On Timed Simulation Relations for Hybrid Systems and Compositionality

Goran Frehse

VERIMAG, goran.frehse@verimag.fr,
WWW home page: http://www-verimag.imag.fr/~frehse

Abstract. Timed and weak timed simulation relations are often used to show that operations on hybrid systems result in equivalent behavior or in conservative overapproximations. Given that systems are frequently designed and verified in a modular approach, it is desirable that this relationship is compositional, which is not the case for hybrid systems in general. We identify subclasses of linear hybrid automata that are compositional with respect to timed, respectively weak timed simulation.

1 Introduction

Hybrid automata are notoriously hard to analyze, so they are often overapproximated with hybrid automata of simpler dynamics, see [1,2,3] and references therein. The proofs used to show that the constructed automata are indeed conservative frequently involve timed simulation, or a weak variant that allows unobservable transitions as long as they don't change the variables. The analysis is usually challenging even for the abstracted system, and increases exponentially with the number of components and variables. Compositional reasoning is known as a valuable tool to counter this problem. However, timed and weak timed simulation are not compositional for hybrid automata with arbitrary dynamics. Consequently, a successful compositional analysis of the abstracted system does not imply safety of the original system when timed simulation was used in proving conservativeness.

In this paper we identify classes of hybrid automata that are compositional with respect to timed, respectively weak timed simulation. If such a class is used to overapproximate a system, conservativeness is consequently guaranteed and compositional reasoning valid. These results are directly applicable to strengthen the overapproximation operators in [1,2,3] with respect to compositionality.

Related Work We use the hybrid automata in [4] with minor modifications. We define a subset of the controlled variables as output variables, specify the activities via their derivatives, include a set of initial states, and consider the same controlled variables in all locations. The hybrid automata in [4] are known to be compositional for trace inclusion [4], see [5] for applications. The controlled variables are needed to prove compositionality. We add output variables to hide internal (non-output) behavior, i.e., so we can compare automata whose output variables behave identically while the internal workings may be different. More sophisticated hybrid input/output-automata (HIOA) are proposed and studied in detail in [6]. HIOA impose input-enabledness that we do not require, so the hybrid automata in this paper are equivalent to the pre-HIOA of [6]. The stricter I/O-distinction in [6] may be used to ensure some liveness properties; we only consider safety. We use a compositional type of simulation from [6], which we call *trace simulation* to set it apart from timed simulation.

Timed simulation is usually defined using labeled transition system (LTS) semantics [7]. Our definition is directly based on runs of hybrid automata, but is otherwise equivalent. In earlier work, we proposed semantic criteria for compositionality of timed simulation without giving an interpretation on the hybrid automaton level, and not for weak timed simulation [8]. The framework used in this paper presents a substantial improvement and simplification, and our previous results on compositionality and assume/guarantee-reasoning from [9,10] can be transferred to it. For the sake of brevity, we provide mostly proof sketches. Detailed proofs for most of the results (except those involving overlap-closure) can be found in [10].

In the following section, we present our hybrid automata and their semantics. In Sect. 3 we define trace and timed simulation, as well as their weak counterparts. In Sect. 4 we identify compositional subclasses for these types of simulation. Finally, we draw some conclusions in Sect. 5.

2 Hybrid Automata

We use a standard hybrid automaton model and parallel composition operator from [4], to which we add a subset of output variables. A variable is either an *uncontrolled variable* (also called input), and can therefore change arbitrarily at any time, or *controlled.* In parallel composition, controlled variables can not be changed independently by other automata in the composition. These elements are essential to compositionality [11]. A subset of the controlled variables are *output* variables, which, together with the uncontrolled variables, define the externally visible behavior of the automaton. Note that the uncontrolled variables may be restricted in their derivatives, and can only change arbitrarily inside the invariant. This allows us to model causal and noncausal coupling between variables, which is useful, e.g., to model conservation laws.

Preliminaries Given a set $X = \{x_1, \dots, x_n\}$ of variables, a *valuation* is a function $v : X \to \mathbb{R}$. We use $\dot{X}$ to denote the set $\{\dot{x}_1, \dots, \dot{x}_n\}$ of dotted variables, and X' to denote the set $\{x'_1, \dots, x'_n\}$ of primed variables. Let $V(X)$ denote the set of valuations over X. The *projection* of v to variables $\bar{X} \subseteq X$ is $v{\downarrow}_{\bar{X}} = \{x \to v(x) | x \in \bar{X}\}$. The *embedding* of a set $U \subseteq V(X)$ into variables $\bar{X} \supseteq X$ is the largest subset of $V(\bar{X})$ whose projection is in U, written as $U|^{\bar{X}}$. When a valuation u over X and a valuation v over $\bar{X}$ *agree*, i.e., $u{\downarrow}_{X \cap \bar{X}} = v{\downarrow}_{X \cap \bar{X}}$, we use $u \sqcup v$ to denote the valuation w defined by $w{\downarrow}_X = u$ and $w{\downarrow}_{\bar{X}} = v$. Arithmetic operations on valuations are defined in the straightforward way. An *activity* over

X is a function $f : \mathbb{R}^{\geq 0} \rightarrow V(X)$. Let $Acts(X)$ denote the set of activities over X. The *derivative* $\dot{f}$ of an activity f is an activity over $\dot{X}$, defined analogously to the derivative in $\mathbb{R}^n$. The extension of operators from valuations to activities is done pointwise. Let $const_X(Y) = \{(v, v')|v, v' \in V(X), v{\downarrow_Y} = v'{\downarrow_Y}\}$. The convex hull of a set of valuations S written as $chull(S)$.

