Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Approximate timed abstractions of hybrid automata

A. D'Innocenzo$^\flat$, A.A. Julius$^\sharp$, M.D. Di Benedetto$^\flat$, G.J. Pappas$^\sharp$

***Abstract*— Given a hybrid automaton and a desired precision, we aim at constructing an approximate abstraction by means of a timed automaton, whose discrete state trajectories approximate the discrete state trajectories of the original system, with the desired precision on switching times. We show that using the Euclidian metric on reals it is not always possible to construct a timed automaton that is *close* to a hybrid automaton with finite precision. For this reason, we motivate and introduce *relative* metrics on reachability time, external language and simulation relation to quantify the precision of our abstraction. Our main result is to propose a novel algorithm to construct a timed automaton that is an approximate timed abstraction of a hybrid automaton with desired precision, and study its convergence properties. For an extended version of this paper refer to [11].**

I. Introduction

Systems characterized by discrete and continuous aspects in their dynamics are called hybrid systems. Hybrid systems are very general as they include continuous and discrete systems as special cases. They are very useful in the analysis of embedded system, to design a digital controller so that a continuous plant satisfies prescribed specifications. Applications of hybrid systems range from biological systems to air traffic management systems, from automotive to communication systems. Their great expressive power has to be paid by the lack of strong theoretical results about their behavior, and consequent difficulties in verifying the properties of a closed loop system. In fact, formal verification (e.g. *model checking* [7]) of properties where the state space is semi–exhaustively searched are complicated by the very large dimensions of the state space. Reachability verification [13], [16], observability verification [5], [8], [10] and model checking [4], [18] for hybrid systems are intensely studied in the automatic control and computer science societies.

One important technique that people use to cope with this problem is *abstraction*. By abstraction, we create a system with smaller state space (even finite) that is equivalent to the original systems. System equivalence is usually defined by the notions of language equivalence and bisimulation [3], [25]. The classical exact notions of language equivalence and bisimulation are very restrictive, since they require perfect equivalence of trajectories. Recently, approximate notions of system equivalence [14], [15], [17], [20], [21] were developed to relax the abstraction problem, where a metric is introduced to quantify the distance between the original system and the abstraction. Other results in approximation theory for timed systems can be found e.g. in [6], [12]. To analyze temporal properties of hybrid automata, it is reasonable to consider as abstracting system the class of timed automata, that can generally be abstracted into finite state systems [3]. For this reason, verification algorithms are generally decidable for timed automata: model checking tools for timed automata are available (e.g. UPPAAL [23] and KRONOS [27]).

A procedure to translate a hybrid automaton into a rectangular automaton was proposed in [26]. The authors proposed in [10] an algorithm to construct a durational graph $\mathcal{G}$ from a hybrid automaton $\mathcal{H}$, and used the abstraction $\mathcal{G}$ to verify observability. However, no analysis of the *distance* between the timed executions of the discrete state of $\mathcal{H}$ and $\mathcal{G}$ was performed. On the basis of the approximation metrics defined in [15], [20], the main contribution of this paper is translating a hybrid automaton into a timed automaton whose discrete state trajectories approximate the discrete state trajectories of the hybrid automaton, with arbitrary desired precision ε on switching times. Our abstraction can be used to model check hybrid automata. It is well known that classical temporal logics such as CTL and LTL [7] are preserved by bisimulation relations, while simulation relations preserve their universal fragment (properties addressed to all executions). For temporal logics such as TCTL [1], that explicitly add time constraints in the formulae, it was proved in [17] that TCTL is robust w.r.t. approximate bisimulation relations. Namely if the abstraction satisfies a TCTL formula, then the original system satisfies an ε–close formula (and viceversa), with ε the precision of the abstraction. We use as abstract system a subclass of timed automata: the durational graphs. It was discussed in the literature that model checking for durational graphs is more efficient than for timed automata [22].

The paper is organized as follows. In section II, we introduce the reader to basic definition of non deterministic hybrid automata, timed automata and durational graphs. In section III, we introduce the framework of metric transition systems which enables us to model both hybrid automata and timed automata. We introduce a *relative metric* on positive reals, reachability time, external language and simulation relation to quantify the precision of our abstraction. Our metric is *relative* and not *absolute*, in the sense that it depends on the elapsed time: i.e. the distance between 1 and 2 seconds is considered the same as the distance between 100 and 200 seconds. A motivation for this choice is that

This work was partially supported by the HYCON Network of Excellence (c.n. FP6–IST–511368), by MIUR (project SCEF - PRIN05), and by NSF Presidential Early CAREER (PECASE, Grant 0132716).

$^\flat$ Department of Electrical Engineering and Computer Science, Center of Excellence DEWS. University of L'Aquila, Italy. email: (adinnoce,dibenede)@ing.univaq.it

$^\sharp$ Department of Electrical and Systems Engineering. University of Pennsylvania, USA. email: (agung@seas,pappasg@ee).upenn.edu

1-4244-1498-9/07/$25.00 ©2007 IEEE.

using Euclidian metric it is not always possible to construct a timed automaton that is *close* to a hybrid automaton with finite precision. Moreover, in practical applications it is often required to construct an approximation of a plant for a finite horizon analysis. In section IV, we propose a novel algorithm to construct a durational graph that is an approximate timed abstraction of a hybrid automaton. We study the convergence properties of the algorithm, and prove that the generated durational graph is approximately bisimilar to the hybrid automaton.

II. Basic definitions

One prominent theoretical framework that is used to model hybrid systems is proposed by Lygeros [24], where the discrete part consists of a labeled oriented graph, and the continuous part is described by a dynamical continuous system associated to each discrete state. The interaction between the continuous and discrete part is described by invariant, guard, and reset conditions. We consider here hybrid automata, that are hybrid systems characterized by dynamics without control input.

