Theoretical Computer Science 429 (2012) 292–304

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Modular discrete time approximations of distributed hybrid automata

P.S. Thiagarajan [a,*], Shaofa Yang [b]

[a] School of Computing, National University of Singapore, Singapore

[b] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China

ARTICLE INFO

Keywords:
Distributed hybrid automata
Discrete time approximation

ABSTRACT

We consider a network of controllers that observe and control a plant whose dynamics is determined by a finite set of continuous variables. At any given time a variable evolves at a constant rate. However, a controller can switch the rates of a designated subset of the continuous variables. These mode changes are determined by the current values of a designated subset of the variables that the controller can observe. Each variable's rate is controlled by exactly one controller and its value is observed by at most one controller. We model this setting as a network of hybrid automata and study its discrete time behavior. We show that the set of global control state sequences displayed by the network is regular. More importantly, we show that one can succinctly represent this regular language as a family of communicating finite state automata. We allow the observation of the variables and the changes in the rates of the variables to incur delays. We also permit the digital clocks associated with the controllers to evolve at different – but rationally related – rates.

1. Introduction

We study the discrete time behavior of a plant supervised by a set of controllers. The plant's state space is determined by a finite set of continuous variables $\mathcal{X}$ evolving at rates determined by the controllers. Each controller can observe a subset of $\mathcal{X}$. Based on these observations, it can effect changes to the rates of a subset of $\mathcal{X}$. Each variable is controlled—i.e. its rate can be changed – by exactly one controller and its value is observed by at most one controller. In effect we are assuming a dedicated actuator and sensor with each variable with the understanding that there may be unobserved variables but no uncontrolled ones. In addition, we allow the sensing of the value of a variable and effecting a change to the rate of a variable to incur delays. Finally, we permit the digital clocks associated with controllers to run at different – rationally related – rates. We use a distributed version of hybrid automata to model this rich setting. We then show that the discrete time behavior of the resulting network of hybrid automata can be succinctly represented as a family of communicating finite state automata. A novel feature of our construction is that a large sequential automaton is succinctly represented as a family of communicating smaller finite state automata which move asynchronously.

In the present study, there is no explicit communication between the controllers. However, there will be information flow between them due to the state space of the plant serving as a shared memory. Specifically, the controller p may observe the value of a variable x which is controlled by another controller q. Since p's mode changes depend on the variables it observes, its behavior can be influenced by q via x. We address this issue again in Section 5.

Fig. 1 shows a plant with three controllers. An arc labeled x from the plant to a controller indicates that the controller can observe the value of x. Thus, the set of variables that controller p observes is $\{x_1, x_4\}$. An arc labeled x from a controller to the plant signifies that x is actuated (controlled) by the controller. Thus q actuates the set of variables $\{x_3, x_4\}$.

* Corresponding author.
E-mail addresses: thiagu@comp.nus.edu.sg (P.S. Thiagarajan), sf.yang@siat.ac.cn (S. Yang).

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.12.050

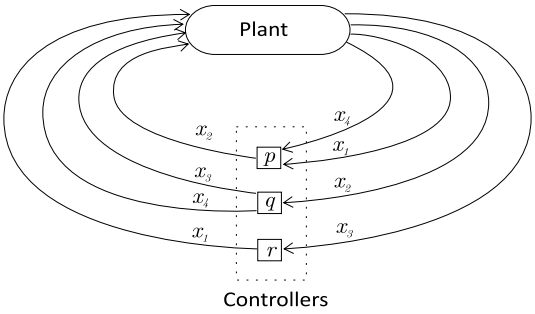


Fig. 1. A plant interacting with three controllers.

In our model there will be one hybrid automaton corresponding to each controller. Each of these automata will be of a familiar kind. It will have finite set of modes (control states) and in each mode a variable will be evolving at a constant rate. Further, the guards governing the mode transitions will be rectangular. Both these restrictions are standard and well-studied [1–4]. Even with these restrictions, the continuous time behavior of a hybrid automaton can be intractable [5,2] and hence one usually imposes two orthogonal types of restrictions to ensure tractability. One type of restriction is to require that the value of a continuous variable should be reset to within a pre-determined range of values whenever the rate of a variable is altered via a mode change [6]. In the setting where the hybrid automaton models the interactions between a digital controller and a continuous plant, this restriction is untenable. Hence we follow here the second type of restriction, namely, that the controllers interact with the plant only at discrete time points. This is natural since the most common controllers are digital. They will interact with the plant only at time points determined by the ticks of their clocks. The basic result here is that the discrete time behavior of the plant–controller combine is a regular language. Further, one can effectively compute a finite state automaton representing this language [1]. Consequently, a variety of verification and (controller) synthesis problems can be solved using standard methods. It turns out that this result holds even if there are delays associated with the sensing of the values of the variables and in actuating rate changes [7]. This "laziness" property makes the model more realistic in plant–controller settings.

We show that in the present setting where we have a network of hybrid automata, the discrete time behavior of the network is still a regular language. Admittedly, this can be shown by a brute force extension of the technique developed in [7]. However, the size of the finite state automaton representing this regular language will be exponential in the number of variables and controllers. Our main result in this paper is that one can instead represent this language in a modular way as a family of communicating finite state automata. More importantly, this representation will be more succinct. Its overall size will be *linear* in the number of variables and in the number of controllers. It will however be exponential in the number of variables that a controller can observe and control. In realistic settings one can expect this number to be a small constant relative to the total number of variables and hence our representation will indeed be succinct. A novel aspect of our representation is that the finite state automata we obtain will move *asynchronously*. Hence the progress of time for the individual automata can be different. But by imposing a simple coordination protocol, similar to that in asynchronous cellular automata [8], we ensure that the discrepancy in the local times of the automata remains bounded. Further, this lets us exploit the basic aspects of the theory of Mazurkiewicz traces [8] to prove our main results. A variety of techniques have been developed by the formal verification community including symbolic representation methods [9] and partial order verification methods [10] to cope with the state explosion problem. These can be readily deployed to analyze our succinct representation of the discrete time behavior of the network of hybrid automata.

In terms of related work, the control systems community has studied in a variety of settings a continuous plant being controlled by a network of discrete controllers (see for instance [11]). A survey of research on networked control systems for instance is provided in [12]. The main objective in this line of research is to minimize the impact of distribution and communication on the control task being implemented rather than on computing a finite state representation of the overall discrete time behavior of the combined system. Decentralized control has also been extensively studied in the setting of discrete event systems (see for instance [13]). However, the plant model and the controllers are all assumed to be finite state machines and thus involve no continuous dynamics.

In the next section we introduce the hybrid automata network model and in Section 3 define its discrete time semantics. In Section 4 we establish our main result. We do so first in a restricted setting to prevent notational clutter from obscuring the key ideas. In Section 5 we then sketch how these restrictions can be relaxed. In the concluding section we summarize and briefly discuss the prospects for future research.

This paper is an improved and expanded version of the conference paper [14]. There it was assumed that each controller controls only one variable but can observe multiple variables. On the other hand while a variable was controlled by exactly one controller, it was allowed to be observed by more than one controller.

Here we instead work with a symmetric and more realistic restriction. A controller can control and observe multiple variables. On the other hand, a variable is controlled by exactly one controller and is observed by at most one controller. In practice there will be a dedicated sensor for each variable and in this sense the present framework is more realistic. We

however show later how the case of a variable being observed by more than one controller can be handled but with increased computational cost.