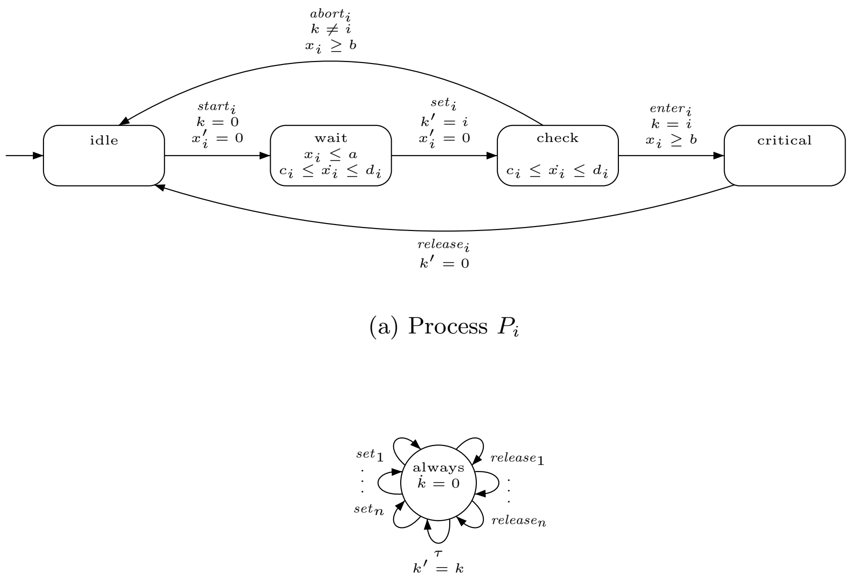
Definition 1 (Hybrid Automaton). (modified from [4]) *A* hybrid automaton *(HA) $A = (Loc, (X, O, C), Lab, Edg, Flow, Inv, Init)$ consists of:*

- *A finite set Loc called* locations.
- *A finite set called* variables *X, a subset C of X called* controlled *variables and a subset $O \subseteq C$ called* output *variables. Let $I = X \setminus C$ be the* input *variables and $E = I \cup O$ the* external *variables. A pair $p = (l, v)$ of a location and a valuation over X is a* state *of the automaton and the* state space *is $S_H = Loc \times V(X)$. For a state $p = (l, v)$ we define $loc(p) := l$ and $val(p) := v$. For a set of variables Y, let $val_Y(p) := v{\downarrow_Y}$.*
- *A finite set Lab of synchronization labels including the* stutter label *τ.*
- *A finite set Edg of edges called* transitions*. Each transition $e = (l, a, \mu, l')$ consists of a* source*, respectively* target *locations $l, l' \in Loc$, a synchronization label $a \in Lab$, and a* jump relation *$\mu \subseteq V(X)^2$. We require that for every location $l \in Loc$ there is a* stutter transition *$(l, \tau, const_X(C), l) \in Edg$.*
- *A set $Flow \subseteq Loc \times V(X \cup \dot{X})$ called* flows.
- *A set $Inv \subseteq Loc \times V(X)$ called* invariant.
- *A set $Init \subseteq Inv$ called* initial states.

A class of hybrid automata of particular interest are *linear hybrid automata* (LHA), since they can be analyzed using simple polyhedral computations [7]. LHA are defined as follows. A *linear constraint* over a set of variables $X = \{x_1, \ldots, x_n\}$ is of the form $\sum_i a_i x_i \bowtie b$, where $\bowtie \in \{<, \leq\}$. A (convex) *linear predicate* is a (conjunctive) boolean combination of linear constraints. A linear hybrid automaton has invariants and initial states defined in each location by a linear predicate over the variables, jump relations defined by a linear predicate over $X \cup X'$, and flow valuations defined by convex linear predicates over $\dot{X}$.

Definition 2 (Parallel Composition). [4] *Hybrid automata H_1, H_2 are* compatible *if $C_1 \cap C_2 = \emptyset$, $X_1 \cap C_2 \subseteq O_2$ and $X_2 \cap C_1 \subseteq O_1$. The* parallel composition *of compatible hybrid automata H_1, H_2 is the hybrid automaton H with*

- *$Loc = Loc_1 \times Loc_2$,*
- *$X = X_1 \cup X_2$, $C = C_1 \cup C_2$, $O = O_1 \cup O_2$, $Lab = Lab_1 \cup Lab_2$*
- *$((l_1, l_2), a, \mu, (l_2', l_2')) \in Edg$ iff*
  - *$(l_1, a_1, \mu_1, l_1') \in Edg_1$ and $(l_2, a_2, \mu_2, l_2') \in Edg_2$*
  - *either $a = a_1 = a_2$, or $a = a_1 \notin Lab_2$ and $a_2 = \tau$, or $a_1 = \tau$ and $a = a_2 \notin Lab_1$,*
  - *$\mu = \{(v, v')|(v{\downarrow_{X_i}}, v'{\downarrow_{X_i}}) \in \mu_i\}$;*
- *$Flow(l_1, l_2) = Flow_1(l_1)|^{X \cup \dot{X}} \cap Flow_2(l_2)|^{X \cup \dot{X}}$;*
- *$Inv(l_1, l_2) = Inv_1(l_1)|^X \cap Inv_2(l_2)|^X$;*
- *$Init(l_1, l_2) = Init_1(l_1)|^X \cap Init_2(l_2)|^X$.*



(a) Process P_i

(b) Shared variable S

Fig. 1. Compositional model of timing based mutual-exclusion protocol in [12]

Example 1. Consider the model of a timing based mutual-exclusion protocol shown in Fig. 1. In every location l of P_i, there is a transition (l, τ, μ, l) with $\mu = \{(v, v')|v(x_i) = v'(x_i), v(k), v'(k) \in \mathbb{R}\}$ (omitted from the figure). The system is considered *safe* if there are never two or more processes in the critical section at the same time. It is a compositional adaptation of the model given in [12], and parameterized to n processes with time constants c_i and d_i that represent the minimal, respectively maximal, skew of their clocks. The processes P_i have a controlled variable x_i to model their local clock and an input variable k that models a semaphore. Because none of the processes controls k, it is modeled separately in an automaton S we call a *shared variable model.* S has k as a controlled variable and fixes its derivative to zero. It gives the processes access to k by synchronizing on transitions that wish to change the value of k. Note that it does not restrict the change of k in these transitions.

