Foundations of Quantitative Predicate Abstraction for Stability Analysis of Hybrid Systems

Pavithra Prabhakar and Miriam García Soto

IMDEA Software Institute, Madrid, Spain,
pavithra.prabhakar@imdea.org,miriam.garcia@imdea.org

Abstract. We investigate the formal connections between "quantitative predicate abstractions" for stability analysis of hybrid systems and "continuous simulation relations". It has been shown recently that stability is not bisimulation invariant, and hence, stronger notions which extend the classical simulation and bisimulation relations with continuity constraints have been proposed, which force preservation of stability. In another direction, a quantitative version of classical predicate abstraction has been proposed for approximation based stability analysis of certain classes of hybrid systems. In this paper, first, we present a general framework for quantitative predicate abstraction for stability analysis. We then show that this technique can be interpreted as constructing a one dimensional system which continuously simulates the original system. This induces an ordering on the class of abstract systems and hence, formalizes the notion of refinement.

1 Introduction

Hybrid systems refer to systems which consist of mixed discrete continuous behaviors. They manifest in embedded control systems, which typically consist of one or more embedded processors controlling physical entities. Stability is a fundamental property in control system design. Intuitively, stability captures the notion that small perturbations to the initial state or input to a system result in only small variations in the behavior of the system. In this paper, we investigate the formal foundations for an abstraction based analysis approach for stability analysis of hybrid systems.

The classical approach to stability analysis in control theory is based on Lyapunov functions (see, for instance, [14]). Here, stability of a continuous dynamical system is established by exhibiting a Lyapunov function - a continuously differentiable function on the state-space such that its value is zero at the equilibrium point and positive everywhere else, and the value of the function decreases along any execution of the system. A Lyapunov function is analogous to the ranking function for proving termination of discrete programs [6]. The approach has been extended to hybrid systems in the form of common and multiple Lyapunov functions [27, 9, 15]. Automated analysis involves starting with a template which

serves as a candidate Lyapunov function, and then using constraint/optimization solvers to deduce the unknown parameters of the template. For instance, for a polynomial template with coefficients as parameters, the requirements of Lyapunov function can be encoded as a sum-of-squares programming problem, which can be efficiently solved using tools such as SOSTOOLS [20, 19, 18]. One of the major limiting factors of this approach is the ingenuity required in providing the right templates; and automatically learning the templates is a challenge which has not been adequately addressed (except for some recent work [13]). Moreover, if a template fails to satisfy the conditions of Lyapunov function, then it typically does not provide insights into the potential reasons for instability or towards the choice of better templates for succeeding iterations.

To overcome some of the limitations of template based search, an alternate approach based on abstractions has been investigated [23, 24]. However, the development of such an approach is not straightforward. Simulations and bisimulations [17] are the foundational basis for abstraction and minimization based analysis. Recent results [21, 22] show that stability is not bisimulation invariant, and a simulation relation between two systems does not suffice to preserve stability. A stronger notion that extends stability with continuity constraints is proposed and shown to preserve stability. These negative results suggest that traditional abstraction techniques will need to be modified for stability analysis.

In [23, 24], a quantitative version of predicate abstraction was proposed for stability analysis. Recall that predicate abstraction [10] constructs a finite graph which simulates a given system. The finite graph is obtained by partitioning the state-space of the system into a finite number of regions using a finite set of predicates. The regions correspond to the nodes of the graph and an edge between two nodes indicates the possibility of an execution starting from the region corresponding to the source of the edge to the region corresponding to its target. Predicate abstraction has been widely applied for safety verification in the context of both discrete and hybrid systems [5, 2, 3, 28]. However, the finite graph does not provide useful information towards deciding the stability of the system. Hence, in [23, 24], a modified abstraction procedure is proposed, which annotates the finite graph with quantitative information for the purpose of stability analysis. The edges of the graph are annotated with a weight which captures the ratio of the distance to the origin of final state to that of the initial state, of the executions corresponding to the edge. Then stability is inferred by analyzing certain structural properties about the graph, such as, the absence of cycles with the product of weights on its edges greater than 1.

In this paper, we investigate the formal foundations for the quantitative predicate abstraction proposed in [23, 24]. First, we present a general framework for quantitative predicate abstraction and identify conditions on the hybrid system and the predicates for which the approach is sound. Next, we establish a formal connection between the abstract weighted graph and the concrete hybrid system using the notion of continuous simulations. For this, we interpret a weighted graph as representing a one-dimensional hybrid system whose executions follow the edges in the graph and satisfy the weight constraints on them. We show

that the one-dimensional hybrid system representing the weighted graph "continuously simulates" the concrete hybrid system from which the graph is constructed. This establishes a partial ordering on the abstract weighted graphs, and formalizes the notion of refinement.

2 Preliminaries

Sets of numbers. Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$ and $\mathbb{N}$ denote the set of real numbers, non-negative real numbers and natural numbers, respectively. We use $[n]$ to denote the set $\{0, \cdots, n\}$. We use a superscript ∞ to indicate that ∞ is included in the set. For example, $\mathbb{R}^{\infty}_{\geq 0}$ denotes the set $\mathbb{R}_{\geq 0} \cup \{\infty\}$. Given a subset $I \subseteq \mathbb{R}$, $last(I)$ denotes the least upper bound of I in $\mathbb{R}^{\infty}$.

Euclidean space $\mathbb{R}^n$. Given $x \in \mathbb{R}^n$, let $(x)_i$ denote the i-th component of x. Let $\|x\|$ denote the Euclidean norm of x, that is, $[\sum_i (x)_i^2]^{1/2}$. Given $\epsilon \geq 0$ and $x \in \mathbb{R}^n$, $B_\epsilon(x)$ denotes the open ball of radius ϵ around x, that is, $B_\epsilon(x) = \{y \,|\, \|x - y\| < \epsilon\}$. Given a finite set Q, we extend the metric on $\mathbb{R}^n$ to an extended pseudo-metric on $Q \times \mathbb{R}^n$ as follows: The distance between $(q_1, x_2), (q_2, x_2) \in Q \times \mathbb{R}^n$, denoted $\|(q_1, x_1) - (q_2, x_2)\|$, is given by, $\|x_1 - x_2\|$. Further, $\|(q, x)\| = \|x\|$ will denote the norm of (q, x).

Functions. Let $dom(f)$ denote the domain of a function f. Given a function $f : A \to B$, and a set $A' \subseteq A$, we use $f(A')$ to denote the set $\{b \,|\, \exists a \in A', f(a) = b\}$. For an element $b \in B$, the inverse of f at b, denoted $f^{-1}(b)$, is the set $\{a \in A : f(a) = b\}$. Given a function $f : A \to B$, where A is equipped with a total ordering with a least element 0 and a difference operator ($a - b$ when $a > b$), we define f_t and f^t to be the function f restricted to the domain up to t and to the domain starting from t. More precisely, f_t is the function with domain $\{t' \in A \,|\, t' \leq t\}$ and $f_t(t') = f(t)$ for all $t' \in dom(f^t)$. Similarly, f^t is the function with domain $\{t' \geq 0 \,|\, \exists t'' \in A, t'' \geq t, t'' - t = t'\}$ and $f^t(t') = f(t'')$, where $t'' - t' = t$, for all $t' \in dom(f^t)$.

Set-valued function. A *set-valued function* $R : A \rightsquigarrow B$ is a function which maps every element of A to a set of elements in B. Given $A' \subseteq A$, $R(A') = \cup_{a \in A'} R(a)$. Every relation $R \subseteq A \times B$ can be interpreted as a set-valued function from A to B, where for any $a \in A$, $R(a) = \{b \,|\, (a, b) \in R\}$. We interchangeably use R to represent both the relation and the set-valued function it represents. The inverse of R, denoted R^{-1}, is the set $\{(b, a) \,|\, (a, b) \in R\}$.

A set-valued function $R : A \rightsquigarrow B$ is said to be *continuous* at a point $a \in A$ if

$$\forall \epsilon > 0, \exists \delta > 0 \,\text{such that}\, R(B_\delta(a)) \subseteq B_\epsilon(R(a)).$$

Sequences. A sequence over a set A is a function $S : D \to A$, where $D = [n]$ for some n, or $D = \mathbb{N}$. The size of the sequence S, denoted $|S|$, is n if $D = [n]$, in which case S is said to be a finite sequence, and ∞, otherwise. We also represent S by enumerating its elements as in $S(0), S(1), \ldots$.

Graphs. A *graph* G is a triple (V, L, E), where V is a finite set of vertices, L a finite set of labels and $E \subseteq V \times L \times V$ is a finite set of edges. A *path* of a graph is a finite or infinite sequence of vertices and edges $\pi = v_0 e_0 v_1 e_1 \ldots$. A *cycle* is a finite path where the first and the last vertices are the same; and it is *simple* if all the vertices except the last are distinct.

A weighted function is an extension of a graph with a weighting function on the edges. A *weighted graph*$G = (V, L, E, W)$ where (V, L, E) is a graph and $W : E \to \mathbb{R}^{\infty}_{\geq 0}$ is a weighting function. The weight of a finite path π is the product of the weights on the edges. Hence, given $\pi = v_0 e_0 v_1 e_1 \ldots e_n v_n$, $W(\pi) = \Pi_{i=0}^{n} W(e_i)$. The maximum weight value of the graph, denoted $MW(G)$, is $\max_{e \in E} W(e)$.

Linear expressions, homogeneity. A *linear expression* is an expression of the form $a \cdot x + b$, where $a \in \mathbb{R}^n$, x is a tuple of n-variables and $b \in \mathbb{R}$; and it is called *homogeneous* if b is the zero vector. Given a linear expression $\eta := a \cdot x + b$, it defines a function $[\![\eta]\!] : \mathbb{R}^n \to \mathbb{R}$ where given a valuation $v \in \mathbb{R}^n$, $[\![\eta]\!]$ maps it to the value $a \cdot v + b$. A *linear constraint or predicate* c is given by $\eta \sim 0$, where η is a linear expression and $\sim$ is a relational operator in $\{<, \leqslant, =\}$. Let $[\![c]\!]$ denote the set of all $v \in \mathbb{R}^n$ such that $v \in [\![\eta]\!]$, $v \sim 0$, where c is given by $\eta \sim 0$. Given a set of linear constraints C, it defines the set $P = \cap_{c \in C} [\![c]\!]$ denoted $[\![C]\!]$. A *convex polyhedral* set is a set defined by a finite set of linear constraints C.

