Towards the Complexity of Controls for Timed Automata with a Small Number of Clocks *

Taolue Chen
CWI
PO Box 94079, 1090 GB Amsterdam, NL
chen@cwi.nl

Jian Lu
Nanjing Uinversity
Nanjing, Jiangsu, 210093, CN
lj@nju.edu.cn

Abstract

Control of timed systems has become a very active research area. In this paper, we revisit the complexity of safety and reachability control problems for timed automata. Generally, these problems turn out to be EXPTIME*-complete and we aim at finding tractable subclasses which admit efficient control. To this purpose, we consider the control for timed automata with a small number of clocks. We first show that for three clock TAs, the complexity has climbed up to* EXPTIME*-complete; We then propose efficient algorithms to control one-clock TAs and show that in this setting, the control problems become P-complete, both in the known and unknown switch condition assumptions; In the end, we prove the* PSPACE*-hardness of control for two-clock TAs.*

1 Introduction

During the last decade, *real-time systems*, where quantitative information on time is required, have received immense attentions. Alur and Dill's *Timed Automata* ([1], TAs for short), i.e., automata extended with *clocks* that progress synchronously with time, have become a well-established and widely used model for real-time systems. Since the mid-90's, works on the control of continuous (e.g. real-time) systems are flourishing, partially promoted by the progress in identifying some tractable ways of dealing with systems modelled as timed automata and hybrid automata. Indeed, it is natural to extend Ramadge and Wonham's framework [16] to real time settings, say, to use timed automata to describe plants and to solve the problem of controller synthesis for them. Indeed, there are now extensive works addressing this problem (see [5] for an excellent survey). In particular, [15] (see also [2] and [3]) investigates this problem when the plant is modelled as a timed automaton and the specification is given as an internal winning condition on the state space of the plant: A controller looks at the values of the plant's clocks and prescribes a set of moves the system should take.

In this paper, we focus on identifying the complexity of control for timed automata[1]. Partially inspired by the work of [14], we revisit the control problems and aim at clarifying the complexity of control for timed automata with *a small number of clocks*. For this purpose, we focus on safety or reachability control problem and investigate the complexity for timed automata with one, two, three (or more) clocks. We first show that for TAs with three clocks, the complexity has climbed up to EXPTIME-complete; We then show that the control becomes P-complete for one clock TAs. Note we work on both the known switch condition and unknown switch condition cases (the terms are coined in [8]) where unknown switch condition control is the one adopted in [15]. At last we show the PSPACE-hardness for control of TAs with two clocks. Our results, to some extent, complement the results in [12] and suggest that two sources of the exponential complexity, namely, many clocks and large constants c_y, are both inherent (provided that the TAs have at least three clocks).

Due to space restriction, all proofs in this extended abstract are omitted. And we only present the results based on *unknown switch condition assumption*. The missing parts can be found in the extended version [9].

2 Preliminaries

In the rest of the paper, we use $\mathbb{N}$, $\mathbb{Q}_{\geq 0}$ and $\mathbb{R}_{\geq 0}$ to denote the sets of natural, nonnegative rational and nonnegative real numbers, respectively. Let Σ be a finite alphabet ranged over by σ.

*This work is partially supported by the Dutch Bsik project BRICKS, the Chinese national 863 program (2007AA01Z178), NSFC (60736015) and JSNSF (BK2006712).

[1] A more precise formulation might be: the control of timed plants which are modelled as timed automata. However, we prefer this more natural formulation.

