly with the increase in the number of modes, for instance, consider the 4D systems $SS4_7$ and $SS8_7$ with 4 and 8 modes and with similar dynamics. However, the time increase with respect to increase in dimensions is steeper. This is expected because of the large number of regions in the partition as the dimension increases. We intend to explore compositional analysis methods to deal with the curse of dimensionality.

Also, note that since the running time depends on the number of regions, which in turn depends in an exponential manner on the number of linear expressions used in the partitioning, it is crucial to develop methods which are clever in the choice of the linear expressions. We will explore tech-

<i>Dimension/name</i>		Averist				Stabhyli		
		<i>Iterations</i>	<i>Expressions</i>	<i>Region</i>	<i>Time</i>	<i>Degree</i>	<i>LF found</i>	<i>Time</i>
2D	AS_1	5	32	129	31	6	Y	8
	SS4.1	1	2	9	< 1	8	–	452
	SS8.1	1	4	17	< 1	6	–	443
	SS16.1	1	8	33	1	4	–	177
	AS_2	5	32	129	34	6	Y	9
	SS4.2	1	2	9	< 1	8	–	418
	SS8.2	1	4	17	< 1	6	–	451
	AS_3	4	16	65	16	2	Y	< 1
	SS4.3	1	2	9	< 1	4	Y	7
SS8.3	1	4	17	1	6	–	417	
3D	AS_4	2	9	147	194	6	–	410
	SS4.4	3	21	771	484	2	Y	75
	SS8.4	3	21	771	470	2	Y	15
	SS16.4	3	21	771	568	2	Y	138
	AS_5	2	9	147	235	6	–	1254
	SS4.5	3	21	771	418	2	Y	2
	SS8.5	3	21	771	484	2	Y	18
	AS_6	2	9	147	220	6	–	1237
	SS4.6	3	21	771	463	2	Y	2
SS8.6	3	21	771	489	2	Y	17	
4D	AS_7	1	4	81	625	2	–	12
	SS4.7	1	4	81	119	2	–	101
	SS8.7	1	6	153	234	2	–	1071
	SS16.7	1	10	297	533	2	–	339
	AS_8	1	4	81	591	4	–	34
	SS4.8	1	4	81	117	4	–	397
	SS8.8	1	6	153	234	2	–	17
	AS_9	1	–	–	out	4	Y	34
	SS4.9	1	4	81	125	4	–	105
SS8.9	1	6	153	247	2	–	16	

Table 1: Comparison of Averist and Stabhyli

niques which dynamically partition the state-space in a non-uniform manner to circumvent this problem. Some ideas are presented in Section 5.

6.2 Comparison with Stabhyli

Next, we compare our method in **Averist** with **Stabhyli** [11], which is a tool for global stability verification based on Lyapunov function search that can handle the class of switched linear systems. Broadly, it searches iteratively with increasing degree of polynomial templates until it finds a solution. It only iterates over even degree polynomials, since the Lyapunov functions are required to be non-negative everywhere. In addition, it has the option for finding either common or multiple Lyapunov functions. We ran **Stabhyli** on our example set with both common or piecewise Lyapunov function options, in turn, and set a time out of 25 minutes as before. The experimental results of **Stabhyli** are summarized in the last three columns of Table 1. The second column from the end indicates whether the system proved stability (Y) or that it timed out (-) with the common Lyapunov function option. The third column from the end shows the maximum degree of the Lyapunov function in the common Lyapunov function case that **Stabhyli** successfully analyzed before the time out in the case of (-), or the degree of the polynomial Lyapunov function returned in the case of (Y). The last column shows the total time taken for solving the templates up to the degree indicated in the

first column. We were not able to prove stability on any of the examples with the multiple Lyapunov function option, hence, we do not report the details.

First, note that **Stabhyli** fails to prove stability on several examples, on which **Averist** succeeded. On *AS_9*, **Stabhyli** succeeded, where as **Averist** timed out. While we do not claim that the set of benchmarks here is representative, we observed several numerical issues with **Stabhyli** with respect to which we claim that our method and tool are well behaved. For instance, in *AS_1*, **Stabhyli** returned with a 6th degree Lyapunov function. We ran **Stabhyli** again with the same example by fixing the degree of the polynomial to be 8, and it returned with an answer that stability could not be proved. This is surprising, and we expect it to be a result of numerical issues which cannot set the coefficient of degree 8 term in the polynomial to 0 exactly. Note that this implies that the iteration over the templates needs to be done in a linear fashion. Next, while it found a common Lyapunov function, it did not succeed in finding a multiple Lyapunov function (note that the same Lyapunov function in each mode is a multiple Lyapunov function). In addition, on the systems *SS4.1* and *SS8.1*, **Stabhyli** did not succeed in proving stability which consist of fewer executions than *AS_1*.