Polyhedral partition. A partition $\mathcal{P}$ of $\mathbb{R}^n$ into convex polyhedral sets is a finite set of convex polyhedral sets $\{P_1, \ldots, P_k\}$ such that $\cup_{i=1}^{k} P_i = \mathbb{R}^n$ and for each $i \neq j$, $P_i \cap P_j = \emptyset$.

3 Hybrid Systems

In this section, we present a semantic model for hybrid systems. We then define a concrete class of hybrid system, namely, piecewise linear dynamical systems, which we use in the sequel to illustrate the theoretical concepts.

3.1 A semantic definition of hybrid systems

Hybrid systems are systems exhibiting mixed discrete and continuous behaviors. We present a semantic model of a hybrid system as consisting of discrete transitions and continuous trajectories. For a concrete specification formalism, see the hybrid automaton model [1, 11]. Let us fix a finite set Q and a set $X = \mathbb{R}^n$, for some n. Given an element $(q, x) \in Q \times X$, $[q, x]_D = q$ and $[q, x]_C = x$.

Trajectories. A *trajectory* over (Q, X) is a function $\tau : I \to Q \times X$, where I is either $[0, T]$ for some $T \in \mathbb{R}_{\geq 0}$ or $[0, \infty)$, such that $[\tau]_D$ is finitely varying ($[\tau]_D$ restricted upto time t has finite number of discontinuities for any $t \in [0, T]$) and $[\tau]_C$ is a continuous function. We denote the set of all trajectories over (Q, X) by *Traj*(Q, X).

The last time of a trajectory τ, $\mathit{ltime}(\tau)$, is $\mathit{last}(\mathit{dom}(\tau))$. The first state of the trajectory τ, denote $\mathit{fstate}(\tau)$, is $\tau(0)$, and if $\mathit{ltime}(\tau) < \infty$, then the last state of τ, denoted $\mathit{lstate}(\tau)$, is $\tau(\mathit{ltime}(\tau))$. The set of states of τ, denoted $\mathit{States}(\tau)$, is the set $\{\tau(t) \mid t \in \mathit{dom}(\tau)\}$. Given a time $t \in \mathit{dom}(\tau)$, the prefix of τ up to time t is the trajectory τ_t and the suffix of τ from time t is τ^t.

Transitions. A *transition* over a pair (Q, X) is a pair $\iota = ((q_1, x_1), (q_2, x_2)) \in (Q \times X) \times (Q \times X)$. We denote the set of all transitions over (Q, X) by $\mathit{Trans}(Q, X)$. For a transition $\iota = ((q_1, x_1), (q_2, x_2))$, $\mathit{ltime}(\iota) = 0$, $\mathit{fstate}(\iota) = (q_1, x_1)$, $\mathit{lstate}(\iota) = (q_2, x_2)$ and $\mathit{States}(\iota) = \{(q_1, x_1), (q_2, x_2)\}$.

Hybrid system definition. A *hybrid system* $\mathcal{H}$ is a tuple (Q, X, Σ, Δ), where:

- Q is a finite set of control locations;
- $X = \mathbb{R}^n$, for some n, is the continuous state-space;
- $\Sigma \subseteq \mathit{Trans}(Q, X)$ is a set of transitions; and
- $\Delta \subseteq \mathit{Traj}(Q, X)$ is a set of trajectories.

The dimension of $\mathcal{H}$ is n and the state-space, $\mathit{States}(\mathcal{H})$, is $Q \times X$.

Executions. An execution of a hybrid system is a finite or infinite sequence of transitions and trajectories. An *execution* of a hybrid system $\mathcal{H}$ is a sequence $\sigma : D \to \Sigma \cup \Delta$, such that for all $0 \leqslant i < |\sigma|$, $\mathit{lstate}(\sigma(i)) = \mathit{fstate}(\sigma(i+1))$, and if σ is an infinite sequence then $\sum_{i:\sigma(i)\in\Delta} \mathit{last}(\mathit{dom}(\sigma(i))) = \infty$. Let $\mathit{Exec}(\mathcal{H})$ denote the set of all executions of $\mathcal{H}$.

Let $\mathit{fstate}(\sigma) = \mathit{fstate}(\sigma(0))$ and if $|\sigma| < \infty$ and $\mathit{last}(\sigma(|\sigma|)) < \infty$, then $\mathit{lstate}(\sigma) = \mathit{lstate}(\sigma(|\sigma|))$. Let $\mathit{States}(\sigma) = \cup_{i\in dom(\sigma)}\ \mathit{States}(\sigma(i))$.

Hybrid time domain. We define a hybrid time domain for an execution, so that we can interpret the execution as a function from this domain to the states of the hybrid system. Given an execution $\sigma : D \to \Sigma \cup \Delta$, the hybrid time domain of σ, denoted $\mathit{htd}(\sigma)$, is the set $\{(i, 0) \mid i \in \mathit{dom}(\sigma), \sigma(i) \in \Sigma\} \cup \{(i, t) \mid i \in \mathit{dom}(\sigma), \sigma(i) \in \Delta, t \in \mathit{dom}(\sigma(i))\}$. The execution σ can be represented as a function f_σ from $\mathit{htd}(\sigma)$ to $\mathit{States}(\mathcal{H})$, where for $(i, t) \in \mathit{htd}(\sigma)$, $f_\sigma(i, t) = \mathit{fstate}(\sigma(i))$ if $\sigma(i) \in \Sigma$, and $\sigma(i)(t)$ otherwise. Note that there is a bijection from the set of executions to the functions they represent. Given two points (i_1, t_1) and (i_2, t_2) in a hybrid time domain, we define an ordering between them as $(i_1, t_1) < (i_1, t_2)$ if $i_1 < i_2$, or $i_1 = i_2$ and $t_1 < t_2$. Also, if $(i_2, t_2) > (i_1, t_1)$, then $(i_2, t_2) - (i_1, t_1) = (i_2 - i_1, t)$, where $t = t_2 - t_1$ if $i_1 \neq i_2$, and t_2, otherwise. We then denote by $\sigma_{(i,t)}$ and $\sigma^{(i,t)}$, prefix of σ up to (i, t) and suffix of σ from (i, t), respectively. $\sigma_{(i,t)}$ is given by the function $(f_\sigma)_{(i,t)}$ and $\sigma^{(i,t)}$ is given by the function $(f_\sigma)^{(i,t)}$.

Splitting trajectories and executions. We say that (τ_1, τ_2) is a *splitting* of a *trajectory* τ, denoted $\tau = \tau_1 \circ \tau_2$, if there exists $t \in \mathit{dom}(\tau)$ such that $\tau_1 = \tau_t$ and $\tau_2 = \tau^t$. Similarly, (σ_1, σ_2) is a *splitting* of an *execution* σ, denoted $\sigma = \sigma_1 \circ \sigma_2$, if there exists an $(i, t) \in \mathit{htd}(\sigma)$ such that $\sigma_1 = \sigma_{(i,t)}$ and $\sigma_2 = \sigma^{(i,t)}$. Note that

splitting is associative that is $\sigma = (\sigma_1 \circ \sigma_2) \circ \sigma_3$ if and only if $\sigma = \sigma_1 \circ (\sigma_2 \circ \sigma_3)$. Hence, for a splitting of σ or τ into n parts, we do not need to specify the splitting order. Further, we write $\sigma = \sigma_1 \circ \sigma_2 \circ \ldots$ to denote a splitting of σ into infinitely many parts, that is, there exist $\sigma'_1, \sigma'_2, \ldots$, such that $\sigma = \sigma_1 \circ \sigma'_1$ and for $i \geq 1$, $\sigma'_i = \sigma_{i+1} \circ \sigma'_{i+1}$.

3.2 Illustration using piecewise linear dynamical systems

Next, we instantiate the semantic model with a concrete class of hybrid systems, namely, piecewise linear dynamical systems. These are systems in which the state-space is partitioned into a finite set of convex polyhedral sets, each of which is associated with a linear dynamical system.

Definition 1. *An n-dimensional* piecewise linear dynamical system *(PLDS)* $\mathcal{M}$ *is a pair* $(\mathcal{P}, F)$*, where* $\mathcal{P}$ *is a finite partition of* $\mathbb{R}^n$ *into convex polyhedral sets and* $F : \mathcal{P} \to \mathbb{R}^{n \times n}$ *is a function associating an* $n \times n$ *matrix with every element of the partition.*

An n-dimensional *PLDS*, $\mathcal{M} = (\mathcal{P}, F)$, is represented as a hybrid system with the tuple (Q, X, Σ, Δ), where

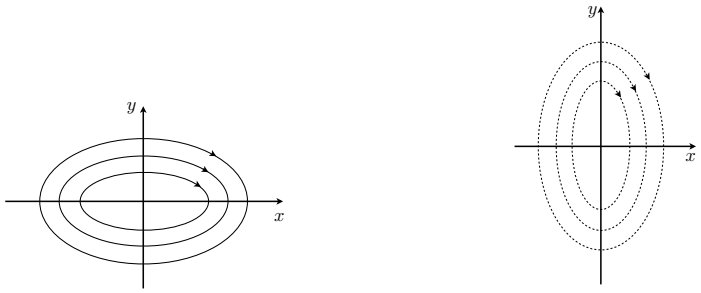
- the control location set Q is equal to the partition $\mathcal{P}$,
- the continuous state-space X is equal to $\mathbb{R}^n$,
- the set of transitions Σ is contained in $\{((P_1, x), (P_2, x)) \in (Q \times X) \times (Q \times X) : P_1 \neq P_2,\ \mathit{Closure}(P_1) \cap \mathit{Closure}(P_2) \neq \emptyset\}$ and
- the set of trajectories Δ includes every $\tau : I \to \mathcal{P} \times \mathbb{R}^n$ such that there exists $P \in \mathcal{P}$ with $[\tau]_D(t) = P$ and $[\dot{\tau}]_C = F(P) \cdot [\tau]_C$ for all $t \in \mathit{dom}(\tau)$.