978-0-7695-3305-6/08 $25.00 © 2008 IEEE
DOI 10.1109/FSKD.2008.410

Clocks, operations on clocks. We consider a finite set $\mathcal{C}$ of real valued *clocks*. A *clock valuation* over $\mathcal{C}$ is a function $v : \mathcal{C} \to \mathbb{R}_{\geq 0}$ which assigns to each clock a time value in $\mathbb{R}_{\geq 0}$. We use $\mathcal{B}(\mathcal{C})$ to denote the set of boolean expressions over atomic formulae [2] of the form $x \sim k$ with $x \in \mathcal{C}$, $k \in \mathbb{N}$, and $\sim \in \{\leq, <, =, \geq, >\}$. The elements of $\mathcal{B}(\mathcal{C})$ are called *clock constraints*, which are interpreted over *valuations* for $\mathcal{C}$. For clock valuation v and time constraint g, we write $v \models g$ meaning v satisfies g. We denote the set of valuations over $\mathcal{C}$ by $\mathbb{R}^{\mathcal{C}}$. For every $v \in \mathbb{R}^{\mathcal{C}}$, the valuation $v + t$ is defined as $(v + t)(x) = v(x) + t$ for any $x \in \mathcal{C}$. For $r \subseteq \mathcal{C}$, we write $v[r := 0]$ for the valuation which maps each clock in r to value 0 and agrees with v over $\mathcal{C} \setminus r$.

Definition 1 [Timed automata] A timed automaton is a 6-tuple $\mathcal{A} = (Q, \Sigma, \mathcal{C}, q_0, \to, \mathcal{I})$ where

- Q is a finite set of (control) *states*, and $q_0 \in Q$ is the initial state;
- Σ is a finite alphabet of *actions*;
- $\mathcal{C}$ is a set of *clocks*;
- $\to \subseteq Q \times \mathcal{B}(\mathcal{C}) \times \Sigma \times \wp(\mathcal{C}) \times Q$ is a finite set of action *transitions*: for $(q, g, \sigma, r, q') \in \to$, g is the enabling condition (a.k.a. guard) of the transition, σ is the action labeling the transition and $r \subseteq \mathcal{C}$ is the set of clocks to be reset with the transition (we usually write $q \xrightarrow{g,\sigma,r} q'$ instead of $(q, g, \sigma, r, q') \in \to$);
- $\mathcal{I} : Q \to \mathcal{B}(\mathcal{C})$ assigns a constraint, called an *invariant*, to any control state.

A *configuration* of a timed automaton $\mathcal{A}$ is a pair (q, v), where $q \in Q$ is the current (control) state and $v \in \mathbb{R}^{\mathcal{C}}$ is the current clock valuation over $\mathcal{C}$. The initial configuration of $\mathcal{A}$ is (q_0, v_0) where v_0 is the valuation mapping all clocks in $\mathcal{C}$ to 0.

The semantics of timed automaton $\mathcal{A}$ is often defined in terms of a *timed transition system* (TTS for short). In general, there are two kinds of transitions. From configuration (q, v), it is possible to perform the *action transition* (a.k.a. discrete transition) $q \xrightarrow{g,\sigma,r} q'$ if $v \models g$ and $v[r := 0] \models \mathcal{I}(q')$ and then the new configuration is $(q', v[r := 0])$. It is also possible to let time elapse, and reach $(q, v + t)$ for some $t \in \mathbb{R}_{\geq 0}$ whenever the invariant $\mathcal{I}(q)$ is not violated. In this case, a *timed transition* (a.k.a. continuous transition) is performed. Formally, we define a timed transition system $\mathcal{T}_A = (S, s_0, \to)$ [3], where

- $S = \{(q, v) \mid q \in Q \text{ and } v \in \mathbb{R}^{\mathcal{C}} \text{ s.t. } v \models \mathcal{I}(q)\}$ and $s_0 = (q_0, v_0)$;
- $\to \subseteq S \times (\Sigma \cup \{\delta(t) \mid t \in \mathbb{R}_{\geq 0}\}) \times S$ and we have $(q, v) \xrightarrow{e} (q', v')$ where $e \in \Sigma \cup \{\delta(t) \mid t \in \mathbb{R}_{\geq 0}\}$ iff
  - either $q' = q$, $v' = v + t$ and $v + t' \models \mathcal{I}(q)$ for any $t' \leq t$. In this case, we write $(q, v) \xrightarrow{\delta(t)} (q, v + t)$;
  - or there exists $q \xrightarrow{g,\sigma,r} q'$ and $v \models g$, $v' = v[r := 0]$ and $v' \models \mathcal{I}(q')$. In this case, we write $(q, v) \xrightarrow{\sigma} (q, v + t)$.