Definition 1 (Hybrid automaton): A hybrid automaton is a tuple $\mathcal{H} = (Q \times X, Q_0 \times X_0, U, \mathcal{E}, E, Inv, G, R)$ such that: $Q \times X$ is the hybrid state space, where Q is a finite set of discrete states $\{q_1, q_2, \cdots, q_N\}$, and $X \subseteq \mathbb{R}^n$ is the continuous state space; $Q_0 \times X_0 \subseteq Q \times X$ is the set of initial discrete and continuous conditions; $\{\mathcal{E}_q\}_{q \in Q}$ associates to each discrete state the continuous time–invariant dynamics $\mathcal{E}_q : \dot{x} = f_q(x, w)$, where $x \in \mathbb{R}^n$ and $w \in \mathbb{R}^p$ is a disturbance that takes value in a bounded set U. Given an initial condition $x(t_0) = x_0$ and a disturbance signal $w : \mathbb{R}_+ \cup \{0\} \to U$, we define the solution at time $t > t_0$ according to f_q by $x(t) = x_{f_q}(t, w|_{t_0}^{t}, x_0)$. The solution is unique with the assumption that f_q is assumed to be continuous with respect to time and Lipschitz continuous with respect to the dependent variables; $E \subseteq Q \times Q$ is a collection of edges; each edge $e \in E$ is an ordered pair of discrete states, the first component of which is the source and is denoted by $s(e)$, while the second is the target and is denoted by $t(e)$; $\{Inv_q\}_{q \in Q}$ associates to each discrete state an invariant set $Inv_q \subseteq X$; $\{G_e\}_{e \in E}$ associates to each edge a guard set $G_e \subseteq Inv_{s(e)}$; $\{R_e\}_{e \in E}$ associates to each edge a reset map $R_e : Inv_{s(e)} \to 2^{Inv_{t(e)}}$.

Notice that this class of hybrid automata is generally non deterministic. The continuous state evolves following non deterministic dynamics, and the discrete state evolution depends only on the continuous state according to the guards, that we assume non–intersecting. We denote $inc(q)$ the set of incoming edges to q and $out(q)$ the set of outgoing edges from q. An execution of a hybrid automaton [24] is a time evolution of the discrete and continuous states satisfying the continuous and discrete dynamics, and their interactions (invariant, guard and reset).

We call *durational graph* a *timed automaton* [2] characterized by only one clock that is reset to 0 for all edges:

Definition 2 (Durational graph): A durational graph is a hybrid automaton $(Q \times X, Q_0 \times X_0, \mathcal{E}, E, Inv, G, R)$ such that: $X = \mathbb{R}_+ \cup \{0\}$ is the continuous state space of the clock variable v; for each $q_0 \in Q_0$, the initial condition is given by $(q_0, 0)$; for each $q \in Q$, the continuous dynamics are defined by $\mathcal{E}_q : \dot{v} = 1$; for each $q \in Q$, the set Inv_q is a rectangular set[1]; for each $e \in E$, the set G_e is a rectangular set and $R_e(v) = \{0\}$.

By this definition, a durational graph can be defined as a tuple $\mathcal{G} = (Q, Q_0, E, Inv, G)$.

III. Approximation metrics for timed abstractions

We introduce the framework of metric transition systems [15], which enable us to model both hybrid automata and timed automata:

Definition 3 (Metric transition system): A labeled metric transition system with observations is a tuple $\mathcal{T} = (Q, Q_0, \Sigma, E, \Omega, \omega)$ that consists of a possibly infinite set Q of states, a possibly infinite set $Q_0 \subseteq Q$ of initial states, a possibly infinite set Σ of labels, a transition relation $E \subseteq Q \times \Sigma \times Q$, a possibly infinite set Ω of observations, an observation map $\omega : Q \to \Omega$ and metrics d_Σ, d_Ω on Σ and Ω.

In what follows, we write $q \xrightarrow{\sigma} q'$ to denote that $(q, \sigma, q') \in E$. We assume that the systems we consider are *non–blocking*, i.e. for all $q \in Q$ there exists at least an outgoing transition $q \xrightarrow{\sigma} q'$. We say that a transition system $\mathcal{T}$ is *deterministic*, if for all $q \in Q, \sigma \in \Sigma$ there exists at most a unique transition $q \xrightarrow{\sigma} q'$, and the set Q_0 contains a single element.

We use the framework of metric transition systems to analyze properties of hybrid automata and timed automata in the same mathematical setting. We are interested in timed abstractions, thus we consider in the rest of the paper transition systems where $\Sigma = \mathbb{R}_+ \cup \{0\}$ represents the continuous time basis. In this setting, a transition $q \xrightarrow{t} q'$ models that state q' can be reached from state q in time t.

A state trajectory of $\mathcal{T}$ is an infinite sequence of transitions $q_0 \xrightarrow{t_0} q_1 \xrightarrow{t_1} q_2 \cdots$, where $q_0 \in Q_0$. An external trajectory of $\mathcal{T}$ is a sequence of elements of $\Omega \times \Sigma \times \Omega$ of the form $\rho = \omega_0 \xrightarrow{t_0} \omega_1 \xrightarrow{t_1} \omega_2 \cdots$ if there exists a state trajectory of $\mathcal{T}$ such that $\forall i \in \mathbb{N}, \omega(q_i) = \omega_i$. We define the language generated by $\mathcal{T}$ as the set of all external trajectories of $\mathcal{T}$, and denote it as $\mathcal{L}^{\mathcal{T}}$.