Example 1. Consider the following linear dynamical systems:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -0.1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \text{ and } \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -4 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

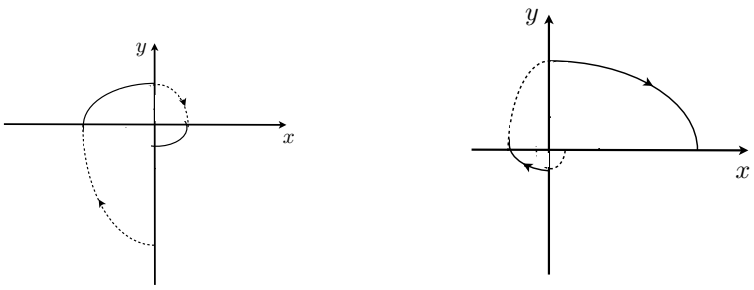
where $x = x(t)$ and $y = y(t)$. Let us call the matrices A and B, respectively. The phase portraits for the systems are shown in Figure 1.

Next, we define two piecewise linear dynamical systems $\mathcal{M}_1$ and $\mathcal{M}_2$, where $\mathcal{M}_1$ follows the dynamics associated with B in the positive quadrant and the quadrant diagonally opposite to it, and the dynamics A in the other two quadrants. $\mathcal{M}_2$ follows A in the quadrants in which $\mathcal{M}_1$ follows B, and follows B in the quadrants in which $\mathcal{M}_1$ follows A. A sample of execution for the systems $\mathcal{M}_1$ and $\mathcal{M}_2$ is depicted in Figure 2(a) and 2(b), respectively. Each of the executions consist of four trajectories each of which belongs to a particular location (a quadrant), and discrete transitions which change locations at the boundaries of the quadrants.



(a) $\dot{x} = Ax$ (b) $\dot{x} = Bx$

Fig. 1. Phase portraits



(a) System $\mathcal{M}_1$ (b) System $\mathcal{M}_2$

Fig. 2. Sample executions

4 Lyapunov and Asymptotic Stability

In this section, we define two classical notions of stability in control theory for hybrid systems, namely, Lyapunov and asymptotic stability. We will focus on stability with respect to an equilibrium point. For simplicity of presentation, we consider the origin in the continuous state-space of the hybrid system to be the equilibrium point. For an n-dimensional hybrid system $\mathcal{H}$, we use $0_{\mathcal{H}}$ to denote the origin $\bar{0} \in \mathbb{R}^n$.

Lyapunov stability captures the notion that small perturbations in the initial state of the system result in only small perturbations of the eventual behaviors.

Definition 2. *A set of executions $S \subseteq Exec(\mathcal{H})$ is said to be* Lyapunov stable *if for every $\epsilon > 0$, there exists a $\delta > 0$ such that for every execution $\sigma \in S$ with $[fstate(\sigma)]_C \in B_\delta(0_{\mathcal{H}})$, $[States(\sigma)]_C \subseteq B_\epsilon(0_{\mathcal{H}})$.*

A hybrid system $\mathcal{H}$ is said to be Lyapunov stable, if $Exec(\mathcal{H})$ is Lyapunov stable.

Asymptotic stability requires convergence in addition to Lyapunov stability. An execution σ of $\mathcal{H}$ is said to *converge* to $0_{\mathcal{H}}$, denoted $Conv(\sigma, 0_{\mathcal{H}})$, if for every $\epsilon > 0$, there exists a pair $(i,t) \in htd(\sigma)$ such that $[States(\sigma_{(i,t)})]_C \subseteq B_\epsilon(0_{\mathcal{H}})$.

Definition 3. *A set of executions $S \subseteq Exec(\mathcal{H})$ is said to be* asymptotically stable *if it is Lyapunov stable and there exists a $\delta > 0$ such that every $\sigma \in S$ with $[fstate(\sigma)]_C \in B_\delta(0_{\mathcal{H}})$, $Conv(\sigma, 0_{\mathcal{H}})$ holds.*

A hybrid system $\mathcal{H}$ is said to be asymptotically stable if $Exec(\mathcal{H})$ is asymptotically stable.

In Example 1, the dynamics of the linear systems of Figure 1(a) and 1(b) describe executions moving along an ellipsoid around the origin, the equilibrium point. Both systems are Lyapunov stable, since the executions remain close to the equilibrium point when they start close to the equilibrium point. For *PLDS* $\mathcal{M}_1$ depicted in Figure 2(a), the executions eventually approach the equilibrium point, hence, $\mathcal{M}_1$ is asymptotically stable. On the other hand, the system $\mathcal{M}_2$ exhibits instability, since its executions, represented in Figure 2(b), diverge with respect to the equilibrium point.

5 Quantitative Predicate Abstraction

In this section, we present a quantitative predicate abstraction technique for analyzing stability of hybrid systems, which generalizes the abstraction techniques in [23] and [24] for the class of piecewise constant derivative systems and polyhedral switched systems, respectively. In particular, we identify a condition on the interaction between the hybrid system and the predicates used in the abstraction, which renders the method sound. We illustrate the approach on the class of piecewise linear dynamical systems.

5.1 Weighted graphs as quantitative abstractions

In the context of safety verification, a finite abstraction of a concrete system is constructed from a partition of the state space of the system into a finite number of regions. The nodes in the finite abstraction correspond to the regions and the edges between two nodes capture the existence of an execution in the concrete system starting from a state in the region corresponding to the first node to a state in the region corresponding to the second node. This defines an abstract system, a finite graph, which over-approximates the behaviors of the concrete system, and hence, safety of the abstract system implies the safety of the concrete system.

However, for stability verification, it does not suffice to merely construct a system which over-approximates the behaviors of the concrete system. We need to capture some quantitative information about the evolution of the distance of the states to the origin along an execution. Hence, we annotate the finite graph with weights. More precisely, we interpret the nodes in the abstract graph as regions, an edge in the graph as the existence of a potential execution from one

region to other evolving through a third region, and the weights as the scaling in the distance to the origin of the execution as it traverses from the first region to the second one.

We need some auxiliary constructs in the construction of the weighted graph. Let $\mathcal{H} = (Q, X, \Sigma, \Delta)$ be a hybrid system and $P_1, P_2, P \subseteq \mathit{States}(\mathcal{H})$. We define a predicate which represents pairs of states (s_1, s_2) such that there exists a trajectory which enters P through P_1 at s_1, remains in P for sometime and exits P through P_2 at s_2. More precisely,

$$\mathit{ReachRel}^C_{\mathcal{H}}(P_1, P, P_2) := \{(s_1, s_2) \in P_1 \times P_2 \mid \exists \tau \in \Delta : \mathit{fstate}(\tau) = s_1,$$

$$\mathit{lstate}(\tau) = s_2 \text{ and } \tau(t) \in P \text{ for all } 0 < t < \mathit{ltime}(\tau)\}.$$

Also, we define a predicate containing the pairs of states (s_1, s_2) such that there exists a transition from P_1 to P_2 where s_1 is contained in P_1 and s_2 in P_2, it is

$$\mathit{ReachRel}^D_{\mathcal{H}}(P_1, P_2) := \{(s_1, s_2) \in P_1 \times P_2 \mid \exists \iota = (s_1, s_2) \in \Sigma\}.$$

Definition 4. *A weighted graph $G = (V, V \cup \{\gamma\}, E, W)$ is a* quantitative abstraction *of a hybrid system $\mathcal{H}$ with respect to an abstraction function $\alpha : \mathit{States}(\mathcal{H}) \to V$ if the following hold. Given $v_1, v_2 \in V$, define*

$$Z^C(v_1, v, v_2) = \mathit{ReachRel}^C_{\mathcal{H}}(\alpha^{-1}(v_1), \alpha^{-1}(v), \alpha^{-1}(v_2)).$$

$$Z^D(v_1, v_2) = \mathit{ReachRel}^D_{\mathcal{H}}(\alpha^{-1}(v_1), \alpha^{-1}(v_2)).$$

- *Edge condition: For every $v_1, v_2 \in V$,*

$$Z^C(v_1, v, v_2) \neq \emptyset \Rightarrow (v_1, v, v_2) \in E, Z^D(v_1, v_2) \neq \emptyset \Rightarrow (v_1, \gamma, v_2) \in E.$$

- *Weight conditions:*
  - *For every edge $e = (v_1, v, v_2)$.*

$$v \neq \gamma \Rightarrow \sup_{(s_1, s_2) \in Z^C(v_1, v, v_2)} \frac{\|s_2\|}{\|s_1\|} \leq W(e).$$

$$v = \gamma \Rightarrow \sup_{(s_1, s_2) \in Z^D(v_1, v_2)} \frac{\|s_2\|}{\|s_1\|} \leq W(e).$$

Note that even when α is fixed, there are several weighted graphs quantitatively abstracting the concrete system. However, there is a minimal graph which quantitatively abstracts the concrete system with respect to a given α.

Definition 5. *A* minimal quantitative abstraction *G of a hybrid system $\mathcal{H}$ with respect to an abstraction function α satisfies the implication on the edge conditions and the inequality in the weight conditions in both directions.*

Next, we identify a condition on the abstraction function α and the hybrid system $\mathcal{H}$ which will ensure that the abstract graph captures all the executions of $\mathcal{H}$.

Definition 6. *A hybrid system $\mathcal{H}$ is well-behaved with respect to an abstraction function $\alpha : States(\mathcal{H}) \to V$ if for every continuous trajectory τ of $\mathcal{H}$, the function $\alpha \circ \tau$ is finitely varying on V.*

From now on, we assume that the following assumption holds.

Assumption 1 *The hybrid system is well-behaved with respect to the choice of the quantitative predicate abstraction.*

The following theorem provides efficiently verifiable conditions on the abstract weighted graph which imply stability of the concrete system.

Theorem 1. *Let $G = (V, L, E, W)$ be a quantitative abstraction of a hybrid system $\mathcal{H}$ which satisfies Assumption 1. Consider the following conditions:*

G1 there is no edge e in G with infinite weight, that is, $W(e) < +\infty, \forall e \in E$,
G2 every simple cycle π of G satisfies $W(\pi) \leqslant 1$ and
G3 every simple cycle π of G satisfies $W(\pi) < 1$.