In the setting we consider here, the technical developments leading to the main result are more challenging due to the laziness property. In fact, we need to develop the main result in the presence of laziness and not add it on as done in [14]. In addition, we discuss in some detail a formal framework in which there is explicit communication between the controllers. We also identify the property of a model being well-behaved and show that it is a decidable property. Further we show that the representation of the discrete time behavior of the subclass of well-behaved models will enjoy a fundamental property. Finally, here we present more complete proofs; in particular, for the case where the controllers operate at different speeds.

2. Distributed hybrid automata

We associate n continuous variables $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ with the plant and fix m controllers $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ and let p, q range over $\mathcal{P}$.

The controller p can observe a subset of the plant's variables denoted R_p and control a subset of the variables denoted W_p. By "control" we mean that at suitable times, it can effect rate changes to the variables in W_p. We require $W_p \cap W_q = \emptyset$ and $R_p \cap R_q = \emptyset$ if $p \neq q$. We also require $\bigcup_p W_p = \mathcal{P}$. Thus each variable will be controlled by exactly one controller and observed by at most one controller.

The controllers are assumed to be digital. For each p there is a clock with T_p as its period. At each tick of its clock, p will sense the values of variables in R_p and based on this, possibly effect a mode change. By a mode change we mean the controller setting new rates to the variables in W_p. To highlight the main ideas, we will first assume that all the clocks run at the same rate (and are perfectly synchronized). We thus fix a suitable granularity of time and assume that $T_p = 1$ for all p.

$\mathbb{R}$ is the set of real numbers and $\mathbb{Q}$ is the set of rational numbers. We fix rationals B_{min}, B_{max} with $B_{min} < B_{max}$ and assume that the feasible values of all the variables lie within the interval $[B_{min}, B_{max}]$. By convention, the plant will get stuck whenever the value of a variable falls outside this interval. We have assumed uniform lower and upper bounds for all the variables merely for convenience.

For $X \subseteq \mathcal{X}$, a rectangular X-guard is a conjunction of inequalities of the form $c \leq x \leq c'$ where c, c' are rationals in $[B_{min}, B_{max}]$ and $x \in X$. In case $c = B_{min}$ ($c' = B_{max}$) we write just $x \leq c'$ ($c \leq x$). Let $Grd(X)$ denote the collection of rectangular guards over X. By an X-valuation, we shall mean a mapping from X to $\mathbb{R}$. An $\mathcal{X}$-valuation will be just called a valuation. The notion of an X-valuation satisfying a rectangular X-guard is defined in the obvious way. From now on we will refer to rectangular guards as just guards.

We will use $\delta^R_{min}, \delta^R_{max}$ to capture delays incurred in sensing the values of the variables. These delay parameters will be rationals in [0, 1] with $\delta^R_{min} \leq \delta^R_{max}$. The interpretation is that the value of x in R_p reported to p at time t_k is the value of x that held at some time in the interval $[t_{k-1} + \delta^R_{min},\ t_{k-1} + \delta^R_{max}]$. In general, the delay assumption will be that the value of x reported at t_k was measured in some bounded interval *before* t_k which could be even earlier than t_{k-1}. However, to simplify the notations, we have assumed that this bounded interval lies between t_{k-1} and t_k.

Similarly, we use the delay parameters $\delta^W_{min}, \delta^W_{max}$ to model delays in actuating changes to the rates of variables. They will also be rationals in [0, 1] with $\delta^W_{min} \leq \delta^W_{max}$. The interpretation is if x is in W_p, and p effects a change in the rate of x at t_k, then this change will not kick in immediately but rather at some time in the interval $[t_k + \delta^W_{min},\ t_k + \delta^W_{max}]$. We note that if $x, y \in W_p$ and p effects changes to the rates of both x and y at t_k, then the time t at which rate change of x kicks in will be, in general, different from t', the time at which the rate change of y does.

We shall further assume that $\delta^W_{max} \leq \delta^R_{min}$ which will ensure that rate change of variable x signaled at time t_k will kick in before the measurement of value of x reported at time t_{k+1} takes place. All of these assumptions can be easily relaxed at the price of increased notational overhead.

The focus of our study will be a *Distributed Hybrid Automaton* ("DHA" for short) of the form $(\{\mathcal{A}_p\}_{p\in\mathcal{P}}, \widehat{\delta})$. For each p, $\mathcal{A}_p$ is a hybrid automaton describing the interactions of the controller p with the plant while $\widehat{\delta}$ is the set of delay parameters. The automata $\{\mathcal{A}_p\}$ will be structures of the form $(S_p, s^{in}_p, R_p, W_p, \longrightarrow_p, Init_p, \rho_p)$ satisfying the following conditions:

- S_p is a finite set of *control states* (modes).
- $s^{in}_p \in S_p$ is the initial control state.
- $R_p \subseteq \mathcal{X}$ is the set of variables observed by p and W_p is the set of variables controlled by p. In what follows, we let $Var_p = R_p \cup W_p$.
- $\longrightarrow_p \subseteq S_p \times Grd(R_p) \times S_p$ is the transition relation such that if $(s_p, \varphi, s'_p) \in \longrightarrow_p$ then $s_p \neq s'_p$.
- $Init_p$ is a map that assigns an interval $[d^x_{min}, d^x_{max}]$ of initial values to each variable in Var_p. We require both d^x_{min} and d^x_{max} to be rational numbers such that $B_{min} \leq d^x_{min} \leq d^x_{max} \leq B_{max}$.
- $\rho_p : S_p \times W_p \to \mathbb{Q}$ where $\rho_p(s, x)$ is the rate of evolution of x in W_p when $\mathcal{A}_p$ resides in the control state s.
- If $x \in Var_p \cap Var_q$ with $p \neq q$ then $x \in R_p$ iff $x \in W_q$. Further $\bigcup_p W_p = \mathcal{P}$.
- Each member of the set of delay parameters $\widehat{\delta} = \{\delta^R_{min}, \delta^R_{max}, \delta^W_{min}, \delta^W_{max}\}$ is a rational in [0, 1]. Further, $\delta^W_{min} \leq \delta^W_{max} \leq \delta^R_{min} \leq \delta^R_{max}$.

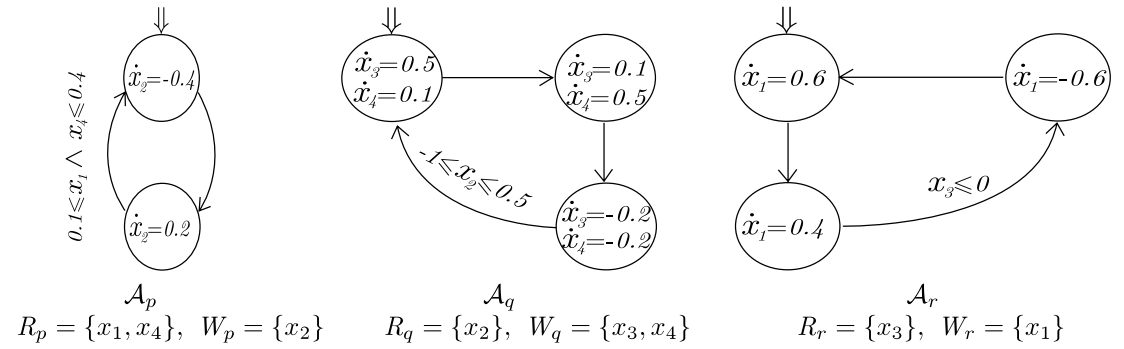


Fig. 2. An example DHA.

Fig. 2 shows a DHA consisting of three automata $\mathcal{A}_p$, $\mathcal{A}_q$, $\mathcal{A}_r$ following the usual graphical notations. To reduce clutter, we have not shown all the guards. We have also not shown the delay parameters.