A run (execution) of $\mathcal{A}$ is a finite or infinite path in $\mathcal{T}_A$. Let $s = (q, v)$ be a configuration. A run ρ from s can be described as a finite or infinite sequence of transitions $\rho = s_0 \xrightarrow{\delta(t_0)} \xrightarrow{\sigma_0} s_1 \xrightarrow{\delta(t_1)} \xrightarrow{\sigma_1} s_2 \cdots$. We write $\mathsf{Runs}(\mathcal{A})$ (resp. $\mathsf{Runs}_f(\mathcal{A})$) the set of runs (resp. finite runs) in $\mathcal{A}$. We say that $\sigma \in \Sigma$ is *enabled* in (q, v) if there exists (q', v') such that $(q, v) \xrightarrow{\sigma} (q', v')$. We say that $\mathtt{t}$ (a special symbol and we assume that $\mathtt{t} \notin \Sigma$) is *enabled* in (q, v) if there exists $t \geq 0$ and (q', v') such that $(q, v) \xrightarrow{\delta(t)} (q', v')$. We write $en(q, v)$ the subset of $\Sigma \cup \{\mathtt{t}\}$ of actions or $\mathtt{t}$ enabled in (q, v).

The *size* of a TA is $|Q| + |\mathcal{C}| + \sum_{(q,g,r,q') \in \to} |g| + \sum_{q \in Q} |\mathcal{I}(q)|$, where the size of a constraint is its length (constants are encoded in binary). We use $|\to|$ to denote the number of transitions in $\mathcal{A}$.

We define the control problem for a timed system given as a timed automaton. A *timed plant* is a timed automaton $\mathcal{A} = (Q, \Sigma, \mathcal{C}, q_0, \to, \mathcal{I})$ [4] where the alphabet Σ is partitioned into two subsets Σ_c and Σ_u corresponding respectively to *controllable* and *uncontrollable* actions. Intuitively, the controller will be able to perform controllable actions, whereas the environment will be able to perform uncontrollable actions. Note that in the literature, several formulations of the control problem have been proposed [5], some of them are based on a two-player game formulation where the "controller" plays against the "environment". In this paper, we focus on "control games", an asymmetric formulation where the controller has to fix his strategy, and this strategy has to be winning whatever the environment does.

Definition 2 [Control map] Let $\mathcal{T}_{\mathcal{A}}$ be a TTS associated with $\mathcal{A}$. A *simple real-time control map* for $\mathcal{A}$ is a function $\kappa : Q \times \mathbb{R}^{\mathcal{C}} \mapsto \wp(\Sigma_c \cup \{\mathtt{t}\})$ that maps every configuration (q, v) to a set $\kappa(p, v) \subseteq en(q, v)$ of enabled actions. The control map κ is *deadlock-free* if $\kappa(p, v) \neq \emptyset$ for any configuration (q, v).

According to this function, the *controller* chooses at any configuration (q, v) whether to issue some enabled (controllable) transition $\sigma \in \Sigma_c$ (if $\kappa(q, v) \subseteq \Sigma_c$) or do nothing and let time go by (if $\mathtt{t} \in \kappa(q, v)$). Here, note that a control map is *not* deterministic as it may propose a *set* of actions in $\Sigma_c \cup \{\mathtt{t}\}$.

[2] Considering diagonal constraints $x - y \sim k$ does not matter for the complexity.

[3] Here, we abuse the notation $\to$ a bit. However, the meaning can be recovered from the context.