Then:

- *$\mathcal{H}$ is Lyapunov stable if conditions G1 and G2 hold and*
- *$\mathcal{H}$ is asymptotically stable if conditions G1 and G3 hold.*

We defer the proof of Theorem 1 to Section 6.3. Once we establish a connection between quantitative abstractions and continuous simulations, the proof of Theorem 1 is straightforward. We briefly explain the motivation for the conditions $G1 - G3$ in the theorem. For every execution of the hybrid system, there is a path in the graph such that the weights on the path provide an upper bound on how far the execution deviates with respect to the origin. Condition $G1$ states that the executions which eventually remain within a particular region do not diverge; while Conditions $G2$ (and $G3$) capture the fact that the executions which switch between regions infinitely often do not diverge (do converge).

Remark 1. One of the main hightlights of the quantitative abstraction based stability analysis is that the method returns a counter-example in the event of a failure, indicating a potential reason for instability. For instance, a cycle of weight greater than one in the weighted graph expresses the possible existence of an infinite diverging execution.

5.2 Illustration on *PLDS*

In this section, we illustrate the quantitative abstraction based stability analysis on the class of piecewise linear dynamical systems. We use as the abstraction function a polyhedral partition of the state-space, and show that piecewise linear dyanmical systems are well-behaved with respect to the polyhedral partition. We then illustrate the abstraction procedure on a simple example.

Let us fix an n-dimensional *PLDS* $\mathcal{M} = (\mathcal{P}, F)$. Recall that for this class of hybrid systems, the control location set is the partition $\mathcal{P}$ and the continuous state-space is $\mathbb{R}^n$. A state is represented as $(P, x) \in \mathcal{P} \times \mathbb{R}^n$. Fix a set of predicates on $\mathbb{R}^n$, which results in a partition $\mathcal{P}'$. The abstraction function is then given by $\alpha_{\mathcal{P},\mathcal{P}'}((P, x)) = (P, P')$, where $x \in P'$ and $P' \in \mathcal{P}'$. Next, we observe that a *PLDS* is well-behaved with respect to the abstraction function defined above, hence Assumption 1 holds.

Proposition 1. *Given a square matrix $A \in \mathbb{R}^n$ and a variable $t \in \mathbb{R}$, the exponential matrix e^{At} is a square matrix whose elements are linear combinations of terms of the form $ct^k e^{at}\cos(bt + d)$, where $a, b, c, d \in \mathbb{R}$ and $0 \leqslant k \leqslant n - 1$ is an integer.*

Proposition 2. *Given an n-dimensional PLDS $\mathcal{M} = (\mathcal{P}, F)$ and a partition $\mathcal{P}'$ of $\mathbb{R}^n$, $\mathcal{M}$ is well-behaved with respect to $\alpha_{\mathcal{P},\mathcal{P}'}$.*

Proof. Consider a trajectory $\tau : [0, T] \to \mathcal{P} \times \mathbb{R}^n$ in $\mathcal{M}$ such that for all t, $[\tau(t)]_D = P$, and $[\tau(t)]_C = e^{F(P)t}x_0$, where $P \in \mathcal{P}$ and x_0 is the initial continuous state of the trajectory. Define B to be the maximum of $bT + d$, where $\cos(bt + d)$ appears in the exponential matrix $e^{F(P)t}$ as given by Proposition 1. It is shown in [16], that the first order theory of reals with addition, multiplication, exponentiation and restricted cos and sin functions is o-minimal. This implies that the subset of reals defined by any formula with one free variable in the logic can be expressed as a finite union of intervals. Restricted cos and sin functions are those which are identical to cos and sin in a finite interval, and 0 everywhere else. Hence, we have that $\langle \mathbb{R}, \leqslant, +, \cdot, e, \sin|_{[0,B]}, \cos|_{[0,B]} \rangle$ is an o-minimal structure.

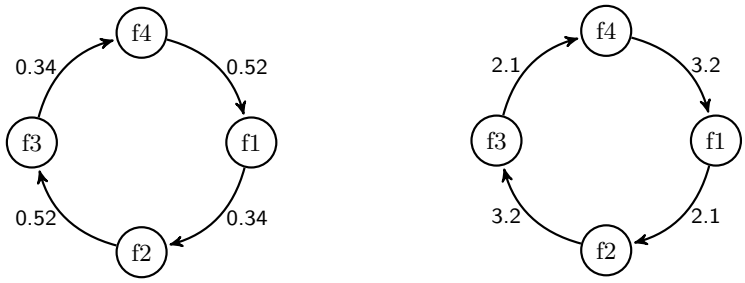
Next, we show that the trajectory enters and leaves a region of $\mathcal{P}'$ only finitely many times. Since, the number of regions is finite, this establishes that τ is finitely varying. Fix a polyhedron $P' \in \mathcal{P}'$. The following first-order logic formula $\varphi(t)$ over $\langle \mathbb{R}, \leqslant, +, \cdot, e, \sin|_{[0,B]}, \cos|_{[0,B]} \rangle$ defines the set of all times at which the trajectory τ is in P'.

$$\varphi(t) = \exists x (x \in P' \wedge 0 \leqslant t \leqslant T \wedge x = e^{F(P)t}x_0)$$

The last conjunct is expressible in the language due to Proposition 1. Futher, though the sin and cos are restricted, they only take arguments in the range $[0, B]$, due to t being restricted to the interval $0 \leq t \leq T$ and the way B is computed. Hence, due to o-minimality, the times at which τ is in P' is a finite union of intervals, and we obtain that τ exits P' only finitely many times in the interval $[0, T]$.

The hybrid system $\mathcal{M}$ is well-behaved with respect to $\alpha_{\mathcal{P},\mathcal{P}'}$, since, any finite restriction of a trajectory (with a possibly infinite domain) has finite number of discontinuities with respect to $\mathcal{P}'$. □

Remark 2. Note that the above proof extends to any partition which is definable in the theory with addition, multiplication and exponentiation.



(a) Weighted graph G_1 (b) Weighted graph G_2

Fig. 3. Quantitative Abstractions

Next, we illustrate the quantitative abstraction construction and analysis on the systems $\mathcal{M}_1$ and $\mathcal{M}_2$ in Example 1. The graphs G_1 and G_2 in Figure 3 are quantitative predicate abstractions of $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively. The state-space is partitioned by using the linear inequalities $\{x = 0, x > 0, x < 0, y = 0, y > 0, y < 0\}$. This partition generates 9 different regions, the four quadrants, the four positive and negative axes, which correspond to the boundary of the quadrants, and the origin. In this example, a simple construction of the weighted graph is presented, where the nodes corresponding to the quadrants and origin are eliminated because of redundancy. In practice, we may need to prune the graph to obtain useful results. The nodes $f1$ and $f3$ correspond to the positive and negative x axes, respectively, and the nodes $f2$ and $f4$ to the positive and negative y axes, respectively (in fact, the Figure 3 is a simplification of the weighted graphs shown in Figure 4 in the Appendix). There are several methods to compute weights based on reachable set computation for linear dynamics (see the discussion in Appendix).

Note that G_1 satisfies conditions G1 and G2 which implies Lyapunov stability of $\mathcal{M}_1$. On the other hand, G_2 does not satisfy conditions G2 or G3. Though we cannot conclude instability of $\mathcal{M}_2$, G_2 returns a counterexample, namely, the cycle $f1f2f3f4f1$ with weight > 1, explaining a potential reason for instability. The counterexample suggests that an infinite diverging execution is feasible by following the cycle infinitely many times. Such an execution exists in this case, for instance, repeating a scaled version of the execution shown in Figure 2(b) infinitely. However, in general, a diverging execution might not exist, as the counter-example could be due to the conservativeness of the abstraction.

6 Foundations of quantitative abstraction

In this section, we present the connection between quantitative abstractions and continuous simulations. First, we present an overview of continuous simulations between hybrid systems and show that they preserve stability. Next, we interpret

a quantitative abstraction as a one dimensional hybrid system, which continuously simulates the original one, and hence, preserves stability. We use these results to provide a proof of Theorem 1. The connection between quantitative abstraction and continuous simulations also enables us to define a partial ordering on the abstract weighted graphs, thus, formalizing the notion of refinement.

6.1 Continuous simulations and stability preservation

Pre-orders on systems which preserve properties of interest form the basis of any abstraction refinement framework. Simulations [17] are the classical pre-orders on systems which preserve various discrete-time properties including safety and the safe fragments of several branching time properties. For instance, if a system $\mathcal{H}_2$ simulates system $\mathcal{H}_1$ and $\mathcal{H}_2$ is safe, then $\mathcal{H}_1$ is safe as well.

Definition 7. *Given two hybrid systems $\mathcal{H}_1$ and $\mathcal{H}_2$, a binary relation $R \subseteq (Q_1 \times X_1) \times (Q_2 \times X_2)$ is said to be a* simulation *from $\mathcal{H}_1$ to $\mathcal{H}_2$, denoted $\mathcal{H}_1 \preceq^R \mathcal{H}_2$, if the following hold for every $(s_1, s_2) \in R$:*

- *for every transition $(s_1, s_1') \in \Sigma_1$, there exists a transition $(s_2, s_2') \in \Sigma_2$ such that $(s_1', s_2') \in R$; and*
- *for every trajectory $\tau_1 \in \Delta_1$ with $fstate(\tau_1) = s_1$, there exists a trajectory $\tau_2 \in \Delta_2$ with $fstate(\tau_2) = s_2$ such that $dom(\tau_1) = dom(\tau_2)$ and for all $t \in dom(\tau_1)$, $(\tau_1(t), \tau_2(t)) \in R$.*

However, it was observed in [21] that simulations do not suffice to preserve stability. Instead, a stronger notion which extends simulations with continuity constraints was proposed and shown to preserve stability. Below we present a simplified version of the definition of the relation and the stability preservation theorem in [21], as required for our setting.