Given a set $W \subset \Omega$, we define the set $\Lambda^{\mathcal{T}}(W) \subseteq 2^{\mathbb{R}_+ \cup \{0\}}$ of all time instants t such that there exists an execution of $\mathcal{T}$ that generates an observation in the set W at time t:

$$\Lambda^{\mathcal{T}}(W) = \Big\{ t \in \mathbb{R}_+ \cup \{0\} : \exists \{\omega_i \xrightarrow{t_i} \omega_{i+1}\}_{i \in \mathbb{N}} \in \mathcal{L}^{\mathcal{T}}, \exists j \in \mathbb{N}, \omega_j \in W, t = \sum_{k=0}^{j-1} t_k \Big\}.$$

[1] a rectangular set in $\mathbb{R}^n$ is any subset that can be defined by a finite union of cartesian products of intervals.

Let $h_{\overrightarrow{\Sigma}}$ and h_Σ denote the directed and undirected Hausdorff distance associated to the metric d_Σ. We define directed and undirected reachability metrics as follows:

$$\begin{aligned} d_{\overrightarrow{\mathcal{R}}}(\mathcal{T}_1,\mathcal{T}_2,W) &= h_{\overrightarrow{\Sigma}}\Big(\Lambda^{\mathcal{T}_1}(W),\Lambda^{\mathcal{T}_2}(W)\Big),\\ d_{\mathcal{R}}(\mathcal{T}_1,\mathcal{T}_2,W) &= h_{\Sigma}\Big(\Lambda^{\mathcal{T}_1}(W),\Lambda^{\mathcal{T}_2}(W)\Big). \end{aligned}$$

The properties of Hausdorff distance imply that the reachability metrics are pseudo–metrics on the set of metric transition systems, and

$$\begin{aligned} d_{\overrightarrow{\mathcal{R}}}(\mathcal{T}_1,\mathcal{T}_2,W)=0 &\Leftrightarrow cl(\Lambda^{\mathcal{T}_1}(W))\subseteq cl(\Lambda^{\mathcal{T}_2}(W)),\\ d_{\mathcal{R}}(\mathcal{T}_1,\mathcal{T}_2,W)=0 &\Leftrightarrow cl(\Lambda^{\mathcal{T}_1}(W))= cl(\Lambda^{\mathcal{T}_2}(W)). \end{aligned}$$

Let $\mathcal{X}(\Sigma,\Omega)$ be the set of external trajectories with labels in $\Sigma=\mathbb{R}_+\cup\{0\}$ and observations in Ω. Given $\rho_1=\{\omega_i^1 \xrightarrow{t_i^1} \omega_{i+1}^1\}_{i\in\mathbb{N}}, \rho_2=\{\omega_i^2 \xrightarrow{t_i^2} \omega_{i+1}^2\}_{i\in\mathbb{N}}\in\mathcal{X}(\mathbb{R}_+\cup\{0\},\Omega)$, we define an undirected distance $d_\mathcal{X}$:

$$d_\mathcal{X}(\rho_1,\rho_2)=\begin{cases}\sup\limits_{i\in\mathbb{N}} d_\Sigma\big(\sum\limits_{j=0}^{i} t_j^1,\sum\limits_{j=0}^{i} t_j^2\big) & \text{if } \forall i\in\mathbb{N}, \omega_i^1=\omega_i^2\\ +\infty & \text{otherwise}\end{cases}$$

Proposition 1: $d_\mathcal{X}$ is a metric on the set of external trajectories $\mathcal{X}(\mathbb{R}_+\cup\{0\},\Omega)$.

Let $h_{\overrightarrow{\mathcal{X}}}$ and $h_\mathcal{X}$ denote the directed and undirected Hausdorff distance associated to the metric $d_\mathcal{X}$. Since $\mathcal{L}^{\mathcal{T}_1}$ and $\mathcal{L}^{\mathcal{T}_2}$ are subsets of $\mathcal{X}(\mathbb{R}_+\cup\{0\},\Omega)$, we can define a language metric as the Hausdorff distance between two languages:

$$\begin{aligned} d_{\overrightarrow{\mathcal{L}}}(\mathcal{T}_1,\mathcal{T}_2) &= h_{\overrightarrow{\mathcal{X}}}(\mathcal{L}^{\mathcal{T}_1},\mathcal{L}^{\mathcal{T}_2})\\ d_{\mathcal{L}}(\mathcal{T}_1,\mathcal{T}_2) &= h_{\mathcal{X}}(\mathcal{L}^{\mathcal{T}_1},\mathcal{L}^{\mathcal{T}_2}) \end{aligned}$$

The directed distance between two languages $\mathcal{L}^{\mathcal{T}_1},\mathcal{L}^{\mathcal{T}_2}$ is ε if for any trajectory of $\mathcal{T}_1$, we can find an ε–close trajectory of $\mathcal{T}_2$ according to the metric $d_\mathcal{X}$. A consequence of the properties of the Hausdorff distance is the following:

$$\begin{aligned} d_{\overrightarrow{\mathcal{L}}}(\mathcal{T}_1,\mathcal{T}_2)=0 &\quad\Leftrightarrow\quad cl(\mathcal{L}^{\mathcal{T}_1})\subseteq cl(\mathcal{L}^{\mathcal{T}_2})\\ d_{\mathcal{L}}(\mathcal{T}_1,\mathcal{T}_2)=0 &\quad\Leftrightarrow\quad cl(\mathcal{L}^{\mathcal{T}_1})= cl(\mathcal{L}^{\mathcal{T}_2}) \end{aligned}$$

We use the definition of approximate simulation and bisimulation relations proposed by Julius and Pappas in [20]. Let $\mathcal{T}_1=(Q_1,Q_0^1,\Sigma_1,E_1,\Omega_1,\omega_1)$ and $\mathcal{T}_2=(Q_2,Q_0^2,\Sigma_2,E_2,\Omega_2,\omega_2)$ be two labeled metric transition systems with the same set of labels ($\Sigma_1=\Sigma_2=\Sigma=\mathbb{R}_+\cup\{0\}$) and the same set of observations ($\Omega_1=\Omega_2=\Omega$). Let d_Σ be defined as above, and let d_Ω be a metric on Ω.