Semantics We define the semantics of hybrid automata with *runs*, which we construct from *atomic runs* that represent a period of elapsing time followed by a (discrete) transition. The change of variables over time is described by an admissible activity. An activity $f(t) \in Acts(X)$ is called *admissible* over an interval $[0, \delta]$ in a location l if $\delta = 0$, or $\forall t, 0 \leq t \leq \delta : f(t) \in Inv(l), f(t) \sqcup \dot{f}(t) \in$

Flow(l). In weak runs, we consider τ-transitions that do not change the variables as unobservable.

Definition 3 (Run). *An* atomic run $\sigma = p \xrightarrow{\delta,f,a} p'$ *consists of* source and target states p, p', *a* duration $\delta \in \mathbb{R}^{\geq 0}$, *an activity f over X called* witness *and a label $a \in Lab$ such that*

- *$p, p' \in Inv$,*
- *f is differentiable and admissible over $[0, \delta]$ in $loc(p)$ and $f(0) = val(p)$,*
- *there is a transition $(loc(p), a, \mu, loc(p')) \in Edg$ with $(f(\delta), val(p')) \in \mu$.*

A run *of a hybrid automaton H is a finite or infinite sequence*

$$\sigma = p_0 \xrightarrow{\delta_0,f_0,a_0} p_1 \xrightarrow{\delta_1,f_1,a_1} p_2 \ldots$$

such that $\sigma_i = p_i \xrightarrow{\delta_i,f_i,a_i} p_{i+1}$ is an atomic run for all $i \geq 0$. For a finite run, its length *is the number of atomic runs in the sequence. A* weak atomic run $\sigma^w = p \overset{\delta,f,a}{\twoheadrightarrow} p'$ *exists iff there is a finite run σ*

$$\sigma = p_0 \xrightarrow{\delta_0,f_0,\tau} p_1 \xrightarrow{\delta_1,f_1,\tau} \ldots \xrightarrow{\delta_{n-2},f_{n-2},\tau} p_{n-1} \xrightarrow{\delta_{n-1},f_{n-1},a} p_n$$

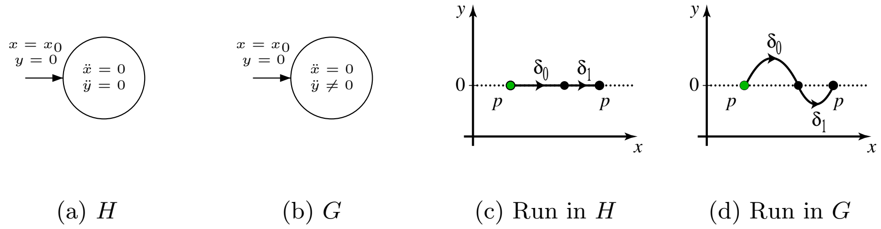
such that $\sum_{k=0}^{n-1} \delta_k = \delta$ and for all i,t, $0 \leq i < n-1$, $t_{i-1} \leq t \leq t_i$, holds $f(t) = f_i(t - t_{i-1})$, with $t_{-1} = 0$ and $t_i = \sum_{k=0}^{i} \delta_k$ for $0 \leq i$. A weak run *is defined analogously to a run as a sequence of weak atomic runs. A weak atomic run with all states in the same location l is called* unilocational, *and denoted by $p \overset{\delta,f,a}{\twoheadrightarrow}_l p'$.*

Remark 1. Due to the stutter transitions, there exists a run $p \xrightarrow{0,f,\tau} p$ in every state $p \in Inv$ and for every activity f with $f(0) = val(p)$. To underline that the activity is of no relevance, we may write $p \xrightarrow{0,\cdot,\tau} p$ instead.

Remark 2. All except the last transition of a weak atomic run leave the variables unchanged, since $f_i(t_i - t_{i-1}) = f(t_i) = f_{i+1}(t_i - t_i)$ for $0 \leq i < n-2$.

3 Simulation Relations

To express that a hybrid automaton G is a valid abstraction of a hybrid automaton H (or equivalently that H refines G) one can establish a *simulation relation* over the product of their states. It relates a state in H to those in G that have the same, or more, behavior. Two types of simulation are predominant in literature: *trace simulation* compares the exact trace between source and target states, while *timed simulation* only considers how much time passed to get from one to the other. *Weak* versions of simulation are defined over weak traces. They are often used to show that a location with complex dynamics can be overapproximated by several locations with simpler dynamics that are connected with τ-transitions, e.g., in [13]. To be consistent with compositionality, two hybrid automata can only be compared if they have comparable inputs and outputs.



(a) H (b) G (c) Run in H (d) Run in G

Fig. 2. H is not trace simulated by G, but timed simulated

Definition 4. *H is* comparable *with G if $X_H = X_G$, $Lab_H = Lab_G$, $C_H \subseteq C_G$ and $O_H = O_G$.*

Note that according to this definition G may use less inputs than H, but not more.