Definition 8. *A binary relation R is a* continuous simulation *from $\mathcal{H}_1$ to $\mathcal{H}_2$, denoted $\mathcal{H}_1 \preceq_c^R \mathcal{H}_2$ if*

A1 R a simulation from $\mathcal{H}_1$ to $\mathcal{H}_2$;
A2 R and R^{-1} are continuous at $0_{\mathcal{H}_1}$ and $0_{\mathcal{H}_2}$, respectively;
A3 if $R((q_1, x_1), (q_2, x_2))$, then $x_1 = 0_{\mathcal{H}_1}$ if and only if $x_2 = 0_{\mathcal{H}_2}$; and
A4 $\exists \gamma > 0, \forall (q, x)$, $[(q, x)]_C \in B_\gamma(0_{\mathcal{H}_1}) \Rightarrow R(q, x) \neq \emptyset$.

Condition A3 states that the states corresponding to the origin in one system are mapped to the states corresponding to the origin in the other. Condition A4 states that the image of the relation R is not empty in a small neighborhood around the origin.

Theorem 2 ([21]). *Let R be a continuous simulation from $\mathcal{H}_1$ to $\mathcal{H}_2$. Then:*

- *$\mathcal{H}_2$ is Lyapunov stable implies $\mathcal{H}_1$ is Lyapunov stable.*
- *$\mathcal{H}_2$ is asymptotically stable implies $\mathcal{H}_1$ is asymptotically stable.*

This result shows that continuous simulations preserve both Lyapunov stability and asymptotic stability. The proof of the theorem is similar to that of the proof in [21] and can be found in the Appendix.

6.2 Quantitative predicate abstraction as a one-dimensional hybrid system

A predicate abstraction procedure constructs a simpler system which simulates the original system. Hence, it preserves all properties preserved by simulations. In order to deduce a similar result for stability analysis, we need to formally relate the hybrid system with a quantitative abstraction of the system. Hence, we interpret the weighted graph as representing a simple one dimensional hybrid system, and show that this one dimensional system continuously simulates the original system. First, we define the one-dimensional system from the graph and specify conditions which characterize their stability properties.

Given a weighted graph G, we construct a one-dimensional hybrid system $\mathcal{H}_G$. The discrete locations of $\mathcal{H}_G$ correspond to the nodes of G. Transitions correspond to pair of states such that scaling associated with continuous states is bounded by the weight of the edge corresponding to the discrete states. Similarly, a trajectory of $\mathcal{H}_G$ corresponds to following a finite or infinite path in G such that the scaling of any prefix of the trajectory corresponding to a prefix of the path is bounded by the weight associated with the prefix of the path. Furthermore, the scalings associated with any prefix of the trajectory is bounded by the maximum weight of an edge in the graph.

Definition 9. *Given an edge $e = (v_1, v, v_2)$ of a weighted graph $G = (V, V \cup \{\gamma\}, E, W)$, we define the set of trajectories corresponding to it, denoted Traj(e), as the set of all finite trajectories τ over $(V, \mathbb{R})$ satisfying:*

- $[\tau(0)]_D = v_1$;
- $\exists v, \forall 0 < t < last(dom(\tau)), [\tau(t)]_D = v$ *and* $\|\tau(t)\|/\|\tau(0)\| \leq MW(G)$; *and*
- $[\tau(ltime(\tau))]_D = v_2$ *and* $0 \leq \|\tau(ltime(\tau))\|/\|\tau(0)\| \leq W(e)$.

A weight on an edge in a quantitative abstraction is an upper bound on the scaling associated with the last time of an execution; however, the scalings associated with all the intermediate time points are bounded by the weight of the edge corresponding to the prefixes of the execution. Hence, in *Traj*(e) we only allow trajectories such that the scalings associated with the intermediate points is bounded by the maximum weight of an edge in the graph.

We will also define a set of infinite trajectories which is allowed by the graph.

$$InfTraj(G) = \{\tau \in Traj(V, \mathbb{R}) \mid ltime(\tau) = +\infty, \exists v \in V,$$

$$\forall t \in dom(\tau), [\tau(t)]_D = v, 0 \leq \|\tau(t)\|/\|\tau(0)\| \leq MW(G)\}$$

Definition 10. *Given a weighted graph $G = (V, V \cup \{\gamma\}, E, W)$, we define a hybrid system $\mathcal{H}_G = (V, \mathbb{R}_{\geq 0}, \Sigma, \Delta)$, where:*
$\Sigma = \{((v_1, r_1), (v_2, r_2)) \mid (v_1, \gamma, v_2) \in E, r_2/r_1 \leq W(v_1, \gamma, v_2)\}$.
$\Delta = \{\tau \mid \exists$ *finite or infinite splitting* $\tau = \tau_1 \circ \tau_2 \circ \ldots$, *such that* $\forall i$,
either $\exists e = (v_1, v, v_2) \in E, \tau_i \in Traj(e)$ *or* $\tau_i \in InfTraj(G)\}$.

The next theorem characterizes when $\mathcal{H}_G$ is Lyapunov stable.

Theorem 3. *Given a weighted graph G, $\mathcal{H}_G$ is Lyapunov stable if and only if*

C1 G does not contain any edges with infinite weights, that is, $W(e) < +\infty$ for every edge e of G.
C2 G does not contain simple cycles whose weight is strictly greater than 1, that is, $W(\pi) \leq 1$ for every simple cycle π of G.

Note that $\mathcal{H}_G$ constructed above is in general not asymptotically stable, since $\mathit{InfTraj}(G)$ consists of infinite trajectories which do not converge. Hence we interpret G as another one-dimensional hybrid system $\mathcal{H}_G^{Conv}$ which consists of infinite trajectories remaining within a single region and converging if there are no infinite weight edges.

$$\mathit{InfTrajConv}(G) = \{\tau \in \mathit{Traj}(V, \mathbb{R}) \mid \mathit{Conv}(\tau, 0)\} \cap \mathit{InfTraj}(G)$$

$\mathcal{H}_G^{Conv}$ is same as $\mathcal{H}_G$ except that $\mathit{InfTraj}(G)$ in the definition of Δ is replaced by $\mathit{InfTrajConv}(G)$ if G has no edges with infinite weight.

Theorem 4. *Given a weighted graph G, $\mathcal{H}_G^{Conv}$ is asymptotically stable if and only if Condition C1 holds and*

C3 G does not contain simple cycles whose weight is greater than or equal to 1, that is, $W(\pi) < 1$ for every simple cycle π of G.

6.3 Quantitative predicate abstraction as continuous simulation

Next, we show that there exists a simulation between $\mathcal{H}$ and the one-dimensional systems $\mathcal{H}_G$ and $\mathcal{H}_G^{Conv}$.

Theorem 5. *Let G be a quantitative abstraction of $\mathcal{H}$ with respect to α. Then $R = \{((q, x), (\alpha((q, x)), \|x\|)) \mid (q, x) \in \mathit{States}(\mathcal{H})\}$ is a continuous simulation from $\mathcal{H}$ to $\mathcal{H}_G$ and from $\mathcal{H}$ to $\mathcal{H}_G^{Conv}$.*

Now we are ready to provide a proof of Theorem 1.

Proof of theorem 1. Let us consider a hybrid system $\mathcal{H}$ and a quantitative abstraction G of $\mathcal{H}$ with respect to α. Suppose conditions G1 and G2 hold for G. We want to prove Lyapunov stability for $\mathcal{H}$. Due to conditions G1 and G2, Theorem 3 states that $\mathcal{H}_G$ is Lyapunov stable. By Theorem 5, we know there exists a continuous simulation R, defined as in the theorem, from $\mathcal{H}$ to $\mathcal{H}_G$. Then, by Theorem 2, we obtain $\mathcal{H}$ is Lyapunov stable.

Next, we prove the second part of Theorem 1. Suppose conditions G1 and G3 hold for G. We want to show asymptotic stability for the hybrid system $\mathcal{H}$. Since, Conditions G1 and G3 hold, we obtain from Theorem 4 that $\mathcal{H}_G^{Conv}$ is asymptotically stable. Then, from Theorem 5, we know that there is a continuous simulation R from $\mathcal{H}$ to $\mathcal{H}_G^{Conv}$. Finally, from Theorem 2, we obtain $\mathcal{H}$ is asymptotically stable. □

6.4 Refinements

The interpretation of the weighted graph as a one-dimensional system also provides a natural notion of refinements on the graphs.

Definition 11. *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be hybrid systems and R a binary relation such that $\mathcal{H}_1 \preceq_c^R \mathcal{H}_2$. A hybrid system $\mathcal{H}_3$ is a* refinement *of $\mathcal{H}_2$ with respect to $\mathcal{H}_1$, if there exist binary relations R_1 and R_2 such that $\mathcal{H}_1 \preceq_c^{R_1} \mathcal{H}_3 \preceq_c^{R_2} \mathcal{H}_2$.*

Theorem 6. *Let $\mathcal{H}$ be a hybrid system, and $\alpha_1 : States(\mathcal{H}) \to V_1$ and $\alpha_2 : States(\mathcal{H}) \to V_2$ be two abstraction functions such that for every $v_2 \in V_2$ there exists $v_1 \in V_1$ with $\alpha_2^{-1}(v_2) \subseteq \alpha_1^{-1}(v_1)$. Let G_1 and G_2 be the minimal quantitative abstractions of $\mathcal{H}$ with respect to α_1 and α_2, respectively. Then:*

1. *$\mathcal{H}_{G_1}$ simulates $\mathcal{H}_{G_2}$.*
2. *$\mathcal{H}_{G_1}^{Conv}$ simulates $\mathcal{H}_{G_2}^{Conv}$.*

In particular, $\mathcal{H}_{G_2}$ is a refinement of $\mathcal{H}_{G_1}$ with respect to $\mathcal{H}$ and $\mathcal{H}_{G_2}^{Conv}$ is a refinement of $\mathcal{H}_{G_1}^{Conv}$ with respect to $\mathcal{H}$.