At time $t_0 = 0$, each x will have its value lying in the interval $[d^x_{min}, d^x_{max}]$ and each $\mathcal{A}_p$ will be in its initial mode. If x is in W_p then it will start evolving at rate $\rho_p(s^{in}_p, x)$. No automaton signals a mode change at t_0.

Suppose at t_k $(k > 0)$ the current values of the variables in $\mathcal{X}$ are given by the valuation V_k and $\mathcal{A}_p$ is in the control state s_p for each p.

First assume $\mathcal{A}_p$ has not signaled a mode change at t_k and hence it was in the control state s_p at time t_{k-1} too (or $k = 0$) . Then each x in W_p will evolve at rate $\rho_p(s_p, x)$ up to t_{k+1}. Consequently the value of x reported to q with $x \in R_q$ will be $V_k(x) + \theta_x \cdot \rho_p(s_p, x)$ where $t_k + \theta_x$ is the time at which its value was measured with $\theta_x \in [\delta^R_{min},\ \delta^R_{max}]$.

Assume next that $\mathcal{A}_p$ has signaled a mode change at t_k and its mode at t_{k-1} was s''_p. Then each x in W_p will evolve at rate $\rho_p(s''_p, x)$ up to some time point $t_k + \tau_x$ with τ_x in $[\delta^W_{min},\ \delta^W_{max}]$. At $t_k + \tau_x$ its rate will change to $\rho_p(s_p, x)$ and it will evolve at this rate up to t_{k+1}. Thus the value of x reported to q with $x \in R_q$ will be $V_k(x) + \tau_x \cdot \rho_p(s''_p, x) + (\theta_x - \tau_x) \cdot \rho_p(s_p, x)$ with $t_k + \theta_x$ being the time at which the value of x was measured where $\theta_x \in [\delta^R_{min},\ \delta^R_{max}]$. We note again that τ_x, θ_x can be different for different x's. At t_{k+1} the following actions occur instantaneously.

- For each x in $\mathcal{X}$, if $x \in R_p$ then $\mathcal{A}_p$ will receive the value of x measured at some time in $[t_k + \delta^R_{min}, t_k + \delta^R_{max}]$.
- Based on the received observed values of the variables in R_p, $\mathcal{A}_p$ will determine if any of the guards associated with the outgoing transitions at s_p is satisfied and hence enabled.
- If an outgoing transition is enabled then the automaton $\mathcal{A}_p$ may choose one of the enabled transitions and move to a new control state, say, s'_p. In this case, it will signal a mode change and hence the new rate, namely $\rho_p(s'_p, x)$, for each $x \in W_p$.
- In case, no mode change is effected all the variables in W_p will continue to evolve at the rate $\rho_p(s_p, x)$ starting from t_{k+1}.

One could force $\mathcal{A}_p$ to make a mode change in case one or more outgoing transitions are enabled by assigning state invariants to its control states. These invariants will be boolean combinations of atomic assertions of the forms $x < c$, $x > c'$ with $x \in R_p$ and $c, c' \in \mathbb{Q}$. We would then demand that at t_{k+1}, the automaton $\mathcal{A}_p$ can choose to stay in s_p only if the state invariant associated with s_p is not violated. Our results will easily go through in the presence of such invariants.

3. The transition system semantics

Through the rest of this section, we fix a distributed hybrid automaton $\mathcal{H}$ as described in the previous section with the associated notations. We shall define the discrete time dynamics of $\mathcal{H}$ in terms of the infinite state transition system $TS_{\mathcal{H}}$. We will often drop the subscript $\mathcal{H}$. States of TS will be termed *configurations*. A configuration is a triple (α, V, α') where α, α' are maps from $\mathcal{P}$ to $\bigcup_p S_p$ such that $\alpha(p), \alpha'(p) \in S_p$ for each p and V a valuation. The idea is that if $\mathcal{H}$ is in the configuration (α, V, α') at time t_k then $\alpha(p)$ is the control state of controller $\mathcal{A}_p$ at time t_k while $V(x)$ is the *actual* value of x at time t_k. On the other hand, $\alpha'(p)$ is the control state of $\mathcal{A}_p$ at the previous time instant t_{k-1}. As already hinted at, α' will be used to determine if a mode change has been signaled by $\mathcal{A}_p$ at t_k. This will become clearer when we define a transition relation on the set of configurations.

Let $Conf$ denote the collection of configurations. The set of *initial* configurations $Conf^{in}$ is given by: (α, V, α') is in $Conf^{in}$ iff for each p, $\alpha(p) = s^{in}_p = \alpha'(p)$ and for each variable x, $V(x) \in Init_p(x)$ with $x \in W_p$. A configuration (α, V, α') is *feasible* if for every x, $B_{min} \leq V(x) \leq B_{max}$. Clearly, every initial configuration is feasible.

We define the transition relation $\Longrightarrow \subseteq Conf \times Conf$ as follows.

$(\alpha, U, \alpha') \Longrightarrow (\beta, V, \beta')$ iff (α, U, α') is feasible and $\beta'(p) = \alpha(p)$ for each p. Moreover, there exist reals $\{\tau_x\}_{x \in \mathcal{X}}$ in $[\delta^W_{min}, \delta^W_{max}]$ and $\{\theta_x\}_{x \in \mathcal{X}}$ in $[\delta^R_{min}, \delta^R_{max}]$, which satisfy the following conditions. In stating these conditions we shall assume that $\alpha(p) = s_p$, $\alpha'(p) = s'_p$ and $\beta(p) = \hat{s}_p$, $\beta'(p) = \hat{s}'_p$ for each p.

- For each $x \in \mathcal{X}$, $V(x) = U(x) + \tau_x \cdot \rho_p(s'_p, x) + (1 - \tau_x) \cdot \rho_p(s_p, x)$, where p is such that $x \in W_p$.

  Intuitively, $t_k + \tau_x$ is the time at which a rate change of x kicks in if it was signaled at t_k. In case $s_p = s'_p$ then there was no mode change at t_k and we will get $V(x) = U(x) + \rho_p(s_p, x)$ which is the value of x obtained by evolving at rate $\rho_p(s_p, x)$ for one unit of time while starting with $U(x)$ at time t_k.
- If $s_p \neq \hat{s}_p$, then there exists a transition $(s_p, \varphi, \hat{s}_p) \in \longrightarrow_p$, such that the R_p-valuation V' given by $V'(x) = V(x) + \tau_x \cdot \rho_p(s'_p, x) + (\theta_x - \tau_x) \cdot \rho_p(s_p, x)$ for each $x \in R_p$, satisfies the guard φ.

  Intuitively, if at time t_k the DHA is in configuration (α, U, α'), then $t_k + \theta_x$ is the time at which the value reported at t_{k+1} is observed (recall that $t_k + \tau_x$ is the time at which the rate change of x kicks in if it had been signaled at t_k).

Now we define the transition system $TS_{\mathcal{H}}$ to be $(RC, Conf^{in}, \Longrightarrow_{RC})$ where RC, the set of reachable configurations, is the least set such that $Conf^{in} \subseteq RC$. Further, if $\xi \in RC$ and $\xi \Longrightarrow \xi'$ then $\xi' \in RC$. Moreover, $\Longrightarrow_{RC}$ is the restriction of $\Longrightarrow$ to $RC \times RC$. Abusing notation, we will often write $\Longrightarrow$ instead of $\Longrightarrow_{RC}$. We note that TS will be an infinite state system unless $Init_p(x_p)$ is a singleton set for every p and that $\delta^W_{min} = \delta^W_{max}$, $\delta^R_{min} = \delta^R_{max}$.