Definition 5 (Trace Simulation). *A relation $R \subseteq S_H \times S_G$ is a* trace simulation relation *between comparable H and G iff for all $(p, q) \in R$, δ, f, a, p',*

$$p \xrightarrow{\delta,f,a}_H p' \quad \Rightarrow \quad \exists g, q' : q \xrightarrow{\delta,g,a}_G q' \wedge (p', q') \in R \wedge \forall t : f(t){\downarrow_{E_G}} = g(t){\downarrow_{E_G}} \ .$$

We write $H \preceq_t G$ iff there exists a trace simulation relation R such that $Init_H \subseteq R^{-1}(Init_G)$. R is called the witness *to the simulation.*

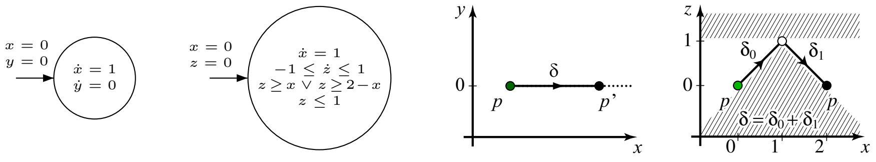
Definition 6 (Timed Simulation). *A relation $R \subseteq S_H \times S_G$ is a* timed simulation relation *between comparable H and G iff for all $(p, q) \in R$, δ, f, a, p',*

$$p \xrightarrow{\delta,f,a}_H p' \quad \Rightarrow \quad \exists g, q' : q \xrightarrow{\delta,g,a}_G q' \wedge (p', q') \in R \wedge f(0){\downarrow_{E_G}} = g(0){\downarrow_{E_G}} \ .$$

We write $H \preceq_0 G$ iff there exists a timed simulation relation R such that $Init_H \subseteq R^{-1}(Init_G)$. R is called the witness *to the simulation.*

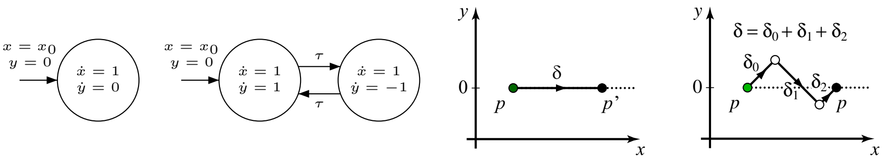
Timed simulation forces G to have an activity that matches in the source and target states of an atomic run. It is, however, not guaranteed that H and G take the same path in between, as the following example demonstrates.

Example 2 (Trace vs. timed simulation). Consider H and G shown in Fig. 2 with $X_H = X_G = \{x, y\}$, $O_H = O_G = \{x\}$ (τ-transitions not shown). Recall that restrictions on the activities of input y are allowed. H has only trajectories in the form of straight lines, while G can nondeterministically chose any parabola with nonzero curvature. Consequently, G can not exactly match the atomic runs of H and the conditions for trace simulation are violated, i.e., $H \not\preceq_t G$. However, for any given atomic run in H, G has an atomic run that, while not being identical over all points in time, matches in the timed sense, i.e., source and target states are equal and takes the same time to get from source to target. In any atomic run that might follow, G can chose a new parabola with a new curvature that matches in the timed sense. As result, $H \preceq_0 G$.



(a) H (b) G (c) Run in H (d) Run in G

Fig. 3. H is not trace simulated by G, but weakly trace simulated



(a) H (b) G (c) Run in H (d) Run in G

Fig. 4. H is not timed simulated by G, but weakly timed simulated

Often it is useful to consider τ-transitions unobservable in the comparison. This is achieved by looking at weak atomic runs instead of atomic runs:

Definition 7 (Weak Simulation). Weak *trace (timed) simulation is defined analogously to trace (timed) simulation over weak atomic runs, and denoted by* $H \preceq_t^w G$ *(*$H \preceq_0^w G$*).*

Weak atomic runs differ from atomic runs in two ways: The witnessing activity only has to be piecewise differentiable instead of differentiable, and the location can change during the period of time elapse. The following examples illustrate how this reflects in the automata that match in weak simulation, but not in simulation.

Example 3 (Weak trace simulation). Consider the LHA H and G shown in Fig. 3 with $X_H = C_H = \{x, y\}$, $X_G = C_G = \{x, z\}$, $O_H = O_G = \{x\}$. Consider the run of H shown in Fig. 3(c). Since y is not an external variable, both can take different activities with respect to y. G does not have any differentiable activity that matches because the only ones that do, e.g., a parabola from p to p', violate the flow constraint $|\dot{z}| \leq 1$. Consequently, $H \not\preceq_t G$. However, G does have a two-piece activity that can be represented by a weak atomic run, see Fig. 4(d), and $H \preceq_t^w G$.

Example 4 (Weak timed simulation). Consider the LHA H and G shown in Fig. 4 with $X_H = X_G = \{x, y\}$, $C_H = C_G = O_H = O_G = \{x\}$. Without taking τ-transitions, G can not match the activities in H, so $H \not\preceq_0 G$. However, $H \preceq_0^w G$ because every transition in H can be matched by a concatenation of transitions in G in which positive and negative change of y cancel each other out, as shown in Fig. 4(d).

We define the following equivalence relation based on simulation:

Definition 8 (Bisimulation). *A simulation relation R is a* bisimulation relation *between H and G iff R is simulation relation for $H \sim G$ and R^{-1} is a simulation relation for $G \sim H$, where $\sim \in \{\preceq_t, \preceq_0, \preceq_t^w, \preceq_0^w\}$. Bisimulation is denoted with $\cong_t, \cong_0, \cong_t^w, \cong_0^w$ depending on what relation was chosen for $\sim$.*

The different types of simulation introduced in this section are ordered with respect to how closely they distinguish behaviors of hybrid automata.