Remark 3. Note that Theorem 6 will not be true for arbitrary abstractions $\mathcal{H}_{G_1}$ and $\mathcal{H}_{G_2}$. Hence, we enforce minimality, however, this can be relaxed to any abstraction construction procedure which is monotonic with respect to the abstraction functions. A consequence of the theorem is that, it establishes a partial ordering on the abstract graphs based on a partial ordering on the abstraction functions. Hence, the ordering on the abstraction functions can be used to obtain refinements of the graphs. For instance, adding more predicates yields a refinement.

7 Conclusion

In this paper, we presented the formal foundations for the quantitative predicate abstraction based stability analysis by establishing connections with continuous simulation relations. Here, we have ignored the computational issues related to the computation of abstractions and refinements. These have been explored to some extent in [23, 24] for the class of piecewise constant derivative systems and polyhedral switched systems. Future work will focus on extending this approach to hybrid systems with richer dynamics. Further, since, a failure to prove stability returns a potential counter-example, one can build a framework of counter-example guided abstraction refinement for stability analysis, which will be explored in the future.

References

1. R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, pages 209–229, London, UK, UK, 1993. Springer-Verlag.

2. R. Alur, T. Dang, and F. Ivancic. Counter-example guided predicate abstraction of hybrid systems. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 208–223, 2003.
3. R. Alur, T. Dang, and F. Ivancic. Predicate abstraction for reachability analysis of hybrid systems. *ACM Transactions on Embedded Computing Systems*, 5(1):152–199, 2006.
4. A. Arapostathis and M. E. Broucke. Stability and controllability of planar, conewise linear systems. *Systems & Control Letters*, 56(2):150–158, 2007.
5. T. Ball, R. Majumdar, T. Millstein, and S. K. Rajamani. Automatic predicate abstraction of c programs. In *Proceedings of the ACM SIGPLAN 2001 Conference on Programming Language Design and Implementation*, PLDI '01, pages 203–213, New York, NY, USA, 2001. ACM.
6. B. Cook, A. Podelski, and A. Rybalchenko. Proving program termination. *Commun. ACM*, 54(5):88–98, 2011.
7. A. Girard. Reachability of uncertain linear systems using zonotopes. In *Proceedings of the International Conference on Hybrid Systems: Computation and Control*, pages 291–305, 2005.
8. A. Girard and C. Guernic. Zonotope/Hyperplane intersecion for hybrid systems reachability analysis. In *Proceedings of the International Conference on Hybrid Systems: Computation and Control*, pages 215–228, 2008.
9. R. Goebel, R. Sanfelice, and A. Teel. Hybrid dynamical systems. *IEEE Control Systems, Control Systems Magazine*, 29:28–93, 2009.
10. S. Graf and H. Saidi. Construction of abstact state graphs with PVS. In *Proceedings of the International Conference on Computer Aided Verification*, pages 72–83, 1997.
11. T. A. Henzinger. The Theory of Hybrid Automata. In *Proceedings of the IEEE Symposium on Logic in Computer Science*, pages 278–292, 1996.
12. M. W. Hirsch, S. Smale, and R. L. Devaney. *Differential equations, dynamical systems, and an introduction to chaos*. Academic Press, Waltham (Mass.), 2013.
13. J. Kapinski, J. V. Deshmukh, S. Sankaranarayanan, and N. Arechiga. Simulation-guided lyapunov analysis for hybrid dynamical systems. In *17th International Conference on Hybrid Systems: Computation and Control (part of CPS Week), HSCC'14, Berlin, Germany, April 15-17, 2014*, pages 133–142, 2014.
14. H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, Upper Saddle River, NJ, 1996.
15. H. Lin and P. J. Antsaklis. Stability and stabilizability of switched linear systems: A survey of recent results. *IEEE Transactions on Automatic Control*, 54(2):308–322, 2009.
16. A. M. Lou van den Dries and D. Marker. The elementary theory of restricted analytic fields with exponentiation. *Annals of Mathematics, Second Series*, 140(1):183–205, 1994.
17. R. Milner. *Communication and Concurrency*. Prentice-Hall, Inc, 1989.
18. E. Möhlmann and O. Theel. Stabhyli: a tool for automatic stability verification of non-linear hybrid systems. In *Proceedings of the International Conference on Hybrid Systems: Computation and Control*, pages 107–112, New York, NY, USA, 2013. ACM.
19. A. Papachristodoulou and S. Prajna. On the construction of Lyapunov functions using the sum of squares decomposition. In *Conference on Decision and Control*, 2002.
20. 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.

21. P. Prabhakar, G. E. Dullerud, and M. Viswanathan. Pre-orders for reasoning about stability. In *Proceedings of the International Conference on Hybrid Systems: Computation and Control*, pages 197–206, 2012.
22. P. Prabhakar, J. Liu, and R. M. Murray. Pre-orders for reasoning about stability properties with respect to input of hybrid systems. In *Proceedings of the International Conference on Embedded Software, EMSOFT 2013, Montreal, QC, Canada, September 29 - Oct. 4, 2013*, pages 1–10, 2013.
23. P. Prabhakar and M. G. Soto. Abstraction based model-checking of stability of hybrid systems. In *Proceedings of the International Conference on Computer Aided Verification*, 2013.
24. P. Prabhakar and M. G. Soto. An algorithmic approach to stability verification of polyhedral switched systems. In *American Control Conference*, 2014.
25. P. Prabhakar and M. Viswanathan. A dynamic algorithm for approximate flow computations. In *Proceedings of the International Conference on Hybrid Systems: Computation and Control*, pages 133–143, 2010.
26. S. Sankaranarayanan and A. Tiwari. Relational abstractions for continuous and hybrid systems. In *Proceedings of the International Conference on Computer Aided Verification*, pages 686–702, 2011.
27. E. D. Sontag. Input to state stability: Basic concepts and results. In *Nonlinear and Optimal Control Theory*, pages 163–220. Springer, 2006.
28. A. Tiwari. Abstractions for hybrid systems. *Formal Methods in System Design*, 32(1):57–83, 2008.

Appendix

A Proof of Theorem 2

Notation We use $Lyap(\mathcal{H}, \epsilon, \delta)$ to denote the fact that for all $\sigma \in Exec(\mathcal{H})$ with $[fstate(\sigma)]_C \in B_\delta(0_\mathcal{H})$, $[States(\sigma)]_C \subseteq B_\epsilon(0_\mathcal{H})$.

Notation We use $Cont(R, \epsilon, \delta, a)$ to represent the fact that $R(B_\delta(a)) \subseteq B_\epsilon(R(a))$.

Suppose $\mathcal{H}_2$ is Lyapunov stable. We will show $\mathcal{H}_1$ is Lyapunov stable. We need to check that $\forall \epsilon > 0$ there exists $\delta > 0$ such that $Lyap(\mathcal{H}_1, \epsilon, \delta)$ holds. Fix $\epsilon > 0$. We have by continuity of R^{-1} that there exists $\epsilon' > 0$ such that $Cont(R^{-1}, \epsilon, \epsilon', 0_{\mathcal{H}_2})$ holds. Then by Lyapunov stability of $\mathcal{H}_2$ there exists $\delta' > 0$ such that $Lyap(\mathcal{H}_2, \epsilon', \delta')$ holds. Due to continuity of R there exists $\delta > 0$ such that $Cont(R, \delta', \delta, 0_{\mathcal{H}_1})$.

Given an execution $\sigma \in Exec(\mathcal{H}_1)$ such that $fstate(\sigma) \in B_\delta(0_{\mathcal{H}_1})$, we have $R(fstate(\sigma)) \in B_{\delta'}(0_{\mathcal{H}_2})$ because $Cont(R, \delta', \delta, 0_{\mathcal{H}_1})$ holds. We know, by condition $A4$, $R(fstate(\sigma))$ is not empty, so pick up a state $s_2 \in R(fstate(\sigma))$. There exists an execution σ' from s_2, by definition of simulation, such that $\sigma' \in R(\sigma)$; and $\sigma'(i)(t) \in B_{\epsilon'}(0_{\mathcal{H}_2})$ for every $(i,t) \in htd(\sigma')$. We will check that every $\sigma(i)(t) \in B_\epsilon(0_{\mathcal{H}_1})$. Fix $(i,t) \in htd(\sigma)$. We know $\sigma'(i)(t) \in R(\sigma(i)(t)) \cap B_{\epsilon'}(0_{\mathcal{H}_2})$, therefore, $\sigma(i)(t) \in B_\epsilon(0_{\mathcal{H}_1})$ since $Cont(R^{-1}, \epsilon, \epsilon', 0_{\mathcal{H}_2})$ holds. Hence, we can claim $Lyap(\mathcal{H}_1, \epsilon, \delta)$.

Now, suppose $\mathcal{H}_2$ is asymptotic stable. We will prove that $\mathcal{H}_1$ is asymptotic stable. We need to add to Lyapunov stability that $\forall \sigma \in Exec(\mathcal{H}_1)$ and $\forall \eta > 0$, there exists $(i,t) \in htd(\sigma)$ such that $[\sigma_{(i,t)}]_C \subseteq B_\eta(0_{\mathcal{H}_1})$. Fix $\eta > 0$. Consider σ and σ' as defined above. By using an analogous reasoning, there is a $\eta' > 0$ such that $Cont(R^{-1}, \eta, \eta', 0_{\mathcal{H}_2})$ holds, and due to asymptotic stability of $\mathcal{H}_2$, there exists a pair $(i,t) \in htd(\sigma')$ with $\sigma'_{(i,t)} \in B_{\eta'}(0_{\mathcal{H}_2})$. Now, fix $(i^*, t^*) \in htd(\sigma_{(i,t)})$. We know $\sigma'(i^*)(t^*) \in R(\sigma(i^*)(t^*)) \cap B_{\eta'}(0_{\mathcal{H}_2})$, hence $\sigma(i^*)(t^*) \in B_\eta(0_{\mathcal{H}_1})$. And this proves that $\sigma_{(i,t)} \subseteq B_\eta(0_{\mathcal{H}_1})$, so we deduce asymptotic stability on $\mathcal{H}_1$. □

B Proof of Theorem 5

We will prove that R is a continuous simulation. Fix $((q,x), (\alpha((q,x)), \|x\|)) \in R$ and a transition $((q,x),(q',x')) \in \Sigma$. We choose the pair $((\alpha((q,x)), \|x\|), (\alpha((q',x')), \|x'\|))$. We will prove that $((\alpha((q,x)), \|x\|), (\alpha((q',x')), \|x'\|)) \in \Sigma_G$ and $((q',x'), (\alpha((q',x')), \|x'\|)) \in R$. It is clear that $\alpha((q,x)) = v$ and $\alpha((q',x')) = v'$ are nodes in the graph $G(\mathcal{H})$, and $Z^D(v, \epsilon, v') \neq \emptyset$ because $((q,x),(q',x')) \in \Sigma$. Hence there exists an edge $e = (v, \epsilon, v') \in E$. By the weight condition we have $\|[(q',x')]_C\| / \|[q,x]_C\| = \|x'\| / \|x\| \leqslant W(e)$. This concludes that $((\alpha((q,x)), \|x\|), (\alpha((q',x')), \|x'\|)) \in \Sigma_G$. We have $((q',x'), (\alpha((q',x')), \|x'\|)) \in R$ just by definition.