Definition 4 (Approximate simulation relation): [20] A relation $\Gamma\subseteq Q_1\times Q_2$ is called a (ε,δ) approximate simulation relation of $\mathcal{T}_1$ by $\mathcal{T}_2$, if for all $(q_1,q_2)\in\Gamma$:

1) $d_\Omega(\omega_1(q_1),\omega_2(q_2))\le\delta$,
2) for all $q_1\xrightarrow{\sigma}q_1'$, there exists $q_2\xrightarrow{\sigma'}q_2'$ such that $(q_1',q_2')\in\Gamma, d_\Sigma(\sigma,\sigma')\le\varepsilon$.

A relation Γ is called a (ε,δ) approximate bisimulation relation when it is both a (ε,δ) approximate simulation relation of $\mathcal{T}_1$ by $\mathcal{T}_2$, and a (ε,δ) approximate simulation relation of $\mathcal{T}_2$ by $\mathcal{T}_1$.

Definition 5 (Approximate simulation): [20] $\mathcal{T}_2$ is a (ε,δ) approximate simulation of $\mathcal{T}_1$ (denoted $\mathcal{T}_1\preceq_{(\varepsilon,\delta)}\mathcal{T}_2$) if there exists Γ, a (ε,δ) approximate simulation relation of $\mathcal{T}_1$ by $\mathcal{T}_2$, such that for all $q_1\in Q_0^1$, there exists $q_2\in Q_0^2$ such that $(q_1,q_2)\in\Gamma$.

If $\mathcal{T}_1\preceq_{(\varepsilon,\delta)}\mathcal{T}_2$ and $\mathcal{T}_2\preceq_{(\varepsilon,\delta)}\mathcal{T}_1$, then we say that $\mathcal{T}_1$ and $\mathcal{T}_2$ are (ε,δ) approximately bisimilar, and write $\mathcal{T}_1\approx_{(\varepsilon,\delta)}\mathcal{T}_2$.

For the intent of this paper, we require from the trajectories of the abstraction the following properties: (1) exact replication of discrete states ($\delta=0$) and (2) approximate synchronization on switching times with precision ($\varepsilon>0$). For this reason, we will consider in the rest of the paper (ε,δ) simulation and bisimulation relations where $\delta=0$. We can define a simulation metric, as the tightest precision ε such that $\mathcal{T}_1\preceq_{(\varepsilon,0)}\mathcal{T}_2$:

$$d_{\overrightarrow{\mathcal{S}}}(\mathcal{T}_1,\mathcal{T}_2)=\inf\{\varepsilon:\mathcal{T}_1\preceq_{(\varepsilon,0)}\mathcal{T}_2\}$$

We can also define a bisimulation metric, as the tightest precision ε such that $\mathcal{T}_1\approx_{(\varepsilon,0)}\mathcal{T}_2$:

$$d_{\mathcal{B}}(\mathcal{T}_1,\mathcal{T}_2)=\inf\{\varepsilon:\mathcal{T}_1\approx_{(\varepsilon,0)}\mathcal{T}_2\}$$

The following diagram summarizes the classical relations between reachability, language and simulation metrics, for all $W\subset\Omega$:

$$\begin{array}{ccccc} d_{\mathcal{B}}(\mathcal{T}_1,\mathcal{T}_2) & \ge & d_{\mathcal{L}}(\mathcal{T}_1,\mathcal{T}_2) & \ge & d_{\mathcal{R}}(\mathcal{T}_1,\mathcal{T}_2,W)\\ \ge & & \ge & & \ge\\ d_{\overrightarrow{\mathcal{S}}}(\mathcal{T}_1,\mathcal{T}_2) & \ge & d_{\overrightarrow{\mathcal{L}}}(\mathcal{T}_1,\mathcal{T}_2) & \ge & d_{\overrightarrow{\mathcal{R}}}(\mathcal{T}_1,\mathcal{T}_2,W) \end{array}$$

The diagram above is very interesting: given $\mathcal{T}_1$ and an abstraction $\mathcal{T}_2$, the following hold:

$$\begin{aligned} \Lambda^{\mathcal{T}_1}(W) &\subseteq B\Big(cl(\Lambda^{\mathcal{T}_2}(W)),d_{\overrightarrow{\mathcal{R}}}(\mathcal{T}_1,\mathcal{T}_2)\Big)\\ &\subseteq B\Big(cl(\Lambda^{\mathcal{T}_2}(W)),d_{\overrightarrow{\mathcal{L}}}(\mathcal{T}_1,\mathcal{T}_2)\Big)\\ &\subseteq B\Big(cl(\Lambda^{\mathcal{T}_2}(W)),d_{\overrightarrow{\mathcal{S}}}(\mathcal{T}_1,\mathcal{T}_2)\Big), \end{aligned}$$

where $B(A,r)$ is the r neighborhood of the set A. The computation of $\Lambda^{\mathcal{T}_2}(W)$ for a simple system $\mathcal{T}_2$ can be used to characterize $\Lambda^{\mathcal{T}_1}(W)$ for a complex system $\mathcal{T}_1$. If the transition systems are deterministic, then the following classical result holds:

$$d_{\overrightarrow{\mathcal{S}}}(\mathcal{T}_1,\mathcal{T}_2)=d_{\overrightarrow{\mathcal{L}}}(\mathcal{T}_1,\mathcal{T}_2),\quad d_{\mathcal{B}}(\mathcal{T}_1,\mathcal{T}_2)=d_{\mathcal{L}}(\mathcal{T}_1,\mathcal{T}_2)$$

The aim of this paper is to construct a timed abstraction of a hybrid automaton $\mathcal{H}$ by means of a timed automaton. We show here that with the classical Euclidian metric $\bar{d}_\Sigma$ on positive reals it is not always possible to construct a timed automaton (and thus not even a durational graph, that is a subclass of timed automata) that is ε–*close* to $\mathcal{H}$, with ε finite: namely such that the discrete state trajectories of $\mathcal{G}$ approximate the discrete state trajectories of $\mathcal{H}$, with a finite desired precision on switching times according to the metric $\bar{d}_\Sigma$. The following example shows by a counterexample that given a hybrid automaton $\mathcal{H}$, there does not always exist a durational graph $\mathcal{G}$ and a finite precision $\varepsilon \geq 0$, such that $d_{\mathcal{L}}(\mathcal{H}, \mathcal{G}) \leq \varepsilon$.