Proposition 1. *Simulation relations satisfy the following partial order:*

$$\begin{tikzcd}
H \preceq_t G \arrow[r, Rightarrow] \arrow[d, Rightarrow] & H \preceq_0 G \arrow[d, Rightarrow] \\
H \preceq_t^w G \arrow[r, Rightarrow] & H \preceq_0^w G
\end{tikzcd}$$

It will become apparent in the next section that the closer a simulation relation distinguishes behaviors, the larger is the class of hybrid automata for which it is compositional.

4 Compositionality

We identify subclasses of hybrid automata for which simulation is compositional. To do so we must show that the behavior of composed automata implies matching behavior of their composed specifications. Zero-duration atomic runs match for all the above types of simulation [8], so we can focus on continuous activities.

Definition 9. *A relation $\sim$ over hybrid automata is* compositional *iff*

$$H_1 \sim G_1 \wedge H_2 \sim G_2 \quad \Rightarrow \quad H_1||H_2 \sim G_1||G_2.$$

We will also use the following equivalent formulation of compositionality:

Lemma 1. *A preorder $\sim$ is* compositional *iff $H \sim G \;\Rightarrow\; H||M \sim G||M$.*

Compositionality is enforced by the fact that variables are controlled by at most one automaton.

Example 5. Consider the mutual-exclusion protocol of Ex. 1. In a noncompositional model, such as the one in [12], the analysis of n processes yields that $P_1|| \dots ||P_n$ is safe. However, this does not imply that $P_1|| \dots ||P_n||M$ is safe. M could reset k at the wrong time and cause more than one process to enter the critical section. In contrast, the compositional model does not allow M to change

k in any way that is not already contained in S. Any transitions that attempt this will be blocked by the composition operator since it imposes that transitions of M either synchronize with existing transitions or with τ-transitions, which have the jump relation $const_X(C)$ and therefore leave k constant.

The simulation relations in this paper are preorders, which is easy to show using proofs similar to those in [10]. Consequently, we can use Lemma 1 to show compositionality.

Proposition 2. *Trace and timed simulation, as well as their weak variants, are preorders for comparable hybrid automata.*

If H is trace simulated by G, the external part of any activity in H must be matched exactly by an activity in G. Because M can inhibit only those same external variables, any activity in $H||M$ entails a matching activity in G. Compositionality is a direct consequence.

Proposition 3. *Trace and weak trace simulation are compositional.*

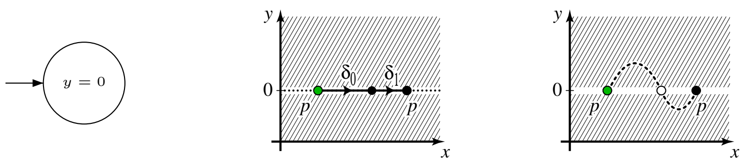
Proof. (Sketch) The compositionality of trace simulation was already shown in [6], but not that of weak trace simulation. We extend this result to weak trace simulation by showing that a weak atomic run in $H||M$ implies a weak atomic run in $G||M$ such that its target state is in the simulation relation. Our proof follows the structure of the one in [6] and relies strongly on the presence of stuttering steps. Let R_0 be the witnessing simulation relation for $H \preceq_t^w G$. We show that

$$R = \{(((l,m),x),((k,m),y)) \mid ((l,x{\downarrow_{X_H}}),(k,y{\downarrow_{X_G}})) \in R_0, x{\downarrow_{X_M}} = y{\downarrow_{X_M}}\}$$

is a witness to $H||M \preceq_t^w G||M$. A weak atomic run $\sigma_{H||M}$ in $H||M$ can be projected to weak atomic runs σ_H and σ_M in H, respectively M. Because G weakly trace simulates H, σ_H implies that there exists a weak atomic run σ_G in G with a matching activity and a matching jump at the end. Now σ_G and σ_M can be padded to have τ-transitions at identical intervals. Since G and H show the same external behavior in σ_G and σ_H, σ_G can be composed with the run σ_M to yield a weak atomic run in $G||M$. Since the external variables in the target states of $\sigma_{H||M}$ and $\sigma_{G||M}$ have the same values, the target states are in R. This shows that R is a simulation relation. It is straightfoward to show $Init_{H||M} \subseteq R^{-1}(Init_{G||M})$, which concludes the proof. □

Timed simulation only forces G to have an activity that matches in the source and target states of an atomic run. It is not guaranteed that H and G take the same path in between, as the following example demonstrates.

Example 6 (Timed simulation and compositionality). Consider H and G shown in Fig. 2. In Ex. 2 we showed that $H \preceq_0 G$. If timed simulation were compositional, Lemma 1 says that simulation should still hold if we compose both sides with any K. Consider K from Fig. 5(a), with $X_K = C_K = \{y\}$. In $G||K$ the invariant $y = 0$ does not allow any timed transitions of nonzero duration, while in $H||K$ time can elapse forever, as illustrated in Fig. 5. Consequently, $H||K \not\preceq_0 G||K$.