[4] To save notations, we overload $\mathcal{A}$ to denote both timed automata and timed plants. However, the exact meaning is clear from the context.

Provided a control map κ, we can define the *closed-loop system* (a.k.a. controlled system) as follows:

Definition 3 [Closed-loop system] Assume $\mathcal{T}_{\mathcal{A}} = (S, s_0, \rightarrow)$ the TTS associated with $\mathcal{A}$ and a simple real-time control map κ for $\mathcal{T}_{\mathcal{A}}$. The *closed-loop system* $\kappa(\mathcal{T}_{\mathcal{A}})$ is the TTS $\mathcal{T}_{\mathcal{A}}^{\kappa} = (S, s_0, \rightarrow_{\kappa})$, where $(q, v) \xrightarrow{e}_{\kappa} (q', v')$ iff

- either $e \in \Sigma_u$ and $(q, v) \xrightarrow{e} (q', v')$;
- or $e \in \kappa(q, v) \cap \Sigma_c$ and $(q, v) \xrightarrow{e} (q', v')$;
- or $e = \delta(t)$ for some $t \in \mathbb{R}_{\geq 0}$ s.t. $(q, v) \xrightarrow{\delta(t)} (q, v + t)$ and for any $t' \in [0, t)$, $\kappa(q, v + t') = \mathsf{t}$.

Remark 1 In this paper, we only consider *simple* real-time control maps. In terms of game theory, these amount to memoryless strategies. Since we shall only cope with safety and reachability control problems, this class of controllers suffices. Of course, one can consider more involved strategies (e.g. history dependent or even random ones). However, this is orthogonal to the works presented here.

Given a timed plant $\mathcal{A} = (Q, \Sigma, \mathcal{C}, q_0, \rightarrow, \mathcal{I})$, a specification φ for $\mathcal{A}$ is a subset of $\mathsf{Runs}(\mathcal{A})$. Intuitively it corresponds to the *desired behaviors* of the plant. In the following, we will consider special cases of specifications such as safety objectives and reachability objectives. Intuitively, for a reachability specification, the goal for the controller is to reach a set of good states, whereas for safety specification, the controller has to try to avoid a set of undesired states. Formally, let $\mathcal{A}$ be a timed plant, and let $\mathbf{Good}, \mathbf{Bad} \subseteq Q$, we define the two specifications as follows:

$$\varphi_{\mathbf{Good}} = \{\rho = (q_0, v_0) \xrightarrow{\sigma_0, t_0} (q_1, v_1) \xrightarrow{\sigma_1, t_1} \cdots \mid \exists i. q_i \in \mathbf{Good}\}$$
$$\varphi_{\mathbf{Bad}} = \{\rho = (q_0, v_0) \xrightarrow{\sigma_0, t_0} (q_1, v_1) \xrightarrow{\sigma_1, t_1} \cdots \mid \forall i. q_i \notin \mathbf{Bad}\}$$

And the safety (resp. Reachability) control problems can be read as: Given a timed plant $\mathcal{A}$, a set of states $\mathbf{Bad}$ (resp. $\mathbf{Good}$) and an initial configuration (q_0, v_0), determine whether there is a control map κ such that all the runs from (q_0, v_0) of the closed-loop system $\kappa(\mathcal{T}_{\mathcal{A}})$ belong to $\varphi_{\mathbf{Bad}}$ (resp. $\varphi_{\mathbf{Good}}$). We note that these two problems are dual. Thus, in each case we can take the liberty to consider only one of them, and the obtained result can be easily translated into the other one modulo duality. Throughout the paper, the word control is used to denote both cases, if we do not specify the type explicitly.