Example 1: Consider the hybrid automaton $\mathcal{H}$ that consists of $Q = Q_0 = \{q\}, X = \mathbb{R}^2$, $X_0 = \{1\} \times [0,2]$, $\mathcal{E}_q : \dot{x_1} = x_1, \dot{x_2} = 0$, $E = \{e = (q,q)\}$, $Inv_q = \{(x_1,x_2) : x_2 > x_1 - 2\}$, $G_e = \{(x_1,x_2) : x_2 = x_1 - 2\}$, $R_e(x_1,x_2) = (1, x_2)$.

We first propose an abstraction $\mathcal{G}_1$ with only one discrete state, with a self loop and a guard $G_e = [t'_1, t'_2]$, and prove that $\forall \varepsilon \in \mathbb{R}_+ \cup \{0\}, d_{\mathcal{L}}(\mathcal{H}, \mathcal{G}_1) > \varepsilon$. Then, we prove that any timed automaton with a finite number of discrete states and clocks is affected by the same pathology of $\mathcal{G}_1$.

We will consider the languages of external trajectories $\mathcal{L}^{\mathcal{H}}$ and $\mathcal{L}^{\mathcal{G}_1}$ of transition systems that model a hybrid automaton $\mathcal{H}$ and a timed automaton $\mathcal{G}_1$. Since we are interested in reproducing the discrete state trajectories of $\mathcal{H}$, we consider as observation the current discrete state, i.e. $\Omega = Q$. The external language generated by $\mathcal{H}$ is given by

$$\mathcal{L}^{\mathcal{H}} = \{q \xrightarrow{t^*} q \xrightarrow{t^*} q \cdots : t^* \in [t_1, t_2]\},$$

with $t_1 = \ln 2, t_2 = \ln 4$. The external language generated by $\mathcal{G}_1$ is given by

$$\mathcal{L}^{\mathcal{G}_1} = \{q \xrightarrow{t^1} q \xrightarrow{t^2} q \cdots \xrightarrow{t^n} q \cdots : \forall i \in \mathbb{N}, t^i \in [t'_1, t'_2]\},$$

Consider the following external trajectory of $\mathcal{G}_1$:

$$\rho' = q \xrightarrow{t'_1} q \xrightarrow{t'_2} q \xrightarrow{t'_1} q \xrightarrow{t'_2} q \cdots$$

Consider the distance $\bar{d}_{\mathcal{X}}(\rho, \rho')$ between any execution $\rho \in \mathcal{L}^{\mathcal{H}}$ and $\rho' \in \mathcal{L}^{\mathcal{G}_1}$, according to the Euclidian metric on positive reals $\bar{d}_{\Sigma}(t_1, t_2) = |t_2 - t_1|$:

$$\bar{d}_{\mathcal{X}}(\rho, \rho') = \sup_{i \in \mathbb{N}} \sum_{j=0}^{\lfloor i/2 \rfloor} \Big(|t'_1 - t^*| + |t'_2 - t^*|\Big) + |t'_1 - t^*| \cdot i (\text{mod} 2)$$

Unless $t_1 = t_2 = t'_1 = t'_2 = t^*$ (that is not the case in our example), it is clear that $d_{\mathcal{X}}(\rho, \rho') = \infty$. Thus, for any $\varepsilon \geq 0$ the following holds:

$$\forall \varepsilon \in \mathbb{R}_+ \cup \{0\}, d_{\mathcal{B}}(\mathcal{H}, \mathcal{G}_1) \geq d_{\mathcal{L}}(\mathcal{H}, \mathcal{G}_1) > \varepsilon.$$

Increasing the number of discrete states of the timed automaton is useless: in fact, one can construct an abstraction $\mathcal{G}_K$ with a finite number K of discrete states with self cycles associated to non–singleton guards $G_{e_k} = [t_1^k, t_2^k]$, such that $\bigcup_{i=1}^{K} G_{e_k} = [t_1, t_2]$ (see Figure 1). In this case, we get the same problem of the case with just one discrete state. Otherwise, one can construct an abstraction $\mathcal{G}_{\infty}$ with an infinite number of discrete states with self cycles, and add cycles with singleton guards $G_{e_i} = \{t^k\}$ such that $\bigcup_{i=1}^{\infty} G_{e_i} = [t_1, t_2]$. In this case, the system has infinite discrete states and thus it is not a timed automaton.

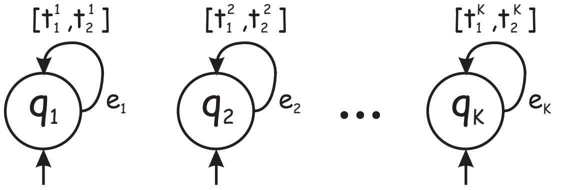

Fig. 1. Timed automaton $\mathcal{G}_K$.

Increasing the number of clocks and introducing clocks that do not reset is also useless, since it is not allowed in a timed automaton to define a guard as a function of clock variables, i.e. guards are rectangular sets. The same reasoning discussed above yields to the necessity of an infinite number of clocks or to non–rectangular guards, that is outside the expressive power of timed automata. Thus, the result follows.