Now, fix $((q,x),(\alpha((q,x)),\|x\|)) \in R$ and a trajectory $\tau \in \Delta$ with $fstate(\tau) = (q,x)$. We want to check that there exists a trajectory $\tau_G \in \Delta_G$ such that $fstate(\tau_G) = R((q,x))$ and $\tau_G \in R(\tau)$. Consider $\tau_G = R(\tau)$, for which trivially holds $\tau_G \in R(\tau)$. We have the following equalities, $fstate(\tau_G) = (\alpha(fstate(\tau))$, $\|[fstate(\tau)]_C\|) = ((\alpha((q,x)),\|x\|))$. It remains to prove that $\tau_G \in \Delta_G$. By the Assumption 1, it says $\alpha \circ \tau$ finitely varying on time, there exist a time sequence $S = \{t_0, t_1, \cdots\}$, finite or infinite, and a set $\{v_0, v_0', v_1, v_1', \cdots\}$ such that $\alpha(\tau(t_i)) = v_i$ and $\alpha(\tau(t)) = v_i'$ for $t \in (t_i, t_{i+1})$. Note that v_i are nodes in the graph $G(\mathcal{H})$. Since τ is evolving from v_i to v_{i+1} through v_i', we conclude $Z^C(v_i, v_i', v_{i+i}) \neq \emptyset$, which implies the existence of $e_i = (v_i, v_i', v_{i+1}) \in E$. Consider the edge $e_i = (v_i, v_i', v_{i+1})$. We have $[\tau_G(t_i)]_D = \alpha(\tau(t_i)) = v_i$. By the weight condition and the τ_G definition we get $[\tau_G(t)]_D = \alpha(\tau(t)) = v_i'$ and $\|\tau_G(t)\|/\|\tau_G(t_i)\| = \|[\tau(t)]_C\|/\|[\tau(t_i)]_C\| \leqslant MW(G)$ for $t \in (t_i, t_{i+1})$. Now, we distingish two cases, the one where the trajectory τ is infinite and S is finite, and the rest. In the first case we have $ltime(\tau) = \infty$, and $S = \{t_0, t_1, \cdots, t_k\}$. Because there is no zeno behaviour at the hybrid system $\mathcal{H}$, we will have for this last t_k that $[\tau_G(t_k)]_D = \alpha(\tau(t_k)) = v_k$, $[\tau_G(t)]_D = \alpha(\tau(t)) = v_k'$ and $\|\tau_G(t)\|/\|\tau_G(t_k)\| = \|[\tau(t)]_C\|/\|[\tau(t_k)]_C\| \leqslant MW(G)$ for $t \in (t_k, \infty)$. For the case τ is infinite with S infinite or τ finite, by the weight condition and the τ_G definition, we get $[\tau_G(t_{i+1})]_D = v_{i+1}$, $\|\tau_G(t_{i+1})\|/\|\tau_G(t_i)\| = \|[\tau(t_{i+1})]_C\|/\|[\tau(t_i)]_C\| \leqslant W(e)$ and, $[\tau_G(t)]_D = v_i'$ and $\|\tau_G(t)\|/\|\tau_G(t_i)\| = \|[\tau(t)]_C\|/\|[\tau(t_i)]_C\| \leqslant MW(G)$ for $t \in (t_i, t_{i+1})$. Thanks to all these conditions we know $\tau_G = \tau_{G_1} \circ \tau_{G_k} \circ \cdots$ where τ_{G_i} is contained in $Traj(e)$ or $InfTraj(G)$ for every $i \in dom(S)$. Hence we conclude $\tau_G \in \Delta_G$. Therefore, it is proven that R is a simulation.

Next, we check the continuity of R at $0_{\mathcal{H}}$. Fix $\epsilon > 0$. We choose $\delta = \epsilon$. We have for every state $s = (q,x) \in B_\epsilon(0_{\mathcal{H}})$ that $s' = (\alpha((q,x)),\|x\|) \in B_\epsilon(0_{\mathcal{H}_G})$, because $|\|[s']_C\| - \|[0_{\mathcal{H}_G}]_C\|| = |\|\|x\|\| - \|0\|| = \|x - 0\| = \|[s]_C - [0_{\mathcal{H}}]_C\| < \epsilon = \delta$.

Now, we prove continuity of R^{-1} at $0_{\mathcal{H}_G}$. Fix $\epsilon > 0$. We choose $\delta = \epsilon$. Fix $s = (v,r) \in B_\epsilon(0_{\mathcal{H}_G})$. We have $R^{-1}(s) = \{(q,x) \in Q \times X : \alpha^{-1}(v) = q$ and $\|x\| = r\}$. Pick a state $s' = (q', x') \in R^{-1}(s)$. It is clear that $|\|[s']_C\| - \|[0_{\mathcal{H}_G}]_C\|| = |\|x'\| - \|0\|| = \|x' - 0\| = r = \|[s]_C - [0_{\mathcal{H}}]_C\| < \epsilon = \delta$. This implies $s' \in B_\epsilon(0_{\mathcal{H}})$.

$A3$ holds because $\|x\| = 0$ iff $x = 0$.

$A4$ is true because α is defined for every $(q,x) \in Q \times X$ and $\|x\|$ is defined for every $x \in \mathbb{R}^n$. So $\forall (q,x) \in Q \times X$ we have $(\alpha((q,x)),\|x\|) \neq \emptyset$. □

C Proof of Theorem 6

We will prove that $\mathcal{H}_{G_1}$ continuous simulate $\mathcal{H}_{G_2}$. We have $\forall v_2 \in V_2\ \exists v_1 \in V_1$ with $\alpha_2^{-1}(v_2) \subseteq \alpha_1^{-1}(v_1)$. We define the following relation, $R : States(\mathcal{H}_{G_2}) \to States(\mathcal{H}_{G_1})$, where $R(v_2, r) = (v_1, r)$ for a state v_1 such that $\alpha_2^{-1}(v_2) \subseteq \alpha_1^{-1}(v_1)$ is a n to 1 mapping.

We will check that R satisfies simulation properties.

Fix $\iota_2 \in \Sigma_{G_2}$ and $(fstate(\iota_2),) \in R$. We need to find a transition ι_1 in $\mathcal{H}_{G_1}$ such that $lstate(\iota_1) = R(lstate(\iota_2))$. We fix $fstate(\iota_1) = R(fstate(\iota_2))$. We know there exist v_2, v_2' nodes in G_2 such that $Z^D(v_2, v_2')$ not empty. Due to the

fact ${ReachRel_{\mathcal{H}}}^{D}(\alpha_2^{-1}(v_2), \alpha_2^{-1}(v_2')) \subseteq {ReachRel_{\mathcal{H}}}^{D}(\alpha_1^{-1}(R(v_2)), \alpha_1^{-1}(R(v_2'))$, we have $Z^D(v_2, v_2') \subseteq Z^D(R(v_2), R(v_2'))$. Hence, there exists and edge $(R(v_2), \gamma, R(v_2'))$ in G_1 such that ${ReachRel_{\mathcal{H}_{G_1}}}^{D}(\alpha_1^{-1}(R(v_2)), \alpha_1^{-1}(R(v_2'))))$ is not empty. Then we have $lstate(\iota_1) \in R(v_2)$, so we choose $lstate(\iota_1) = R(lstate(\iota_2))$ and we also get $0 \leqslant \|lstate(\iota_1)\|/\|fstate(\iota_1)\| \leqslant W(R(v_2), R(v_2'))$.

We prove, for simplicity, for a finite execution. In other cases there will be necessary a small extension. Fix a trajectory $\tau_2 \in \mathcal{H}_{G_2}$ with $fstate(\tau_2) = s_1$ and $(s_1, s_1') \in R$. We need to find a trajectory $\tau_1 \in \mathcal{H}_{G_1}$ such that $fstate(\tau_1) = s_1'$ and $\tau_1 \in R(\tau_2)$. Define the trajectory $\tau_1 = R(\tau_2)$. We have trivially $fstate(\tau_1) = s_1' = R(s_1) = fstate(\tau_2)$. Let check now that τ_1 is a trajectory in $\mathcal{H}_{G_1}$. We know, by $\mathcal{H}_{G_2}$ construction, that there exists a splitting of trajectory τ_2. The τ_2 splitting is determined by a sequence of times $S = \{t_1, t_2, \cdots, t_m\}$, and a sequence of nodes in G_2 $\{v_1, v_1', \cdots, v_{m-1}', v_m\}$ with $\alpha_2(\tau_2(t_i)) = v_i \ \forall 1 \leqslant i \leqslant m$ and ${ReachRel_{\mathcal{H}_{G_2}}}^{C}(\alpha_2^{-1}(v_i), \alpha_2^{-1}(v_i'), \alpha_2^{-1}(v_{i+1}))$ not empty $\forall i \in \{1, \cdots, m-1\}$. Then, let say $Z^C(v_i, v_i', v_{i+1}) \neq \emptyset \ \forall i \in \{1, \cdots, m-1\}$. Define the set $\{R(v_1), R(v_1'), \cdots, R(v_{m-1}'), R(v_m)\}$ of nodes in G_1. The fact ${ReachRel_{\mathcal{H}_{G_2}}}^{C}(\alpha_2^{-1}(v_i), \alpha_2^{-1}(v_i'), \alpha_2^{-1}(v_{i+1})) \subseteq {ReachRel_{\mathcal{H}_{G_1}}}^{C}(\alpha_1^{-1}(R(v_i)), \alpha_1^{-1}(R(v_i')), \alpha_1^{-1}(R(v_{i+1})))$ implies $Z^C(v_i, v_i', v_{i+1}) \subseteq Z^C(R(v_i),\ R(v_i'), R(v_{i+1})) \ \forall i \in \{1, \cdots, m-1\}$. Since $Z^C(v_i, v_i', v_{i+1}) \neq \emptyset$, we have also that $Z^C(R(v_i), R(v_i'),\ R(v_{i+1})) \neq \emptyset \ \forall i \in \{1, \cdots, m-1\}$. Therefore there exists a path $\pi = R(v_1)R(v_1') \cdots\ R(v_m)$ in G_1. Then we have $fstate(\tau_1) = R(v_1)$, $\|\tau_1(t)\|/\|\tau_1(0)\| \leqslant MW(G_1)$ and $0 \leqslant \|\tau_1(t_m)\|/\|\tau_1(t_i)\| \leqslant W(R(v_i), R(v_i'), R(v_{i+1})) \ \forall i \in \{1, \cdots, m-1\}$.