We will say that $\mathcal{H}$ is *well-behaved* if every reachable configuration of $TS_{\mathcal{H}}$ is feasible. Well-behaved DHAs constitute a natural and important subclass. In Section 4 we will show that it is decidable whether a DHA is well-behaved. We will also show that our representation of the discrete time behavior of a DHA enjoys a strong property if the DHA is well-behaved.

A *run* of $\mathcal{H}$ is a finite sequence of configurations $\xi_0\xi_1 \ldots \xi_k$ such that $\xi_0 \in Conf^{in}$ and $\xi_i \Longrightarrow \xi_{i+1}$ for $0 \leq i < k$.

A *global control state* is a map s from $\mathcal{P}$ to $\bigcup_p S_p$ such that $s(p) \in S_p$ for each p. The global control state induced by the configuration ξ is denoted as $st(\xi)$ and it is the state s satisfying $s(p) = \alpha(p)$ for each p, where $\xi = (\alpha, V, \alpha')$.

The control state sequence induced by the run $\sigma = \xi_0\xi_1 \ldots \xi_k$ is denoted $st(\sigma)$ and it is the sequence $st(\xi_0)st(\xi_1) \ldots st(\xi_k)$. We let $L(\mathcal{H})$ denote the set of control state sequences of $\mathcal{H}$.

Based on the results in [7], it is easy to show that $L(\mathcal{H})$ is regular and a finite state automaton representing this language can be effectively constructed. However, the size of this finite state automaton in terms of its number of states, will be exponential in the number of controllers and in the number of variables. Our main result in this paper is that this state explosion problem in the *representation* of $L(\mathcal{H})$ can be avoided. We show that $L(\mathcal{H})$ can be succinctly represented as a family of finite state automata. This family can be effectively constructed and its overall size will be linear in the number of variables and the number of controllers but exponential in (the maximum of) $|Var_p|$. As pointed out earlier, $|Var_p|$ will often be much smaller than $|\mathcal{X}|$ and $|\mathcal{P}|$. One can also associate actions with the transitions of $\mathcal{A}_p$, define a language of action sequences, show that the resulting language is regular and construct a family of finite state automata representing this language. Consequently, one can effectively tackle a variety of verification and controller synthesis problems related to the discrete time behavior of $\mathcal{H}$.

4. The representation result

In this section, we establish our main result, namely, the language of control state sequences of a DHA can be represented succinctly as a family of finite state automata. For doing so, we fix a distributed hybrid automaton $\mathcal{H}$ with the associated notations. The finite state automata we construct will communicate with each other in the manner of asynchronous cellular automata [8]. However our presentation will be self-contained.

4.1. The communication graph structure

Through the rest of this section, we set $\Upsilon = \mathcal{P} \cup \mathcal{X}$ and let η, η' range over Υ.

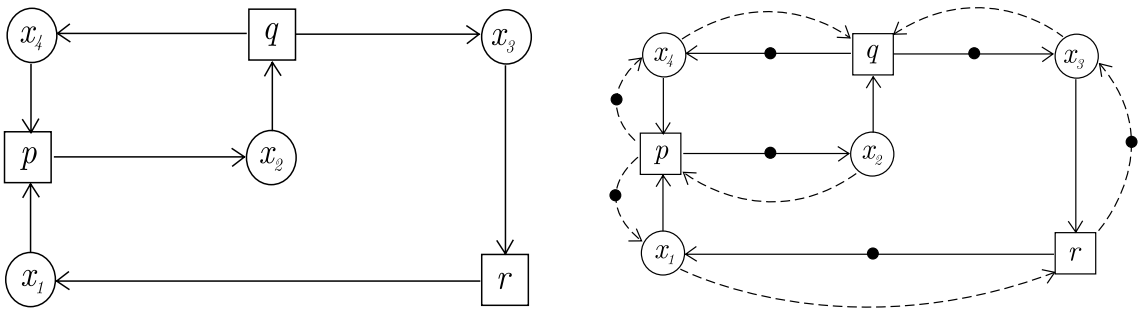
We will construct a family of automata $\{\mathcal{B}_\eta\}_{\eta \in \Upsilon}$ (hereafter the subscript is dropped). We will then define $\mathcal{B}$, the product of $\{\mathcal{B}_\eta\}$ in a standard way. $\mathcal{B}$ will contain a richer set of behaviors $L(\mathcal{H})$. Through a simple (automata-theoretic) operation we then restrict $\mathcal{B}$ to a finite state automaton that accepts precisely $L(\mathcal{H})$. We emphasize again that the analysis of many local and global properties can be carried out using the succinct representation $\{\mathcal{B}_\eta\}$ without having to construct $\mathcal{B}$.

The moves of $\mathcal{B}_p$ will depend on its current state and on the current states of the automata in $\{\mathcal{B}_x\}_{x \in Var_p}$ (recall that $Var_p = R_p \cup W_p$). On the other hand, the moves of the automaton $\mathcal{B}_x$ will depend on its current state and on the current states of $\mathcal{B}_p$ and $\mathcal{B}_q$ with $x \in W_p$, $x \in R_q$. We note that p and q are uniquely determined for each x.

The flow of information between the automata in $\{\mathcal{B}_\eta\}$ is conveniently represented by the *communication graph* of $\mathcal{H}$ denoted $CG_{\mathcal{H}}$. As before, we will often drop the subscript $\mathcal{H}$. We define $CG = (\Upsilon, A)$ where $A = \{(p, x) \mid x \in W_p\} \cup \{(x, p) \mid x \in R_p\}$. Thus CG will be bipartite with arcs going from variable nodes to controller nodes and from controller nodes to variable nodes.

Fig. 3(i) displays the communication graph of the DHA in Fig. 2. We have used circles to denote the variables and boxes to denote the controllers merely to emphasize the bipartite nature of the communication graph. For the present the reader should ignore Fig. 3(ii) and not attach any significance to the Petri net like graphical notations chosen for representing a communication graph.

$\mathcal{B}_p$ will keep track of the control state of $\mathcal{A}_p$ at t_k as well as its control state at t_{k-1}. Using this, it can infer the rates of the variables in W_p during the interval $[t_k, t_{k+1}]$. On the other hand, $\mathcal{B}_x$ will track the value of x at time t_k as well as the observed value of x reported at time t_k. Since $\mathcal{B}_x$ is required to be finite state, we will quantize $\mathbb{R}$ into *finitely* many sub-intervals and represent a value of a continuous variable by the sub-interval it lies in.



(i) Communication graph (ii) Marked graph representation

Fig. 3.

4.2. The quantization of the value space

We quantize $\mathbb{R}$ to ensure that each automaton in our family is finite state. Let Δ be the largest rational which integrally divides every rational in the set $\{1, \delta^R_{min}, \delta^R_{max}, \delta^W_{min}, \delta^W_{max}\}$. Let Ω be the least set of rational numbers given by:

- $B_{min}, B_{max} \in \Omega$.
- Suppose $Init_p(x) = [d^x_{min}, d^x_{max}]$. Then $d^x_{min}, d^x_{max} \in \Omega$.
- $\rho_p(s, x) \cdot \Delta$ is in Ω for every p and every s in S_p and every x in W_p.
- Suppose $(s_p, \varphi, s'_p) \in \longrightarrow_p$ and c appears in φ. Then $c \in \Omega$.