Another important limitation of the Euclidian metric is that in practical applications it is often required to construct an approximation of a plant for a finite horizon analysis. In this case, it is necessary to construct an abstraction that tightly reproduces the plant behavior for the time interval of interest, and not for all time instants. For the motivations above, we define a *relative* distance d_{Σ} on $\Sigma = \mathbb{R}_+ \cup \{0\}$:

$$d_{\Sigma}(t_1, t_2) = \begin{cases} 0 & \text{if } t_1 = t_2 = 0 \\ \frac{|t_1 - t_2|}{t_1 + t_2} & \text{otherwise} \end{cases}$$

Proposition 2: $d_{\Sigma}(t_1, t_2)$ is a metric on Σ.

This metric is *relative* and not *absolute*, in the sense that it depends on the elapsed time: i.e. the distance between 1 and 2 seconds is considered the same as the distance between 100 and 200 seconds. Roughly speaking, it specifies the region of times where we are interested to define a tight approximation (short time horizon) and where we just need a more relaxing one (long time horizon).

IV. Abstraction algorithm

Given a hybrid automaton $\mathcal{H} = (Q \times X, Q_0 \times X_0, U, \mathcal{E}, E, Inv, G, R)$ and a desired precision ε, we propose a novel algorithm to construct a durational graph $\mathcal{G}$ such that $d_{\mathcal{B}}(\mathcal{H}, \mathcal{G}) \leq \varepsilon$. Define a relation $\gamma \subseteq Q \times \big(Q \times (E \cup Q_0)\big)$ as follows:

$$\gamma = \{(q, (q, l)), q \in Q : \text{either} \quad l \in inc(q) \quad \text{or} \quad l = q \in Q_0\}$$

Intuitively, γ relates each discrete state q to a pair (q, l) where l is either an incoming edge in q, or $l = q$ if q is an initial discrete state. In the following, we use the notation:

$$\Re_l \triangleq \begin{cases} X_0(q_0) & \text{if} \quad l = q_0 \in Q_0 \\ Im(R_e) & \text{if} \quad l = e \in E \end{cases}$$

where $X_0(q_0)$ is the set of initial continuous conditions associated to the initial discrete state q_0, and $Im(R_e)$ is the image of the reset R_e. Let $Reach_I(X_0)$ denote the set of states reachable from X_0 at time $t \in I$.

Algorithm 1: Let a hybrid automaton $\mathcal{H} = (Q \times X, Q_0 \times X_0, U, \mathcal{E}, E, Inv, G, R)$ and $\varepsilon \in \mathbb{R}_+$ be given. We define a durational graph $\mathcal{G} = (Q', Q'_0, E', Inv', G')$ as follows:

1) Define $Q' \triangleq \{(q, l) : (q, (q, l)) \in \gamma\}$ and $Q'_0 \triangleq \{(q, l) \in Q' : l = q \in Q_0\}$;

2) For each $\big((q_1,l_1),(q_2,l_2)\big) \in Q' \times Q'$ such that $l_2 = (q_1,q_2)$, define finite partitions $\{\Re_{l_1,i}\}_{i=1}^{m_1}, \{\Re_{l_2,j}\}_{j=1}^{m_2}$ of the sets $\Re_{l_1}, \Re_{l_2}$, such that for each element $\Re_{l_1,i}$ there exists $t_i \in \mathbb{R}_+$ that satisfies the following:

$$(i)\ \{t \in \mathbb{R}_+ \cup \{0\} : \exists w|_0^t, \exists x_0 \in \Re_{l_1,i}, x_{f_{q_1}}(t, w|_0^t, x_0) \in G_{l_2}\} \subseteq [t_i(1 - \frac{2\varepsilon}{1+\varepsilon}), t_i(1 + \frac{2\varepsilon}{1+\varepsilon})] \triangleq B(t_i,\varepsilon),$$

$$(ii)\ \exists! e \in out(q) : Reach_{B(t_i,\varepsilon)}(\Re_{l_1,i}) \cap G_e \neq \varnothing$$

and at least one of the following conditions holds:

$$(iii) \quad \forall x_0 \in \Re_{l_1,i}, Reach_{B(t_i,\varepsilon)}(\Re_{l_1,i}) \subseteq Reach_{B(t_i,\varepsilon)}(x_0)$$

$$(iv) \quad \exists \Re_{l_2,j} : R_{l_2}\Big(Reach_{B(t_i,\varepsilon)}(\Re_{l_1,i})\Big) \subseteq \Re_{l_2,j}$$

Split the discrete states $(q_1,l_1),(q_2,l_2)$ respectively in m_1, m_2 states $\{(q_1,l_1,i)\}_{i=1}^{m_1}, \{(q_2,l_2,j)\}_{j=1}^{m_2}$. If $R_{l_2}\Big(Reach_{B(t_i,\varepsilon)}(\Re_{l_1,i})\Big) \cap \Re_{l_2,j} \neq \varnothing$, then add $e' \triangleq \big((q_1,l_1,i),(q_2,l_2,j)\big)$ to E' and define $G'_{e'} \triangleq \{t_i\}$;

Algorithm 1 is guaranteed to converge if the discrete layer contains cycles, or if a disturbance is present. If the hybrid model contains cycles and the disturbance is not present, then the convergence is not guaranteed: in fact, if the limit trajectories of the system converge to a boundary of a partition, the algorithm might not reach a fixed point in finite time. However, the presence of a disturbance on the continuous dynamics guarantees convergence also in presence of cycles: in fact, the algorithm stops when each element of the partition $\{\Re_{l_1,i}\}_{i=1}^{m_1}$ either has only one outgoing edge, or it is so small that for any initial continuous condition the disturbance allows to trigger all outgoing edges. As the partition classes become smaller and smaller a fixed point is reached in finite time, since the reach set due to the disturbance includes the equivalence class. Notice that the presence of a disturbance on the continuous dynamics is a point of strength and not a limitation, since it provides robustness to the analysis of the system behavior. In addition to this, notice that many applications can be described by hybrid automata that do not contain cycles, see e.g. hybrid models proposed in [8], [9] for air traffic management procedures.