3 Control timed automata with three clocks

In this section, we settle the complexity of control for timed automata with three clocks (3C-TAs for short). We devote ourselves to EXPTIME-hardness proof, for which we reduce the halting problem for *alternating Turing machines* (ATMs for short) using polynomial space, which is EXPTIME-hard (since EXPTIME= APSPACE) to the reachability control problems for 3C-TAs. The reduction is adapted from [10]. We stress that w.l.o.g., here we consider a model of alternation with a binary branching degree and assume that the tape alphabet of ATM is $\{0, 1\}$. We obtain the following result:

Theorem 1 The control problems for timed automata with three clocks are EXPTIME-complete.

4 Control timed automata with one clock

In this section, we turn to the control of timed automata with one clock, which we denote as 1C-TAs. We will assume that in 1C-TAs, the guards are given by two constants defining the minimal (resp. maximal) value for the only clock to perform the transition. This does *not* lose any generality in that it is always possible to reduce, in polynomial time, any 1C-TAs to an equivalent automaton admitting such a property.

4.1 Zone constructions

Assume a 1C-TA $\mathcal{A} = (Q, \Sigma, \mathcal{C}, q_0, \rightarrow, \mathcal{I})$. We denote by $\mathsf{Cst}_{\mathcal{A}} \subseteq \mathbb{N} \cup \{\infty\}$ the set of all constants occurring in $\mathcal{A}$ (either in guards or in invariants) plus 0. We use $b_0, b_1, \ldots, b_k$ to range over $\mathsf{Cst}_{\mathcal{A}}$ and assume that $0 = b_0 < b_1 < \cdots < b_k$ and $|\mathsf{Cst}_A| = k + 1$. The set Cst_A defines a set $\mathcal{I}_{\mathsf{Cst}_{\mathbb{A}}}$ of $2(k+1)$ intervals $\lambda_0, \lambda_1, \ldots$ with $\lambda_0 = [b_0, b_0]$, $\lambda_1 = (b_0, b_1)$, $\lambda_2 = [b_1, b_1], \ldots, \lambda_{2k+1} = (b_k, \infty)$. We say an interval λ_j is a *boundary zone*, if j is an even number; Otherwise, it is a *non-boundary zone*. Given any $x \in \mathbb{R}_{\geq 0}$, we define an *index mapping* $\ulcorner \cdot \urcorner : \mathbb{R}_{\geq 0} \rightarrow \{0, \ldots, 2(k+1)\}$ as $x \in \lambda_{\ulcorner x \urcorner}$. We say $\ulcorner x \urcorner$ is the *index* of x. Given any interval λ, we write $lb(\lambda)$ (resp. $ub(\lambda)$) as the left endpoint (resp. right endpoint) of λ. It is essential to note that $k \leq 2 \cdot | \rightarrow | + |Q|$, namely, the cardinality of $\mathcal{I}_{\mathsf{Cst}_{\mathbb{A}}}$ is bounded by $\mathcal{O}(| \rightarrow | + |Q|)$.

The following lemma, albeit simple, demonstrates some nice properties on the zone construction.

Lemma 1 Assume 1C-TA $\mathcal{A}$, the following properties hold:

1. for any transition $p \xrightarrow{g, \sigma, r} p'$, $x \models g$[5] implies that for any $y \in \lambda_{\ulcorner x \urcorner}$, $y \models g$.
2. for any $q \in Q$, $x \models \mathcal{I}(q)$ implies that for any $y \in \lambda_{\ulcorner x \urcorner}$, $y \models \mathcal{I}(q)$.

[5] Here we abuse the notation $\models$ a little: x denotes the value of the clock. Since there is only one clock, it is not harmful.

Remark 2 We note that this result can *not* be directly derived from the standard results for general TAs, since there the region construction [1] is applied which is much finer than the construction here. We note that the one-clock assumption is essential for the correctness of Lemma 1.