Now let Γ be the largest rational which integrally divides every number in Ω. Let N_{min} and N_{max} be integers such that $B_{min} = N_{min} \cdot \Gamma$ and $B_{max} = N_{max} \cdot \Gamma$. We then partition $\mathbb{R}$ into finitely many intervals $(-\infty, N_{min} \cdot \Gamma)$, $\{N_{min} \cdot \Gamma\}$, $(N_{min} \cdot \Gamma, (N_{min} + 1) \cdot \Gamma)$, $\{(N_{min} + 1) \cdot \Gamma\}, \ldots, ((N_{max} - 1) \cdot \Gamma, N_{max} \cdot \Gamma)$, $\{N_{max} \cdot \Gamma\}$, $(N_{max} \cdot \Gamma, \infty)$. Note that for convenience we denote the interval $[v, v]$ as simply $\{v\}$.

Let $\mathcal{I}$ be the collection of these open intervals and closed singleton intervals. Clearly, $\mathcal{I}$ is a finite set. Now we define the map $\|\cdot\| : \mathbb{R} \to \mathcal{I}$ via: $\|v\| = I$ iff $v \in I$. Next suppose $X \subseteq \mathcal{X}$ and V is an X-valuation. Then $\|V\|$ is the map $\|V\| : X \to \mathcal{I}$ given by: $\|V\|(x) = \|V(x)\|$ for each x.

4.3. The construction of the family of local automata

The following technical lemma will play a key role in ensuring that our quantization leads to a family of finite state automata with the desired properties. We set $\mathcal{RT}_p = \{\rho_p(s_p, x) \mid s_p \in S_p, x \in W_p\}$ for each p and use this notation in the lemma and elsewhere.

Lemma 1. *The following assertions hold:*

(i) *Suppose $X \subseteq \mathcal{X}$. Let φ be an X-guard which appears in $\mathcal{H}$ and U, V be X-valuations such that $\|U\| = \|V\|$. Then U satisfies φ iff V satisfies φ.*

(ii) *Suppose $X \subseteq \mathcal{X}$. Given a collection of X-indexed intervals $\{I_x\}_{x \in X}$ where $I_x \in \mathcal{I}$, one can effectively check whether there exists an X-valuation U such that $U(x) \in I_x$ for each x and U satisfies φ.*

(iii) *Let $u, v \in \mathbb{R}$ with $\|u\| = \|v\|$. Then $u \in [B_{min}, B_{max}]$ iff $v \in [B_{min}, B_{max}]$.*

(iv) *Let $r, r' \in \mathcal{RT}_p$ and $I, J \in \mathcal{I}$. For u in $\mathbb{R}$, let $\Phi_{r,r',I,J}(u)$ be the predicate: there exist reals τ in $[\delta^W_{min}, \delta^W_{max}]$ and θ in $[\delta^R_{min}, \delta^R_{max}]$ such that $u + \tau \cdot r' + (1 - \tau) \cdot r$ is in I and $u + \tau \cdot r' + (\theta - \tau) \cdot r$ is in J.*

Then for u, v with $\|u\| = \|v\|$ and $u \in [B_{min}, B_{max}]$, it will be the case that $\Phi_{r,r',I,J}(u)$ holds iff $\Phi_{r,r',I,J}(v)$ holds. Moreover, given r, r', I, J and an interval M in $\mathcal{I}$, one can effectively determine whether there exists u in M such that $\Phi_{r,r',I,J}(u)$ holds.

Proof. Let $\varphi = \bigwedge_{x \in X} c_x \leq x \leq c'_x$. Consider $x \in X$. Let $\|U\|(x) = \|U\|(x) = I_x$ and suppose $I_x = (j \cdot \Gamma, (j + 1) \cdot \Gamma)$ with $N_{min} \leq j < N_{max}$. Since c_x and c'_x are integral multiples of Γ, it follows that $c_x \leq U(x) \leq c'_x$ (and $c_x \leq V(x) \leq c'_x$) iff I_x is contained in $[c_x, c'_x]$. The same observation can be established in a similar but simpler way for the cases that I_x is a singleton interval or lies outside $[B_{min}, B_{max}]$. From these arguments, it is clear that U satisfies φ iff V satisfies φ. This establishes the first two parts of the lemma.

Part (iii) follows from the definitions.

The proof of (iv) is the heart of the matter and is more involved. Consider first the case where I, J are both open intervals. A basic fact in the first order theory of reals is that the sentence $\exists z \in \mathbb{R} \bigwedge_{i \in Ind} a_i < z < b_i$, where Ind is a finite index set and each a_i, b_i are sentences not involving z, is equivalent to a sentence of the form $\bigwedge_{i,j \in Ind} a_i < b_j$ in which z does not appear. This follows from the Fourier–Motzkin quantifier elimination procedure [15]. Using this and the definition of Γ, one can deduce that $\Phi_{r,r',I,J}(u)$ is in fact equivalent to the assertion that u lies in an interval whose end points are integral multiples of Γ (where these end points will depend on r, r', I, J). The claim can then be established by considering the cases whether

$\|u\|$ is a singleton closed interval or is an open interval. The cases where I or J or both are closed singleton intervals can be similarly handled. □

We now turn to the construction of the collection of automata $\{\mathcal{B}_\eta\}$. We first recall that a move in TS at t_{k+1} consists of (i) the value of each x at t_{k+1} being determined from the value at t_k based on (a) its rate of evolution at time t_{k-1} (b) the time at which a possible rate change induced at t_k kicks in (c) its rate of evolution at t_k (ii) the controller p reading the observed values of each $x_p \in R_p$ at t_{k+1} and determining if there is to be mode change; if so, choosing a new control state and the corresponding new rates of evolution for the variables in W_p. Clearly the observed value of x depends on the time at which the possible rate change induced by t_k kicks in *and* the time at which measurement of x takes place. Such a move in TS will be simulated by $\{\mathcal{B}_\eta\}$ in two stages.

For explaining this and for later use we define the *neighbors* of a node η denoted $Nbr(\eta)$ in the communication graph. It is given by, $Nbr(\eta) = \{\eta' \mid (\eta', \eta) \in A\} \cup \{\eta'' \mid (\eta, \eta'') \in A\}$.

For a variable x, note that $Nbr(x) = \{p, q\}$ for the unique p, q with $x \in W_p, x \in R_q$. The automaton $\mathcal{B}_x$ will read – but not alter – the current states of automata $\{\mathcal{B}_p, \mathcal{B}_q\}$. Using this information, it will *simultaneously* update the actual quantized value of x and the quantized observed value of x. It will do so by using its current quantized value, the rate of its evolution prescribed by the current mode of p and previous mode of p as recorded by $\mathcal{B}_p$ (where $x \in W_p$), and by "guessing" the time at which the possible rate change of x has kicked in and the time at which measurement of x has taken place in the just concluded time interval. Due to delays in actuating and sensing, there may be more than one choice of current and quantized observed value of x. On the other hand, if $q \neq p$, then the information that automaton $\mathcal{B}_x$ read from $\mathcal{B}_q$ will be used only for proper coordination as will become evident below.

For a controller p, note that $Nbr(p) = R_p \cup W_p$. The automaton $\mathcal{B}_p$ will read – but not alter – the states of the automata $\{\mathcal{B}_x\}_{x \in Nbr(p)}$ to obtain the quantized observed values of the variables $x \in R_p$ and determine the new control state. Again, in case $y \in Nbr(p)$ but $y \notin R_p$, then state of the automaton $\mathcal{B}_y$ is read only for proper coordination.

To coordinate the moves of automata in $\{\mathcal{B}_\eta\}$, each state of $\mathcal{B}_x$ and of $\mathcal{B}_p$ will also maintain a parity bit. Initially, every automaton will be in its initial state with parity 0. The automaton $\mathcal{B}_x$ can make a move only when its parity is the same as that of $\mathcal{B}_p$ for every $p \in Nbr(x)$. It will flip its parity whenever it makes a move. On the other hand, $\mathcal{B}_p$ will make a move only when its parity is different from that of $\mathcal{B}_x$ for every $x \in Nbr(p)$. It will also flip its parity whenever it makes a move. The automata will move asynchronously. Hence at any given time, different automata in $\{\mathcal{B}_\eta\}$ may have made different number of moves and have different views on how much global time has passed. Our construction will ensure that automata belonging to the same connected component of the communication graph of $\mathcal{H}$ will be out of synch by only a *bounded* amount.