(a) K (b) Run in $H||K$ (c) No run in $G||K$

Fig. 5. Timed simulation for H, G from Fig. 2 is not compositional with K

Since timed simulation abstracts the exact activites away it is, in general, not compositional. We now show that it is, however, compositional for LHA with convex invariants. In the proof we use a lemma from [12], which states that if there is any admissible activity, there is also a linear one:

Lemma 2. (adapted from [12]) *Let l be a location of any linear hybrid automaton with a convex invariant $Inv(l)$, and $v, v' \in Inv(l)$ be any valuations inside it. If there exists an activity f that is admissible in l over some interval $[0, \delta]$ and $f(0) = v$, $f(\delta) = v'$ then $f'(t) = v + t/\delta(v' - v)$ is an equally admissible activity.*

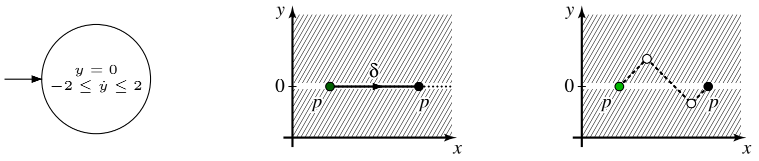
As a consequence of this lemma, whenever there are activities with identical source and target states in LHA with convex invariants H and G, there is also a linear activity that is admissible in both automata. From the existence of two different activities we can thus infer the existence of a common activity, which immediately leads to compositionality:

Proposition 4. *Timed simulation is compositional for LHA with convex invariants.*

Proof. Timed simulation is compositional for compatible automata H, M if for any atomic runs $(k, u) \xrightarrow{\delta,f,\tau}_H (k', u')$ and $(l, v) \xrightarrow{\delta,g,\tau}_M (l', v')$ with $u{\downarrow}_{X_H \cap X_M} = v{\downarrow}_{X_H \cap X_M}$ and $u'{\downarrow}_{X_H \cap X_M} = v'{\downarrow}_{X_H \cap X_M}$ there is an admissible differentiable activity h in location (k, l) of $H||M$ with $h(0){\downarrow}_{X_H} = u$, $h(0){\downarrow}_{X_M} = v$ and $h(\delta){\downarrow}_{X_H} = u'$, $h(\delta){\downarrow}_{X_M} = v'$ [8]. If H, M are LHA with convex invariants, there exist, according to Lemma 2, linear activities f' and g' that witness $(k, u) \xrightarrow{\delta,f',\tau}_H (k', u')$ and $(l, v) \xrightarrow{\delta,g',\tau}_M (l', v')$. Since $f'{\downarrow}_{X_H \cap X_M} = g'{\downarrow}_{X_H \cap X_M}$, the activity h defined by $h{\downarrow}_{X_H} = f'$, $h{\downarrow}_{X_M} = g'$ is differentiable and admissible in $H||M$. □

We will later discuss compositionality of LHA with nonconvex invariants using weak simulation.

If one admits weak atomic runs in timed simulation, i.e., regards τ-transitions as unobservable, compositionality is lost even for LHA. We show this with the following counterexample.



(a) K (b) Run in $H||K$ (c) No run in $G||K$

Fig. 6. Weak timed simulation for H, G from Fig. 4 is not compositional with K

Example 7 (Non-compositional LHA for weak timed simulation). Consider the LHA H and G from Ex. 4, shown in Fig. 4. $H \preceq_0^w G$ because every atomic run in H can be matched by a concatenation of atomic runs in G in which positive and negative change of y cancel each other out. Now consider the composition of H and G with K shown in Fig. 6(a), with $X_K = C_K = \{y\}$. For $H||K$ time can elapse forever, while for $G||K$ the invariant $y = 0$ does not allow any atomic runs of nonzero duration, as illustrated in Fig. 6. Consequently, $H||K \not\preceq_0^w G||K$.

We now identify a class of hybrid automata for which weak timed simulation is compositional. The alternation of τ-transitions with passing time allows an automaton to asymptotically mimic any activity that is a convex piecewise combination of admissible activities. Our compositional class is simply one for which we know that all the activities that the automaton can mimic are actually admissible, possibly in another location. The relevant τ-transitions in a weak atomic run do not change the variables, see Remark 2. The mimicking must therefore take place in the vicinity of the intersection of two invariants that are connected with τ-transitions. We demand that any such mimicking can take place entirely within one location, formally as follows:

Definition 10. *A hybrid automaton H is* overlap-closed *if for any (l, u) there is a $\delta_{max}(l, u)$ such that $(l, u) \xrightarrow{\delta,f,\tau} (l', u')$ with $\delta \leq \delta_{max}(l, u)$ implies*

(i) a run $(l, u) \xrightarrow{\delta,f,\tau}_l (l, u') \xrightarrow{0,\cdot,\tau} (l', u')$, or
(ii) a location k such that $(l, u) \xrightarrow{0,\cdot,\tau} (k, u) \xrightarrow{\delta,f,\tau}_k (k, u') \xrightarrow{0,\cdot,\tau} (l', u')$.

H is strongly overlap-closed *if* $\inf_{l,u} \delta_{max}(l, u) > 0$.

Remark 3. Note that any hybrid automaton is overlap-closed if it does not have different locations connected by τ-transitions.

According to Lemma 2, LHA with convex invariants always have a linear activity between two points of the same location. Combining this fact with the assumption of overlap-closedness, we can conclude that if $H \preceq_0^w G$, a weak run in H is matched in G with a weak run witnessed by the same external activity. From there it is straightforward to show compositionality as follows:

Proposition 5. *Let H_1, H_2 be LHA and G_1, G_2 be strongly overlap-closed LHA with convex invariants and bounded derivatives. If $H_1 \preceq_0^w G_1$ and $H_2 \preceq_0^w G_2$, then $H_1||H_2 \preceq_0^w G_1||G_2$.*