In the research of timed automata, the *symbolic* approach usually means exploring the infinite timed transition system directly by manipulating constraints that may represent infinite state sets (e.g. the clock zone method). [15] applies this approach in order to avoid unnecessary blow-up of explicit region construction. However, since they considered the timed plant with any number of clocks, as Section 3 indicates, in general (i.e. in the worst case and for timed plants with at least three clocks), the region construction is unavoidable, and in theoretical complexity it is even optimal. In other words, if we are not lucky enough, this zone construction which is much coarser, will unfortunately be refined step by step and in the end turns into the region construction. However, the case is completely different if 1C-TAs are considered. In contrast, we show that even in the worst case, the region construction, which leads to the exponential blow-up is unnecessary. Thanks to the simplicity of 1C-TA (in particular, the elimination of interference between different clocks), we propose a *zone* construction, which provides a suitable *granularity* for the state space partition in the sense that, on the one hand, it is fine enough for the closure property of control operation (e.g. see π below); on the other hand, it provides a much coarser partition than the one corresponding to the region graph, and thus leads to a polynomial algorithm.

To proceed, we first explain how to solve reachability and safety control problems in the timed framework ([15], the presentation is close to [5]). Let $\mathcal{A} = (Q, \Sigma, \mathcal{C}, q_0, \rightarrow, \mathcal{I})$ be a timed plant, we define the controllable and uncontrollable discrete predecessors of a set of configurations $D \subseteq Q \times \mathbb{R}_{\geq 0}$ as follows:[6]

- controllable discrete predecessors:

$$\mathsf{cPred}(D) = \left\{ (q,x) \in Q \times \mathbb{R}_{\geq 0} \;\middle|\; \begin{array}{l} \exists c \in \Sigma_c,\, c \in en(q,x) \text{ and } \forall.(q',x') \in Q \times \mathbb{R}_{\geq 0}, \\ (q,x) \xrightarrow{c} (q',x') \implies (q',x') \in D \end{array} \right\}$$

- uncontrollable discrete predecessors:

$$\mathsf{uPred}(D) = \{(q,x) \in Q \times \mathbb{R}_{\geq 0} \mid \exists u \in \Sigma_u \; u \in en(q,x) \text{ s.t. } (q,x) \xrightarrow{u} (q',x') \text{ and } (q',x') \in D \;\}$$

We explain the intuition as follows. The set $\mathsf{cPred}(D)$ is the set of configurations from which we can enforce a configuration of D by doing a controllable action. The set $\mathsf{uPred}(D)$ is the set of configurations from which the environment can do an uncontrollable action which leads to a configuration in D. In the timed framework, these two *discrete* controllable and uncontrollable predecessors are *not* sufficient, and we need to define a *time controllable predecessor operator* of a set D of configurations: a configuration (q, v) is in $\pi(D)$ iff

(1) it is possible to let t time units elapse for some $t \geq 0$ and use a controllable action to reach D and no uncontrollable action played before or at t leads outside D; or

(2) D can be reached by just letting time elapse and no uncontrollable action leads outside D.

Formally the operator π is defined as follows:

$$\pi(D) = \left\{ (q,x) \in Q \times \mathbb{R}_{\geq 0} \;\middle|\; \begin{array}{l} \exists t.(q,x) \xrightarrow{\delta(t)} (q',x'), (q',x') \in \mathsf{cPred}(D) \\ \text{and } \mathsf{Post}_{[0,t]}(q,x) \cap \mathsf{uPred}(\bar{D}) = \emptyset; \text{ or} \\ \exists t.\mathsf{Post}_{[t,+\infty]}(q,x) \subseteq D \text{ and} \\ \mathsf{Post}_{[0,+\infty]}(q,x) \cap \mathsf{uPred}(\bar{D}) = \emptyset \end{array} \right\}$$

where $\mathsf{Post}_I(q,x) = \{(q, x+t) \mid t \in I \text{ and } x+t \models \mathcal{I}(q)\}$.