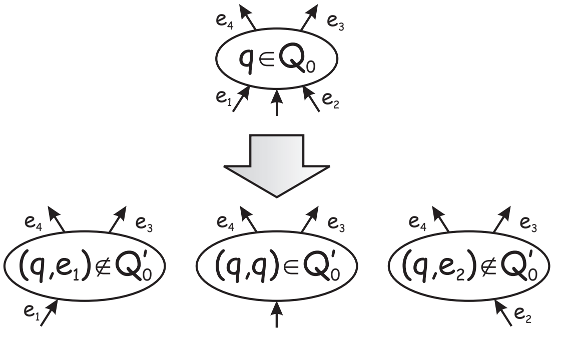

Fig. 2. Split induced by the relation γ.

The intuition behind the algorithm is the following: we first split each discrete state depending on the number of incoming edges and initial conditions (Figure 2). Notice that the first split ensures that any discrete state has only one incoming edge. Then, a further split is performed according to a partition of the image of the reset associated to the incoming edge (Figure 3). Because of condition (i), the following holds:

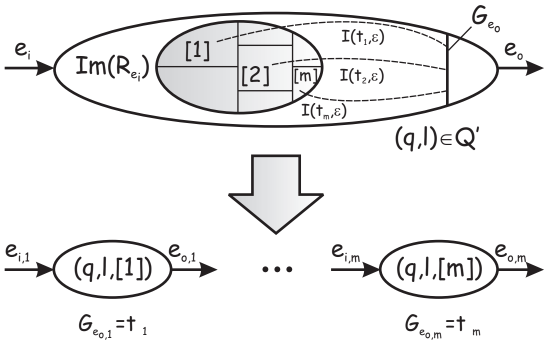

Fig. 3. Split induced by the arrival time to the guards.

Proposition 3: For all $e' = \big((q_1,l_1,i),(q_2,l_2,j)\big) \in E'$ and for any equivalence class $\Re_{l_1,i}$:

$$\forall x_0 \in \Re_{l_1,i}, \forall w|_0^{t^*}, d_\Sigma(t^*, t_i) \leq \varepsilon$$

where $w|_0^{t^*}$ is a disturbance signal and t^* is the arrival time to the guard G_{l_2}, namely such that $x_{f_{q_1}}(t^*, w|_0^t, x_0) \in G_{l_2}$, and $\forall t \in [0,t^*), x_{f_{q_1}}(t, w|_0^t, x_0) \in Inv_{q_1}$.

Roughly speaking given a partition $\{\Re_{l_1,i}\}_{i=1}^{m_1}$, the arrival time to a guard set starting from all continuous states in an equivalence class $\Re_{l_1,j}$ belongs to the set $B(t_j,\varepsilon)$. The partition obtained by Algorithm 1 is not uniform, and is *smart* in the sense that it depends on the continuous dynamics. Because of conditions (iii) and (iv), the following holds:

Proposition 4: For each discrete state (q_1,l_1,i) generated by the partition $\{\Re_{l_1,i}\}_{i=1}^{m_1}$, let $out\big((q_1,l_1,i)\big) = \{e_1,\cdots,e_k\}$. Then either $k=1$, or for all $j \in \{1,\cdots,k\}$ the following holds:

$$\forall x_0 \in \Re_{l_1,i}, \exists t^*, \exists w|_0^{t^*} : x_{f_{q_1}}(t^*, w|_0^{t^*}, x_0) \in G_{e_j}$$

Roughly speaking each discrete state (q_1,l_1,i) generated by the second split either has only one outgoing edge, or every continuous trajectory starting from $\Re_{l_1,i}$ can trigger all the outgoing edges.

Theorem 1: Given a hybrid automaton $\mathcal{H}$, a required precision $\varepsilon \in \mathbb{R}_+$, and $\mathcal{G}$ constructed by Algorithm 1, then $d_\mathcal{B}(\mathcal{T},\mathcal{H}) \leq \varepsilon$.

It is clear that $d_\mathcal{L}(\mathcal{H},\mathcal{G}) \leq d_\mathcal{B}(\mathcal{H},\mathcal{G}) \leq \varepsilon$, and that if $\mathcal{H}$ is deterministic, then $d_\mathcal{B}(\mathcal{H},\mathcal{G}) = d_\mathcal{L}(\mathcal{H},\mathcal{G})$. We conclude this section by applying our algorithm to the system $\mathcal{H}$ defined in Example 1.

Example 2: Consider $\mathcal{H}$ as defined in Example 1 and a desired precision $\varepsilon = 0.02$, we can use Algorithm 1 to construct a durational graph $\mathcal{G}$. The discrete layers of $\mathcal{H}$ and $\mathcal{G}$ are depicted in Figure 4. Following Algorithm

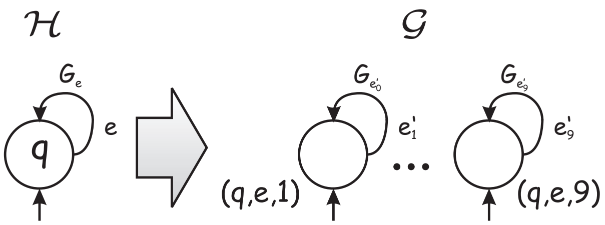

Fig. 4. System $\mathcal{H}$ of Example 2 and abstraction $\mathcal{G}$.