Proof. (Sketch) Let R_1, R_2 be witnessing weak simulation relations for $H_1 \preceq_0^w G_1$ and $H_2 \preceq_0^w G_2$, respectively. We show that

$$R = \{(((k_1, k_2), x), ((l_1, l_2), y)) \mid ((k_i, x{\downarrow_{H_i}}), (l_i, y{\downarrow_{G_i}})) \in R_i \text{ for } i = 1, 2\}$$

is a witnessing simulation relation for $H_1||H_2 \preceq_0^w G_1||G_2$. The containment of initial states in R follows straightforwardly from the containment in R_1 and R_2. It remains to demonstrate that for any pair of states in R, a weak atomic run in $H_1||H_2$ implies a matching weak atomic run in $G_1||G_2$ such that the target states are again in R. A weak atomic run $p \xrightarrow{\delta,f,\alpha}\!\!\!\!\rightarrow p'$ can be split in two: a run $p \xrightarrow{\delta,f,\tau}\!\!\!\!\rightarrow p''$ containing only τ-transitions that leave the variables unchanged (see Remark 2) and a run $p'' \xrightarrow{0,f,\alpha}\!\!\!\!\rightarrow p'$ of zero duration. The definition of weak simulation for a run of zero duration is the same as that of timed simulation, so with Prop. 4 we can deduce that the latter part satisfies compositionality. The rest of the proof is therefore concerned with the former part of the run, which only includes τ-transitions that do not change the variables.

Because the H_i and H are LHA, a weak atomic run in $H_1||H_2$ has a witnessing run whose activity is piecewise linear [12]. We pad it with τ-transitions to obtain a run

$$\sigma_H = r_0 \xrightarrow{\delta_0, f_0, \tau} r_1 \xrightarrow{\delta_1, f_1, \tau} \ldots \xrightarrow{\delta_{n-2}, f_{n-2}, \tau} r_{n-1} \xrightarrow{\delta_{n-1}, f_{n-1}, \tau} r_n$$

with durations $\delta_j \leq \delta_{min}$ for some arbitrarily small $\delta_{min} > 0$, and linear activities f_1. Every one of the atomic runs $\sigma_{j,H} = r_j \xrightarrow{\delta_j, f_j, \tau} r_{j+1}$ in σ_H projects in the H_i onto corresponding runs σ_{j,H_i}. Since H_i is weakly simulated by G_i, there must be matching weak runs σ^w_{j,G_i}, i.e., with the same valuations of the external variables in the source and target state. Let $\delta_{min} = \inf_{l,u} \delta_{max}(l, u)$ from Def. 10. According to Def. 10, this implies that there is also matching weak run $\bar{\sigma}^w_{j,G_i}$ with all time-elapse inside a single location. Because the G_i are LHA with convex invariants, it follows from Lemma 2 that the linear activity between source and target state is also admissible in that location, thus matching the one in H_i.

It remains to show that the runs in G_1 and G_2 compose to a run in $G_1||G_2$, and that the target state of this run lies in R. Recall that the source and target states of σ_{j,H_i} and $\bar{\sigma}^w_{j,G_i}$ lie in R_i, which means they have the same values in the shared external variables. The shared variables of H_1 and H_2 also have the same values in the respective states, and due to comparability the same holds for the shared variables of G_1 and G_2. By padding with τ-transitions, we can obtain witnessing nonatomic runs for $\bar{\sigma}^w_{j,G_1}$ and $\bar{\sigma}^w_{j,G_2}$ that have the same length and whose atomic runs have the same duration. Because all the nonzero activities are linear, the source and target states still match. Consequently, the runs in G_1 and G_2 compose to a run in $G_1||G_2$. Because the target states of the runs in H_i and G_i lie in R_i, the target states of the runs in $H_1||H_2$ and $G_1||G_2$ lie in R. $\square$

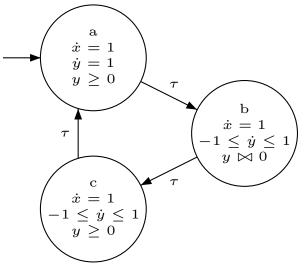


Fig. 7. LHA G', overlap-closed if $\bowtie = \geq$

Example 8 (Compositional LHA for weak timed simulation). [1] Consider the LHA H from Ex. 4, shown in Fig. 4, and G' from Fig. 7, with $X_{G'} = \{x, y\}$, $C_{G'} = O_{G'} = \{x\}$. G' is overlap-closed for the sign $\bowtie = \geq$, and not overlap-closed if $\bowtie = >$. In both cases $H \preceq_0^w G'$ because every atomic run in H can be matched by a concatenation of atomic runs in G in which positive and negative change of y cancel each other out. Now consider the composition of H and G' with K shown in Fig. 6(a), with $X_K = C_K = \{y\}$. In $G'||K$ with $\bowtie = \geq$, there are silent transitions to location c, where time can elapse forever, and consequently $H||K \preceq_0^w G'||K$. If $\bowtie = >$, there is no run from the initial location a to location c in $G'||K$, because the invariant of location b is empty. Consequently, $H||K \not\preceq_0^w G'||K$.

For weak runs, trace simulation has the advantage over timed simulation because it is compositional for any hybrid automata. In [1], timed simulation was used to show that the invariant of a hybrid automaton can be partitioned into arbitrarily small parts with a splitting operation that does not modify the behavior. This is useful in many applications, e.g., to transform nonconvex into convex invariants, or to overapproximate the automaton with one of simpler dynamics [1]. The splitting operation is defined as follows:

Definition 11 (Invariant split). (modified from [1]) *An* (open) split $\mathcal{S}$ *for a hybrid automaton* H *maps each location* l *to a finite set* $\{S_1^l, \ldots, S_k^l\}$ *of sets of valuations over* X *such that there exists a finite (open) cover* $\mathcal{O}^l = \{O_1^l, \ldots, O_k^l\}$ *of* $Inv(l)$ *with* $S_i^l = Inv(l) \cap O_i^l$ *for* $i = 1, \ldots, k$*. The* split *of* H *along* $\mathcal{S}$ *is the hybrid automaton* $split(H, \mathcal{S}) = (Loc_{\mathcal{S}}, (X, C, O), Lab, \rightarrow_{\mathcal{S}}, Flow_{\mathcal{S}}, Inv_{\mathcal{S}}, Init_{\mathcal{S}})$ *with*

- $Loc_{\mathcal{S}} = \{(l, S) \mid l \in Loc, S \in \mathcal{S}(l)\}$,
- $\rightarrow_{\mathcal{S}} = \{((l, S), a, \mu, (l', S')) \mid (l, a, \mu, l') \in \rightarrow\}$,
- $Flow_{\mathcal{S}}((l, S)) = Flow(l)$, $Inv_{\mathcal{S}}((l, S)) = Inv(l) \cap S$, $Init_{\mathcal{S}}((l, S)) = Init(l) \cap S$.