Assuming a 1C-TA $\mathcal{A}$ and $Q = \{q_0, \ldots, q_m\}$. Clearly, any set of configurations can be written as $\mathcal{K} = \bigcup_{0 \leq i \leq m} \{q_i\} \times P_i$, where $P_i \subseteq \mathbb{R}_{\geq 0}$. Thus, we can view π as a transformation on $\mathcal{K}$. Let $\mathcal{Z} = \{(q,\lambda) \mid q \in Q \text{ and } \lambda \in \mathcal{I}_{\mathsf{Cst}_{\mathbb{A}}}\}$. $\mathcal{K}$ is called *well formed 1C-zone* if for any $0 \leq i \leq m$, there exists some index set $\mathcal{J}_i$ such that $P_i = \bigcup_{j \in \mathcal{J}_i} \{\lambda_j \mid (q_i, \lambda_j) \in \mathcal{Z}\}$.

Clearly, well formed 1C-zones are closed under union, intersection and complementation (w.r.t. $\mathcal{Z}$). We then attempt to prove the closure of well formed 1C-zone under the transformation of π. At first, we introduce a technical lemma.

Lemma 2 Assume $D \subseteq \mathcal{Z}$ and $q \in Q$. Then the following properties hold:

1. For any $x \in \mathbb{R}_{\geq 0}$, $(q,x) \in \mathsf{cPred}(D)$ implies that for any (q,y) such that $y \in \lambda_{\ulcorner x \urcorner}$, $(q,y) \in \mathsf{cPred}(D)$;
2. For any $x \in \mathbb{R}_{\geq 0}$, $(q,x) \in \mathsf{uPred}(D)$ implies that for any (q,y) such that $y \in \lambda_{\ulcorner x \urcorner}$, $(q,y) \in \mathsf{uPred}(D)$.

Theorem 2 Assume $\mathcal{K} = \bigcup_{0 \leq i \leq m} \{q_i\} \times P_i$. If $\mathcal{K}$ is a well formed 1C-zone, so is $\pi(\mathcal{K}) = \bigcup_{0 \leq i \leq m} \{q_i\} \times P'_i$.

We now construct an increasing and a decreasing version of π to solve reachability and safety control: $\pi_{\text{reach}}(D) = D \cup \pi(D)$ and $\pi_{\text{safe}}(D) = D \cap \pi(D)$. Moreover, note that $\pi^*_{\text{reach}}(\mathbf{Good}) = \lim_{i \to \infty} \pi^i_{\text{reach}}(\mathbf{Good})$ and

[6] Here, we instantiate the general construction in the setting of 1C-TA.

$\pi^*_{\text{safe}}(\mathbf{Bad}) = \lim_{i\to\infty} \pi^i_{\text{safe}}(\mathbf{Bad})$. To implement this, set $D_0 = \{(q, \mathbb{R}_{\geq 0}) \mid q \in \mathbf{Good}\}$. Clearly, $D_0 \subseteq \mathcal{Z}$. By Theorem 2 and induction, it is easy to see that $\pi^*_{\text{reach}}(\mathbf{Good}) = \bigcup_{0\leq i\leq |\mathcal{Z}|} \pi^i_{\text{reach}}(D_0) \subseteq \mathcal{Z}$ (here, the convergence follows from the finiteness of $\mathcal{Z}$). The same argument applies to the safety control. These boil down to our main result of this section:

Theorem 3 Assume $\mathcal{A} = (Q, \Sigma, \mathcal{C}, q_0, \rightarrow, \mathcal{I})$ to be a timed plant with one clock. The unknown switch condition control problem can be solved in $\mathcal{O}(|Q| \cdot (|Q| + |\rightarrow|))$.

Theorem 3 entails that control of a 1C-TA is not more intricate than control of an untimed plant from the perspective of complexity theory. It is well-known that the latter is P-hard, which also gives us the desired lower bound. To conclude this section, we have

Theorem 4 The control problems for timed automata with one clock are P-complete.