1, we construct a partition of the set $\Re_e = \{1\} \times [0, 2]$. More precisely, we partition the rectangular interval $[0, 2]$ as follows:

$$\{[0, 0.16], [0.16, 0.33], [0.33, 0.52], [0.52, 0.72], [0.72, 0.94], [0.94, 1.17], [1.17, 1.42], [1.42, 1.71], [1.71, 2]\}.$$

The partition contains 9 equivalence classes and is not uniform, i.e. it is finer when close to 0 and coarser when close to 2. The guard of each edge $e'_i, i \in \{0, \cdots, 9\}$ is given by the average arrival time of each interval, e.g., $G_{e'_2} = t_2 = \frac{\ln(0.16+2)+\ln(0.33+2)}{2} = 0.807s$. Theorem 1 implies that $d_{\mathcal{B}}(\mathcal{T}, \mathcal{H}) \leq 0.02$.

V. Conclusions

We proposed a novel algorithm to construct a durational graph that is an approximate timed abstraction of a hybrid automaton. To define the precision of the approximation, we motivated and introduced a *relative* metric on reachability time, external language and simulation relation. The discrete state trajectories of our abstraction approximate the discrete state trajectories of the hybrid automaton, with a desired precision on switching times. We guaranteed the convergence of our algorithm for a general class of hybrid automata with disturbance in the continuous dynamics. Our abstraction can be useful for automatic verification of properties of hybrid automata. Current work aims to automatize the algorithm, using the computational framework developed in [19]. As a future extension, we aim to establish a relation between the number of discrete states of the abstraction and the number of discrete states and continuous dynamics of the hybrid automaton.

References

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

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

[3] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. *Proccedings of the IEEE*, 88(2):971–984, July 2000.

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

[5] M. Babaali and G. J. Pappas. Observability of switched linear systems in continuous time. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *Lecture Notes in Computer Science*, pages 103–117. Springer Verlag, 2005.

[6] Paul Caspi and Albert Benveniste. Toward an approximation theory for computerised control. In *Proceedings of the 2^{nd} International Workshop on Embedded Software, EMSOFT2002, Grenoble*, page Volume 2491 in Lecture Notes in Computer Science, October 2002.

[7] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. The MIT Press, Cambridge, Massachusetts, 2002.

[8] E. De Santis, M. D. Di Benedetto, S. Di Gennaro, A. D'Innocenzo, and G. Pola. Critical observability of a class of hybrid systems and application to air traffic management. *Book Chapter of Lecture Notes on Control and Information Sciences, Springer Verlag*, 2005.

[9] M. D. Di Benedetto, S. Di Gennaro, and A. D'Innocenzo. Critical states detection with bounded probability of false alarm and application to air traffic management. In *Proceedings of the 2^{nd} IFAC Conference on Analysis and Design of Hybrid Systems (ADHS), Alghero, Sardinia, Italy*, June 7-9 2006.

[10] A. D'Innocenzo, M. D. Di Benedetto, and S. Di Gennaro. Observability of hybrid automata by abstraction. In J. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control*, volume 3927 of *Lecture Notes in Computer Science*, pages 169–183. Springer Verlag, 2006.

[11] A. D'Innocenzo, A.A. Julius, M.D. Di Benedetto, and G.J. Pappas. Approximate timed abstractions of hybrid automata. Technical report, University of L'Aquila, 2007. www.diel.univaq.it/tr/web/web_search_tr.php.

[12] D. Forstner, M. Jung, and J. Lunze. A discrete-event model of asynchronous quantised systems. *Automatica*, 38(8):1277–1286(10), August 2002.

[13] A. Girard. Reachability of uncertain linear systems using zonotopes. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *Lecture Notes in Computer Science*, pages 291–305. Springer Verlag, 2005.

[14] Antoine Girard and George J. Pappas. Approximate bisimulation relations for constrained linear systems. *Automatica*, Accepted for publication.

[15] Antoine Girard and George J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, Accepted for publication.

[16] Zhi Han and B. H. Krogh. Reachability analysis of large–scale affine systems using low–dimensional polytopes. In J. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control*, volume 3927 of *Lecture Notes in Computer Science*, pages 287–301. Springer Verlag, 2006.

[17] T. A. Henzinger, R. Majumdar, and V. Prabhu. Quantifying similarities between timed systems. In *Proceedings of the Third International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, volume 3829 of *Lecture Notes in Computer Science*, pages 226–241. Springer, 2005.

[18] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. Hytech: A model checker for hybrid systems. *Software Tools for Technology Transfer*, 1:110–122, 1997.

[19] A.A. Julius, G. Fainekos, M. Anand, I. Lee, and G.J. Pappas. Robust test generation and coverage for hybrid systems. In *Hybrid Systems: Computation and Control, To appear*, Lecture Notes in Computer Science. Springer Verlag, 2007.

[20] A.A. Julius and G.J. Pappas. Approximate equivalence and approximate synchronization of metric transition systems. In *Proceedings of the 45^{th} IEEE Conference on Decision and Control, San Diego, CA, USA*, December 2006.

[21] A.A. Julius and G.J. Pappas. Approximate abstraction of stochastic hybrid systems. *Accepted for publication to the IEEE Trans. Automatic Control*, 2007.

[22] F. Laroussinie, N. Markey, and P. Schnoebelen. Efficient timed model checking for discrete-time systems. *Theoretical Computer Science*, 353(1-3):249–271, march 2006.

[23] K. G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer*, 1(1):134–152, December 1997.

[24] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specications for hybrid systems. *Automatica, Special Issue on Hybrid Systems*, 35, 1999.

[25] G.J. Pappas. Bisimilar linear systems. *Automatica*, 39(12):2035–2047, December 2003.

[26] O. Stursberg and S. Kowalewsky. Approximating switched continuous systems by rectangular automata. In *Proceedings of the 1999 European Control Conference, Karlsruhe, Germany*, 1999.

[27] S. Yovine. Kronos: A verification tool for real-time systems. *International Journal of Software Tools for Technology Transfer, Springer-Verlag*, 1(1):123–133, October 1997.