By convention, $\mathcal{B}_x$ gets stuck if the quantized value of x it is maintaining falls outside $[B_{min}, B_{max}]$. Once $\mathcal{B}_x$ gets stuck, any $\mathcal{B}_p$ with $p \in Nbr(x)$ will also get stuck. Due to our construction every automaton that lies in the same connected component of the communication graph will then get stuck within a bounded amount of time.

We will first describe the states of the automata in $\{\mathcal{B}_\eta\}$. A state of $\mathcal{B}_x$ will be of the form (I, J, β), where $I \in \mathcal{I}$ is the quantized interval in which the current value of x lies, $J \in \mathcal{I}$ is the quantized interval in which the observed value of x lies, and $\beta \in \{0, 1\}$ is a parity bit. We let Λ_x be the set of states of $\mathcal{B}_x$. Thus $\Lambda_x = \mathcal{I} \times \mathcal{I} \times \{0, 1\}$. The set of initial states of $\mathcal{B}_x$ denoted Λ_x^{in} is $\{(\|v\|, \|v\|, 0) \mid v \in Init_p(x)\}$ with $x \in W_p$.

A state of $\mathcal{B}_p$ will be of the form (s_p, s'_p, β) where $s_p \in S_p$ is the current state of $\mathcal{A}_p$, s'_p the control state of $\mathcal{A}_p$ at the previous time instant, and $\beta \in \{0, 1\}$ is the parity bit. We let Λ_p be the set of states of $\mathcal{B}_p$. Thus $\Lambda_p = S_p \times S_p \times \{0, 1\}$. The set of initial states of $\mathcal{B}_p$ denoted Λ_p^{in} is a singleton set and consists of $(s_p^{in}, s_p^{in}, 0)$. Clearly Λ_η is a finite set for every η.

To define the transition relations, we will make use of the notion of Q-states. Suppose $Q \subseteq \Upsilon$. Then a Q-state is a map which assigns to every element η in Q a state in Λ_η. Let Λ_Q be the set of Q-states. If $Q = \{\eta\}$ is a singleton, we will say η-state instead of $\{\eta\}$-state.

The transition relation of each $\mathcal{B}_x$, denoted $\rightsquigarrow_x$, is a subset of $\Lambda_x \times \Lambda_{Nbr(x)} \times \Lambda_x$ defined as follows. Suppose $Nbr(x) = \{p, q\}$ with $x \in W_p$ and $x \in R_q$. Let $\lambda_x = (I, J, \beta)$ and $\hat{\lambda}_x = (\hat{I}, \hat{J}, \hat{\beta})$ be x-states. Let z be a $Nbr(x)$-state with $z(p) = (s_p, s'_p, \beta_p)$ and $z(q) = (s_q, s'_q, \beta_q)$. Then $(\lambda_x, z, \hat{\lambda}_x) \in \rightsquigarrow_x$ iff the following conditions are satisfied:

- $\beta = \beta_p = \beta_q$, and $\hat{\beta} = 1 - \beta$.
- I is contained in $[B_{min}, B_{max}]$.
- There exist a value u in I which satisfies the following: there exist reals τ_x in $[\delta_{min}^W, \delta_{max}^W]$, θ_x in $[\delta_{min}^W, \delta_{max}^W]$, such that $u + \tau_x \cdot \rho_p(s'_p, x) + (1 - \tau_x) \cdot \rho_p(s_p, x)$ is in $\hat{I}$, and $u + \tau_x \cdot \rho_p(s'_p, x) + (\theta_x - \tau_x) \cdot \rho_p(s_p, x)$ is in $\hat{J}$.

Note that the condition asserted on u is precisely the predicate $\Phi_{r,r',\hat{I},\hat{J}}(u)$ in part (iv) of Lemma 1 with $r = \rho_p(s_p, x)$ and $r' = \rho_p(s'_p, x)$. Hence the following proposition can be easily shown to hold:

Proposition 2. *The transition relation $\rightsquigarrow_x$ is well-defined and can be effectively computed.*

Note the first clause in the definition of $\rightsquigarrow_x$ ensures that $\mathcal{B}_x$ can make a move only when its parity is the same as that of $\mathcal{B}_p$ and of $\mathcal{B}_q$ where $Nbr(x) = \{p, q\}$. Further, $\mathcal{B}_x$ flips its parity at the end of the move. The second condition dictates the current quantized value of x to be within the feasible value range. The last condition ensures that the quantized value of x

and the quantized observed value of x are updated according to the rates dictated by the current control state of p and the control state of p at the previous time instant.

The transition relation $\rightsquigarrow_p$ of $\mathcal{B}_p$, is the subset of $\Lambda_p \times \Lambda_{Nbr(p)} \times \Lambda_p$ defined as follows. Let $\lambda_p = (s, s', \beta)$ and $\hat{\lambda}_p = (\hat{s}, \hat{s}', \hat{\beta})$ be p-states and z be a $Nbr(p)$-state. Assume $z(x) = (I_x, J_x, \beta_x)$ for every x in $Nbr(p)$. Then $(\lambda_p, z, \hat{\lambda}_p) \in \rightsquigarrow_p$ iff the following conditions are satisfied:

- $\beta \neq \beta_x$ for each x, and $\hat{\beta} = 1 - \beta$.
- $\hat{s}' = s$.
- I_x is contained in $[B_{min}, B_{max}]$ for each x.
- Either there exists a transition $(s, \varphi, \hat{s})$ of the hybrid automaton $\mathcal{A}_p$ such that φ is satisfied by some $Nbr(p)$-valuation V with $V(x) \in J_x$ for each x, or $s = \hat{s}$.

Again, it is easy to argue that this transition relation is well-defined and can be effectively computed. The last condition asserts that the current control state of $\mathcal{B}_p$ is updated according to the (quantized) values of variables that p observes. This completes the construction of the family of automata $\{\mathcal{B}_\eta\}$.

4.4. The product of the family of local automata

We next define $\mathcal{B}$, the product (parallel composition) of $\{\mathcal{B}_\eta\}$. To repeat, $\mathcal{B}$ is only needed to establish that $\{\mathcal{B}_\eta\}$ recognizes $L(\mathcal{H})$. Verification problems concerning $L(\mathcal{H})$ are to be addressed in terms of $\{\mathcal{B}_\eta\}$ using the variety of currently available methods for tackling the state explosion problem.

Anticipating later needs, it will be convenient to associate action labels with the transitions of $\mathcal{B}$. We set $\Sigma_x = \mathbb{I} \times \mathbb{I} \times \{x\} \times \mathbb{I} \times \mathbb{I}$ for each x. The letter $(I, J, x, \hat{I}, \hat{J})$ will be used to label a transition of $\mathcal{B}$ in which $\mathcal{B}_x$ makes a move from (I, J, β) to $(\hat{I}, \hat{J}, 1 - \beta)$ where $\beta \in \{0, 1\}$. For each p, we define $\Sigma_p = S_p \times S_p \times \{p\} \times S_p \times S_p$. The letter $(s, s', p, \hat{s}, \hat{s}')$ will be used to record a move of $\mathcal{B}_p$ from (s, s', β) to $(\hat{s}, \hat{s}', 1-\beta)$ where $\beta \in \{0, 1\}$. We set $\Sigma = \bigcup_{\eta \in \Upsilon} \Sigma_\eta$ and let e, e' range over Σ.