5 Control timed automata with two clocks

In this section, we tackle timed automata with two clocks. In this case, the trick in the previous section simply does not work and it seems that there is an EXPTIMEcomplexity blow-up for control. We provide "partial" evidence supporting this by establishing a PSPACE *lower* bound [7]. In details, we propose a polynomial transformation from validity of *Quantified Boolean Formulae* (QBF) into reachability control of timed automata with two clocks and thus obtain that:

Theorem 5 The control problems for timed automata with two clocks are PSPACE-hard.

6 Related Works

As for real-time control, besides the excellent work done by [15], we would mention some earlier works: An extension of the RW framework for *discrete* timed systems has been proposed in [7] and unlike this approach, [13] works within the RW framework on time automata. More recently, [11] considered the timed control for external specification, and [6] and [4] followed this line, by studying the partial observation of the controller or taking MTL formulae as external specifications. As for the complexity of verification (we only mention the most relevant ones), [10] considered the complexity of reachability problem for timed automata. [14] investigated the complexity of timed model checking for one or two clocks.

[7] We say "partial" because this does not mean that *exponential* blow-up is doomed, since PSPACE $\subseteq$ EXPTIME. However, at least it indicates that this problem is not tractable anymore provided one admitting that the PSPACE does not collapse to P.

References

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

[2] E.Asarin, O. Maler and A.Pnueli. Symbolic controller synthesis for discrete and timed systems. Hybrid Systems II, LNCS 999, pp. 1-20, Springer, 1995.

[3] E. Asarin, O. Maler, A.Pnueli and J. Sifakis. Controller synthesis for timed automata. In IFAC Symposium on System Structure and Control, pp. 469-474, Elsevier, 1998.

[4] P. Bouyer, L. Bozzelli and F. Chevalier. Controller synthesis for MTL specifications. In Proc. of CONCUR'06, LNCS 4137, pp. 450-464, Springer, 2006.

[5] P. Bouyer and F. Chevalier. On the control of timed and hybrid systems. *Bulletin of the EATCS*, no. 89, pp. 79-96, 2006.

[6] P. Bouyer, D. D'Souza, P. Madhusudan and A. Petit. Timed control with partial observability. In Proc. of CAV'03, LNCS 2725, pp. 180-192, Springer, 2003.

[7] B. Brandin and W. Wonham. Superviosry control of timed discrete event processes. *IEEE Transaction on Automatic Control*, 39, pp. 329-342, 1994.

[8] F. Cassez, T. Henzinger and J.F. Raskin. A comparison of control problems for timed and hybrid systems. In Proc. of HSCC'02, LNCS 2289, pp. 134-148, Springer 2002.

[9] T. Chen and J. Lu. Towards the complexity of controls for timed automata with a small number of clocks. Technical Report, 2008.

[10] C. Courcoubetis and M. Yannakakis. Minimum and maximum delay problems in real-time systems. *Formal Methods in System Design*, 1(4): 385-415, 1992.

[11] D. D'Souza and P. Madhusudan. Timed control synthesis for external specifications. In Proc. of STACS'02, LNCS 2285, pp. 571-582, Springer, 2002.

[12] T. Henzinger and P. Kopke. Discrete-time control for rectangular hybrid automata. *Theoretical Computer Science*, 221(1-2): 369-392, 1999.

[13] G. Hoffmann and H. Wong-Toi. The input-output control of real-time discrete event system. In Proc. of RTSS'92, pp. 256-265, IEEE, 1992.

[14] F. Laroussinie, N. Markey and Ph. Schnoebelen. Model checking timed automata with one or two clocks. In Proc. of CONCUR'04, LNCS 3170, pp. 387-401, Springer, 2004.

[15] O. Maler, A. Pnueli and J. Sifakis. On the synthesis of discrete controllers for timed systems (an extended abstract). In Proceedng of STACS'95, LNCS 900, pp. 229-242, Springer, 1995.

[16] P. Ramadge and W. Wonham. Superviosry control of a class of discrete event processes. *SIAM Journal of Control and Optimizations*, 25, pp. 206-230, 1987.