Continuity for R and R^{-1} is obvious, since the continuous part of the relation R is just the identity. We know identity is a continuous function, and its inverse is also identity.

$A3$ holds because $\|r\| = 0$ iff $r = 0$.

$A4$ is true because R is defined for every state in $\mathcal{H}_{G_2}$. □

D Proof of Proposition 1

The solution of a linear dynamical system $\dot{x}(t) = Ax(t)$ corresponds to $x(t) = e^{At} \cdot x_0$ where x_0 refers to the initial point.

Theorem 7 ([12]). *Let A be an n-dimensional real matrix. Then, there exists a real invertible matrix P and a real matrix J (so-called real Jordan matrix) of the form*

$$\begin{pmatrix} J_1 & 0 & \cdots & 0 & 0 \\ 0 & J_2 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & J_{k-1} & 0 \\ 0 & 0 & \cdots & 0 & J_k \end{pmatrix},$$

where the blocks J_i are either of the form

$$\begin{pmatrix} \lambda_i & 1 & \cdots & 0 & 0 \\ 0 & \lambda_i & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_i & 1 \\ 0 & 0 & \cdots & 0 & \lambda_i \end{pmatrix}$$

or of the form

$$\begin{pmatrix} a_i & -b_i & 1 & 0 & 0 \cdots \cdots & 0 \\ b_i & a_i & 0 & 1 & 0 \cdots \cdots & 0 \\ 0 & 0 & a_i & -b_i & 1 \; 0 \; \cdots & 0 \\ 0 & 0 & b_i & a_i & 0 \; 1 \; \cdots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & & \cdots & \cdots & 0 \; a_i & -b_i \\ 0 & & \cdots & \cdots & 0 \; b_i & a_i \end{pmatrix}$$

with $\lambda_i, a_i, b_i \in \mathbb{R}$ for every i, such that $A = PJP^{-1}$.

It can be easily verified that $e^{At} = Pe^{Jt}P^{-1}$ from definition of matrix exponential.

Exponential of a block diagonal matrix is equal to the matrix where the blocks are replaced by their exponentials. Note that the real Jordan matrix is block diagonal, hence

$$e^J = \begin{pmatrix} e^{J_1} & 0 & \cdots & 0 \\ 0 & e^{J_2} & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & e^{J_k} \end{pmatrix}$$

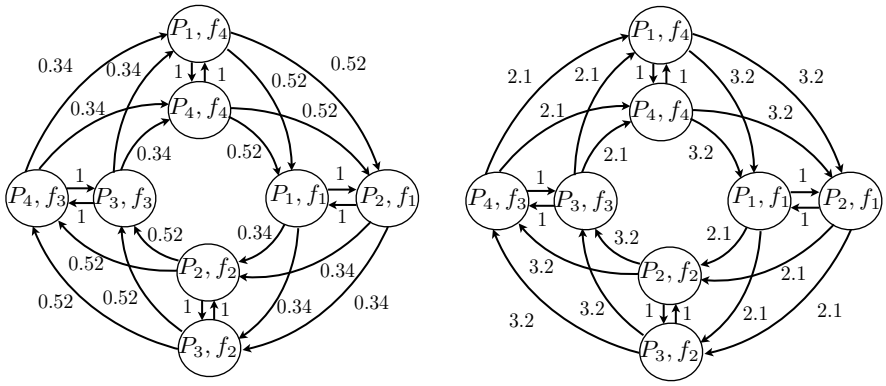
When J_i is of the first form, then $e^{J_i t}$ is given by

$$e^{\lambda_i t} \begin{pmatrix} 1 & t & \frac{t^2}{2!} & \cdots & \frac{t^{m-1}}{(m-1)!} \\ 0 & 1 & t & \cdots & \frac{t^{m-2}}{(m-2)!} \\ \vdots & & 1 & \ddots & \vdots \\ 0 & & & \ddots & t \\ 0 & \cdots & & 0 & 1 \end{pmatrix},$$

while in the case of J_i of the second form, $e^{J_i t}$ is equal to

$$e^{at} \begin{pmatrix} R & Rt & R\frac{t^2}{2!} & \cdots & R\frac{t^{m-1}}{(m-1)!} \\ 0 & R & Rt & \cdots & R\frac{t^{m-2}}{(m-2)!} \\ & & \ddots & \ddots & \\ & & & \ddots & 0 \\ 0 & & \cdots & 0 & R \end{pmatrix}, \text{ where } R = \begin{pmatrix} cos(b_i t) & sin(b_i t) \\ sin(b_i t) & cos(b_i t) \end{pmatrix}$$

Note that every element of e^{Jt} is of the form $ct^k e^{at} cos(bt+d)$ where $a, b, c, d \in \mathbb{R}$ (note $sin(x) = cos(\frac{\pi}{2} - x)$). Therefore, the product of the matrices $Pe^{Jt}P^{-1}$



(a) Weighted graph G_1 (b) Weighted graph G_2

Fig. 4. Exact Quantitative Abstractions

results in a matrix where every element is a linear combination of terms of the form $ct^k e^{at} cos(bt + d)$ with $a, b, d \in \mathbb{R}$. □

E Discussion on reach relations

As we already said in the work, the crux of constructing the weighted graph is in computing the reachability relations to determine the weights. There are some recent results for abstracting the reachability relation [26]. However, there is extensive research on the computation of over-approximation of reachable sets of linear dynamical systems including those with bounded error approximation [25, 7, 8]. To exploit these results and accompanying tools, we reduce the computation of the reachability relation to the computation of a reachable set starting from a compact convex polyhedral set.

Based on the two defined reachability relations, reachability sets are defined to be the states reached in P_2 from P_1, considering in one case trajectories evolving through the same set P and in the other just single transitions. Let us define the reachability sets,

$$Reach^C(P_1, P, P_2) := \{s_2 \mid \exists s_1 : (s_1, s_2) \in ReachRel^C(P_1, P, P_2)\},$$

$$Reach^D(P_1, P_2) := \{s_2 \mid \exists s_1 : (s_1, s_2) \in ReachRel^D(P_1, P_2)\}.$$

Our main observations are the following:

Lemma 1. *Given $P_1, P, P_2 \in \mathcal{P}$ generated by homogeneous linear expressions, $(s_1, s_2) \in ReachRel^C(P_1, P, P_2)$, define $P_1' = \{s \in P_1 \mid \|s\| = 1\}$. Then:*

$$Reach^C(P_1', P, P_2) = \emptyset \Leftrightarrow ReachRel^C(P_1, P, P_2) = \emptyset$$

$$\sup_{s \in Reach^C(P_1', P, P_2)} \|s\| = \sup_{(s_1, s_2) \in ReachRel^C(P_1, P, P_2)} \|s_2\| / \|s_1\|.$$

Proof. The elements of the partition satisfy the property that they are invariant with respect to positive scaling, that is, if $x \in P$, then $kx \in P$ for $k \geq 0$. Next, we observe that if $(s_1, s_2) \in ReachRel^C(P_1, P, P_2)$, then $s_2/\|s_1\| \in Reach(P_1', P, P_2)$. If σ is an execution from s_1 in P_1 to s_2 in P_2 while remaining within P, then $\sigma/\|fstate(\sigma)\|$ is an execution from P_1' to P_2 while remaining within P. Therefore, the two conditions hold. □

Lemma 2. *Given $P_1, P, P_2 \in \mathcal{P}$ generated by homogeneous linear expressions, $(s_1, s_2) \in ReachRel^D(P_1, P_2)$, define $P_1' = \{s \in P_1 \mid \|s\| = 1\}$. Then:*

$$Reach^D(P_1', P_2) = \emptyset \Leftrightarrow ReachRel^D(P_1, P_2) = \emptyset$$

$$\sup_{s \in Reach^D(P_1', P_2)} \|s\| = \sup_{(s_1, s_2) \in ReachRel^D(P_1, P_2)} \|s_2\| / \|s_1\|.$$

The proof is analogous to the one of the previous lemma.

Choosing the norm to be the infinity norm, the closure of the set P_1' is a finite union of compact convex polyhedral sets. Further, computing a polyhedral over-approximation of the set $Reach^C(P_1', P, P_2)$ or $Reach^D(P_1', P_2)$ reduces the weight computation problem to a finite set of linear optimization problems as optimizing $\|s\|$ corresponds to optimizing each component of the continuous state, whereas optimizing over $\|s_2\|/\|s_1\|$ is not a linear objective. Hence, a quantitative abstraction of $\mathcal{M}$ can be computed efficiently, and Conditions G1 and G2 can be verified in time polynomial in the size of the graph. A sufficient condition for ensuring condition C4 is that the matrices representing the linear dynamics are Hurwitz. However, it is not necessary. A necessary condition is that for every $P \in \mathcal{P}$, the eigenvalues associated with the eigenvectors of $F(P)$ which lie within P are negative (see [4] for more details).