We now define $\mathcal{B} = (\Lambda, \Lambda^{in}, \hookrightarrow)$ via:

- Λ is the set of Υ-states.
- Λ^{in} is the set of initial states and is given by: $z \in \Lambda^{in}$ iff $z(\eta) \in \Lambda_\eta^{in}$ for every $\eta \in \Upsilon$.
- $\hookrightarrow \subseteq \Lambda \times \Sigma \times \Lambda$ is the transition relation and is the least set which satisfies the following.

1. Suppose $(\lambda_x, u_Q, \hat{\lambda}_x)$ is a transition of $\mathcal{B}_x$ with $Q = Nbr(x)$. Let $z, \hat{z} \in \Lambda$ such that $z(x) = \lambda_x$, $z(p) = u_Q(p)$ for each $p \in Nbr(x)$, $\hat{z}(x) = \hat{\lambda}_x$, and moreover $z(\eta) = \hat{z}(\eta)$ for each $\eta \in \Upsilon$ with $\eta \neq x$. Let $\lambda_x = (I, J, \beta)$, $\hat{\lambda}_x = (\hat{I}, \hat{J}, \hat{\beta})$. Then $(z, e, \hat{z}) \in \hookrightarrow$ where $e = (I, J, x, \hat{I}, \hat{J})$.
2. Suppose $(\lambda_p, u_Q, \hat{\lambda}_p)$ is a transition of $\mathcal{B}_p$ with $Q = Nbr(p)$. Let $z, \hat{z} \in W$ such that $z(p) = \lambda_p$, $z(x) = u_Q(x)$ for each $x \in Nbr(p)$, $\hat{z}(p) = \hat{\lambda}_p$, and moreover $z(\eta) = \hat{z}(\eta)$ for each $\eta \in \Upsilon$ with $\eta \neq p$. Let $\lambda_p = (s, s', \beta)$, $\hat{\lambda}_p = (\hat{s}, \hat{s}', \hat{\beta})$. Then $(z, e, \hat{z}) \in \hookrightarrow$ where $e = (s, s', p, \hat{s}, \hat{s}')$.

Thus $\mathcal{B}$ is a finite state automaton which captures the global interleaved behavior of $\{\mathcal{B}_\eta\}$. It should be clear to the informed reader that $\{\mathcal{B}_\eta\}$ interact like asynchronous cellular automata [8].

We will first define the behavior of $\mathcal{B}$ in terms of its *firing sequences*. Then we will identify the subset of *complete* firing sequences. Every complete firing sequence will induce in a canonical way a control state sequence of $\mathcal{H}$. We will then argue that this induced set of control sequences is precisely $L(\mathcal{H})$.

$FS_{\mathcal{B}} \subseteq \Sigma^\star$ will denote the set of firing sequences of $\mathcal{B}$. As usual we will often drop the subscript $\mathcal{B}$. This set is defined inductively. In doing so, we will also inductively define an extended version of $\hookrightarrow$. By abuse of notation this extension will also be denoted as $\hookrightarrow$. We will also often write $z \overset{e}{\hookrightarrow} \hat{z}$ instead of $(z, e, \hat{z}) \in \hookrightarrow$.

- The null sequence ϵ is in FS. And $z \overset{\epsilon}{\hookrightarrow} z$ for each $z \in \Lambda^{in}$.
- Suppose $\sigma \in FS$ and $z^{in} \overset{\sigma}{\hookrightarrow} z$ where $z^{in} \in \Lambda^{in}$. If $z \overset{e}{\hookrightarrow} \hat{z}$ where $e \in \Sigma$, then $\sigma e \in FS$ and $z^{in} \overset{\sigma e}{\hookrightarrow} \hat{z}$.

We let $\#(\sigma, \eta)$ denote the number of times letters in Σ_η appear in the firing sequence σ. This represents the total number of times the automaton $\mathcal{B}_\eta$ has moved during the execution of σ.

Next we define the firing sequence σ to be *complete* iff $\#(\sigma, \eta) = \#(\sigma, \eta')$ for every $\eta, \eta' \in \Upsilon$. Thus σ is complete iff every automaton $\mathcal{B}_\eta$ has made equal number of moves during the execution of σ.

Using the definitions of $\{\mathcal{B}_\eta\}$ and $\mathcal{B}$ it is tedious but straightforward to establish the following result for the case where the communication graph is connected. The more general case will be disposed off in the later part of this section.

Proposition 3. (i) *Let σ be a firing sequence and $x \in Nbr(p)$. Then $\#(\sigma, p) \leq \#(\sigma, x) \leq 1 + \#(\sigma, p)$.*
(ii) *Suppose the communication graph CG is connected. Then there exists a non-negative integer K which depends only on CG such that for every firing sequence σ and every η, η', $|\#(\sigma, \eta) - \#(\sigma, \eta')| \leq K$.*

Note that the second part of the proposition bounds the amount by which the automata in $\{\mathcal{B}_\eta\}$ can get away from each other. The proof of Proposition 3 basically exploits the fact that a parity of an automaton in $\{\mathcal{B}_\eta\}$ can flip twice only if all its neighboring automata have flipped their parities at least once.

We can now extract a control state sequence from a complete firing sequence. To this end, let σ be a complete firing sequence and $h = \#(\sigma, \eta)$ for some η. By the definition of a complete firing sequence, h does not depend on the choice of η. We now define $s_0 s_1 \ldots s_h$ to be the control state sequence induced by σ as follows.

For each p and for each $j \in \{0, 1, \ldots, h\}$ fix a prefix π_j^p of σ such that $\#(\pi_j^p, p) = j$. Let $z_j^{in} \overset{\pi_j^p}{\hookrightarrow} z_j^p$ for each sequence where $z_j^{in} \in \Lambda^{in}$. Then for $0 \leq j \leq h$, s_j is the global control state given by: $s_j(p) = z_j^p(p)$ for each p.

According to our definition there is a great deal of choice when it comes to fixing the prefixes π_j^p. Using the definition of $\mathcal{B}$, it is easy to argue however that all the different choices will lead to the same control state sequence. We let $L(\mathcal{B})$ denote the set of control state sequences induced by the set of complete firing sequences of $\mathcal{B}$. Our main result is that $L(\mathcal{H}) = L(\mathcal{B})$.

For proving this result, it will be convenient to use Mazurkiewicz traces [8] to group firing sequences into equivalence classes. We recall that a Mazurkiewicz trace alphabet is a pair (Θ, I_Θ) where Θ is a finite alphabet and $I_\Theta \subseteq \Theta \times \Theta$ is an irreflexive and symmetric independence relation. $D_\Theta = \Theta \times \Theta - I_\Theta$ is the dependence relation and is reflexive and symmetric.

We first observe that there is a natural dependence relation $D_\Sigma \subseteq \Sigma \times \Sigma$ by: $e\, D_\Sigma\, f$ iff one of the following holds:

(i) $e = f$.
(ii) Let $e = (I, J, x, \hat{I}, \hat{J}), f = (s, s', p, \hat{s}, \hat{s}')$. Then $(x, p) \in A$ or $(p, x) \in A$.
(iii) Let $e = (s, s', p, \hat{s}, \hat{s}'), f = (I, J, x, \hat{I}, \hat{J})$. Then $(p, x) \in A$ or $(x, p) \in A$.