On the examples, where **Stabhyli** finds a Lyapunov function, it does so quickly. On all examples, except *SS4.3*,

Stabhyli performs better. However, the time required in checking whether a template is a Lyapunov function, increases much faster with the increase in the degree, as compared to, for instance, with the increase in the number of predicates for *Averist*. For instance, in the 4D example *AS_7*, it took 12 seconds to check the polynomial of degree 2, but timed out for the polynomial with degree 4.

To summarize, we believe that the strength of our approach and our tool is that it is less prone to numerical problems, since the tool mainly uses a linear optimization solver at the backend with rational arithmetic. In addition, our tool behaves in a monotonic manner with respect to simpler systems. Finally, it returns counterexamples when it fails to prove stability. We plan to utilize this to devise smarter refinement algorithms in the future.

7. CONCLUSION

In this paper, we proposed a new hybridization approach for stability verification of linear hybrid automata. Our experimental results show the practical feasibility of our approach and exhibit the merits in comparison to existing approaches. Also, our approach returns a counterexample which provides information on the possible cause of instability, that can be used to guide the choice of subsequent partitions. There are several interesting future directions. One direction is to perform an indepth analysis of the class of systems for which the method is complete, that is, will eventually prove stability if the system is stable. The other direction is to explore this approach for non-linear hybrid systems. While theoretically, extending the approach to the general class of hybrid systems is straightforward, the practical challenge is in finding the right class of partitions to consider for hybridization.

8. REFERENCES

- [1] E. Asarin, T. Dang, and A. Girard. Hybridization methods for the analysis of nonlinear systems. *Acta Informatica*, 43(7):451–476, 2007.
- [2] R. Bagnara, P. M. Hill, and E. Zaffanella. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 72(1–2):3–21, 2008.
- [3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.
- [4] M.S. Branicky. Stability of hybrid systems. In H. Unbehauen, editor, *Encyclopedia of Life Support Systems*, volume Theme 6.43:Control Systems, Robotics and Automation, chapter Article 6.43.28.3. UNESCO Publishing, 2004.
- [5] T. Dang, O. Maler, and R. Testylier. Accurate hybridization of nonlinear systems. In *Proceedings of the International Conference on Hybrid Systems: Computation and Control*, pages 11–20, 2010.
- [6] Laurent Doyen, Thomas A. Henzinger, and Jean-françois Raskin. Automatic rectangular refinement of affine hybrid systems. In *Proceedings of Formal Modeling and Analysis of Timed Systems*, pages 144–161. Springer, 2005.
- [7] S. Graf and H. Saidi. Construction of abstract state graphs with PVS. In *Proceedings of the International Conference on Computer Aided Verification*, pages 72–83, 1997.
- [8] Morris William Hirsch, Stephen Smale, and Robert Luke Devaney. *Differential equations, dynamical systems, and an introduction to chaos*. Academic Press, Waltham (Mass.), 2013.
- [9] Daniel Liberzon, Joao P. Hespanha, and A. Stephen Morse. Stability of switched systems: a lie-algebraic condition. *Systems Control Lett*, 37:117–122, 1999.
- [10] Hai Lin and Panos J. Antsaklis. Stability and stabilizability of switched linear systems: A survey of recent results. *IEEE Transactions on Automatic Control*, 54(2):308–322, 2009.
- [11] Eike Möhlmann and Oliver E. Theel. Stabhyli: a tool for automatic stability verification of non-linear hybrid systems. In *HSCC*, pages 107–112, 2013.
- [12] P. A. Parrilo. *Structure Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, Pasadena, CA, May 2000., 2000.
- [13] Pavithra Prabhakar, Geir E. Dullerud, and Mahesh Viswanathan. Pre-orders for reasoning about stability. In *Proceedings of the International Conference on Hybrid Systems: Computation and Control*, pages 197–206, 2012.
- [14] Pavithra Prabhakar and Miriam Garcia Soto. Abstraction based model-checking of stability of hybrid systems. In *CAV*, pages 280–295, 2013.
- [15] Pavithra Prabhakar and Miriam Garcia Soto. An algorithmic approach to stability verification of polyhedral switched system. In *American Control Conference*, 2014.
- [16] S. Prajna and A. Papachristodoulou. Analysis of switched and hybrid systems - beyond piecewise quadratic methods. In *American Control Conference, 2003. Proceedings of the 2003*, volume 4, pages 2779–2784 vol.4, June 2003.
- [17] A. Puri, V.S. Borkar, and P. Varaiya. ϵ -approximation of differential inclusions. In *Proceedings of the International Conference on Hybrid Systems: Computation and Control*, pages 362–376, 1995.