We rephrase the following results of [1] and [3] using weak trace simulation, thus expanding their applicability to the context of compositional reasoning.

[1] Thanks to the anynomous reviewer who inspired the example.

Proposition 6. *For any H, $H \cong_t^w$ split$(H, \mathcal{S})$ if $\mathcal{S}$ is an open split or the admissible activities of H are analytic functions.*

Proof. In [1], it is shown that $H \cong_0^w$ *split*$(H, \mathcal{S})$ if $\mathcal{S}$ is an open split. While timed simulation is used formally, the corresponding proof shows that the activities match identically over time. It is therefore straightforward to strengthen the result to weak trace bisimulation. In [3] it is shown, based on the results of [1], that the split does not have to be open if the admissible activities of H are analytic functions. □

The condition of analytic activities applies, e.g., to LHA, or hybrid automata with affine dynamics [3], whose flows are defined by conjunctions of linear constraints over $X \cup \dot{X}$.

5 Conclusions

Timed and weak timed simulation are often used to show equivalence and abstraction between hybrid automata. We identify the following subclasses of linear hybrid automata (LHA) for which these relations are compositional: LHA with convex invariants for timed simulation, and strongly overlap-closed LHA with convex invariants and bounded derivatives for weak timed simulation. An advantage of timed simulation relations is that for many LHA they can be computed, e.g., with PHAVer [9,8,2]. In addition, LHA can overapproximate any hybrid automata arbitrarily close [1]. Using the above results we can overapproximate with a compositional subclass of LHA, and thus apply compositional and assume/guarantee-reasoning to arbitrary hybrid automata.

On the downside, timed simulation is not compositional in general. Weak trace simulation, which is compositional for hybrid automata with arbitrary dynamics, can sometimes be used instead. E.g., one may substitute it for weak timed simulation in the proofs of [3,1] without having to change any essential parts of the proofs. The result is a notion of equivalence that is stronger per se and compositional. In future work we will identify subclasses for which timed simulation implies trace simulation.

6 Acknowledgements

The author wishes to thank Bruce Krogh and Oded Maler for their indispensable guidance and support, as well as Susanne Graf for her valuable comments. A special thanks goes to the anonymous reviewer who spotted a bug in Prop. 5.

References

1. Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Trans. Automatic Control*, 43(4):540–554, 1998.

2. Goran Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. In Manfred Morari and Lothar Thiele, editors, *Hybrid Systems: Computation and Control (HSCC'05), Mar. 9–11, 2005, Zürich, CH*, volume 2289 of *LNCS*. Springer, 2005. PHAVer is available at `http://www.cs.ru.nl/~goranf/`.
3. Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Automatic rectangular refinement of affine hybrid systems. In *Proc. FORMATS'05*, volume 3829 of *LNCS*, pages 144–161. Springer, 2005.
4. Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A. Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theor. Comp. Science*, 138(1):3–34, 1995.
5. Erika Ábrahám-Mumm, Ulrich Hannemann, and Martin Steffen. Verification of hybrid systems: Formalization and proof rules in PVS. In *Proc. IEEE Int. Conf. on Engineering of Complex Computer Systems (ICECCS 2001)*, June 2001.
6. Nancy A. Lynch, Roberto Segala, and Frits W. Vaandrager. Hybrid I/O automata. *Information and Computation*, 185(1):105–157, 2003.
7. Thomas A. Henzinger. The theory of hybrid automata. In *Proc. 11th Annual IEEE Symposium on Logic in Computer Science, LICS'96, New Brunswick, New Jersey, 27-30 July 1996*, pages 278–292. IEEE Computer Society Press, 1996.
8. Goran Frehse, Zhi Han, and Bruce H. Krogh. Assume-guarantee reasoning for hybrid i/o-automata by over-approximation of continuous interaction. In *Proc. IEEE Conf. Decision & Control (CDC'04), Dec. 14–17, 2004, Atl., Bahamas*, 2004.
9. Goran Frehse. Compositional verification of hybrid systems with discrete interaction using simulation relations. In *Proc. IEEE Conf. Computer-Aided Control System Design (CACSD'04), September 1–4, 2004, Taipei, Taiwan*, 2004.
10. Goran Frehse. *Compositional Verification of Hybrid Systems using Simulation Relations*. PhD thesis, Radboud Universiteit Nijmegen, October 2005.
11. Nancy A. Lynch and Michael J. Fischer. On describing the behavior and implementation of distributed systems. *Theoretical Computer Science*, 13(1):17–43, 1981. Special issue on Semantics of Concurrent Computation.
12. Rajeev Alur, Thomas A. Henzinger, and Pei-Hsin Ho. Automatic symbolic verification of embedded systems. *IEEE Trans. Soft. Engineering*, 22:181–201, 1996.
13. Thomas A. Henzinger and Howard Wong-Toi. Linear phase-portrait approximations for nonlinear hybrid systems. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems III: Verification and Control*, volume 1066 of *LNCS*, pages 377–388. Springer, 1996.