We set the independence relation I_Σ to be $\Sigma \times \Sigma - D_\Sigma$. The Mazurkiewicz trace alphabet (Σ, I_Σ) induces the equivalence relation $\approx \subseteq \Sigma^\star \times \Sigma^\star$ given by: Suppose $\sigma e e' \sigma'$, $\sigma e' e \sigma'$ are in $\Sigma^\star$ such that $e\, I_\Sigma\, e'$. Then $\sigma e e' \sigma' \approx \sigma e' e \sigma'$.

As usual, we let $[\sigma]_\approx$ denote the $\approx$-equivalence class containing σ and often drop the subscript $\approx$. Using our definitions and basic Mazurkiewicz trace theory, one can easily establish the following facts.

Proposition 4. (i) *Suppose σ is a firing sequence. Then $[\sigma] \subseteq FS$.*
(ii) *Suppose σ and σ' are firing sequences. Then σ is complete iff σ' is complete.*
(iii) *Suppose $\sigma \approx \sigma'$ and σ is complete. Then the control state sequence induced by σ is the same as the one induced by σ'.*
(iv) *Recall that $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$. Let σ be a complete firing sequence and $\#(\sigma, e_\eta) = h$ for some η with $h > 0$. Then there exists $\hat{\sigma} \in [\sigma]$ such that $\hat{\sigma} = \pi_1 \pi_2 \ldots \pi_h$, where each π_j is of the form $e_{x_1} e_{x_2} \ldots e_{x_n} e_{p_1} e_{p_2} \ldots e_{p_m}$ with $e_\eta \in \Sigma_\eta$ for each η.*

4.5. The main result

We are now ready to prove the main result.

Theorem 5. *Let $\mathcal{H}$, $\mathcal{B}$ be as described above. Then $L(\mathcal{H}) = L(\mathcal{B})$ and $L(\mathcal{H})$ is regular.*

Proof. First, we show that $L(\mathcal{H}) \subseteq L(\mathcal{B})$. Let $\hat{s}_0 \hat{s}_1 \ldots \hat{s}_k \in L(\mathcal{H})$ be a control state sequence induced by the run $\sigma = \xi_0 \xi_1 \ldots \xi_k$ of $\mathcal{H}$. We shall construct a complete firing sequence π of $\mathcal{B}$ such that the control state sequence induced by π is $\hat{s}_0 \hat{s}_1 \ldots \hat{s}_k$.

The proof proceeds by induction on k. The base case $k = 0$ is clear. So assume inductively that there is a complete firing sequence π such that the control state sequence induced by $\sigma = \xi_0 \xi_1 \ldots \xi_k$ is identical to the control state sequence induced by π. Let $z^{in} \in \Lambda^{in}$ and $z \in \Lambda$ be states of $\mathcal{B}$ such that $z^{in} \overset{\pi}{\hookrightarrow} z$, and for each x, the value of x in configuration ξ_0 lies in the interval indicated by the first component of $z^{in}(x)$. Further assume (and this is easily verified for the base case) that in configuration ξ_k the value of each x lies in the interval indicated by the first component of $z(x)$.

Now suppose $\xi_k \Longrightarrow \xi_{k+1}$. In each $\mathcal{B}_x$, we choose the transition that will update the current quantized value of x and the quantized observed value of x using the rates of x prescribed by the current and previous control states of p as obtained from $\mathcal{B}_p$ with $x \in W_p$. Suppose this move takes $\mathcal{B}_x$ from (I, J, β) to $(\hat{I}, \hat{J}, \hat{\beta})$. Then by Lemma 1 and the induction hypothesis, we have that in configuration ξ_{k+1}, the value of x lies in $\hat{I}$. Thus we can extend π via $\pi' = \pi e_{x_1} e_{x_2} \ldots e_{x_n}$ (recall that $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$), such that $e_x \in \Sigma_x$ for each x. And for some suitable $z' \in \Lambda$, we have $z^{in} \overset{\pi'}{\hookrightarrow} z'$. Moreover z' has the property that in configuration ξ_{k+1}, the value of each x lies in the interval indicated by first component of $z'(x)$. It is also easy to show that π' is indeed a firing sequence.

Next we consider $\mathcal{A}_p$ for some p and let s_p,$\hat{s}_p$ be respectively the first components of ξ_k and ξ_{k+1}. Note that in going from configuration ξ_k to ξ_{k+1}, the change of mode of $\mathcal{A}_p$ from s_p to $\hat{s}_p$ (in case $s_p \neq \hat{s}_p$) can be mimicked by a suitable move in $\mathcal{B}_p$. Suppose this move takes $\mathcal{B}_p$ from (s_p, s_p', β) to $(\hat{s}_p, \hat{s}_p', \hat{\beta})$, then again, using the definitions, Lemma 1 and the induction hypothesis one can ensure that the chosen move is such that in configuration ξ_{k+1}, the current control state of p is $\hat{s}_p$ and the previous control state of p is s_p (which is the same as $\hat{s}_p'$). We now extend π' to $\pi'' = \pi' e_{p_1} e_{p_2} \ldots e_{p_m}$ (recall

that $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$), such that $e_p \in \Sigma_p$ for each p. Further we can find a suitable $z'' \in \Lambda$ with $z^{in} \overset{\pi''}{\hookrightarrow} z''$ such that in configuration ξ_{k+1}, the current and previous control states of p agree with the first and second components of $z''(p)$ for every p. Note that in going from z' to z'' in $\mathcal{B}$, none of $\{\mathcal{B}_x\}$ makes a move. Thus, following the property observed on z', one concludes that in configuration ξ_{k+1}, the value of each x lies in the interval indicated by the first component of $z''(x)$. It is now easy to verify that π'' is a complete firing sequence and that the control state sequence it induces is identical to the one induced by $\sigma\xi_{k+1}$.

To show inclusion in the other direction, let σ be a complete firing sequence of $\mathcal{B}$. To start with, assume that $\sigma = e_{x_1} \ldots e_{x_n} e_{p_1} \ldots e_{p_m}$ so that σ is simulating just the first move made by $\mathcal{H}$. The special form of σ we are assuming is justified by Proposition 4.

Suppose further that $z^{in} \overset{\sigma}{\hookrightarrow} z$ where $z^{in} \in \Lambda^{in}$ and $z \in \Lambda$. Then by repeating the arguments developed in the first half of this proof, we can find an initial configuration ξ_0 and a configuration ξ_1 such that the following conditions hold:

- For every $x \in \mathcal{X}$, the value of x in ξ_0 lies in I_x, where $z^{in}(x) = (I_x, I_x, 0)$ for every x.
- For every $x \in \mathcal{X}$, the value of x in ξ_1 lies in $\hat{I}_x$ where $z(x) = (\hat{I}_x, \hat{J}_x, 1)$ for every x.
- For each p, in configuration ξ_1, the current and previous control states of p agree respectively with the first and second components of $z(p)$.

By repeated applications of Proposition 4(iv) we now have the required inclusion.

Lastly, we argue that $L(\mathcal{H})$ is regular. This already follows from results of [7]. Instead, here one can also construct a finite state automaton that runs alongside $\mathcal{B}$ and checks whether the firing sequence that has been generated so far is complete. If the communication graph is connected, then due to the bound K established in Proposition 3, only a finite amount of information will have to be maintained to check this and hence this automaton can be constructed. This establishes that the language of complete firing sequences is regular and hence $L(\mathcal{B})$ is also regular. □

The case where the communication graph is not connected can be dealt with by observing that the behavior of $\mathcal{B}$ can be decomposed into behaviors of automata in each connected component. In particular, every action label arising from a connected component will be independent of every action label belonging to any other connected component. Using this observation and some notational overhead, it is easy to establish Theorem 5 for the case of multiple connected components.

Recall that a DHA is well-behaved iff every reachable configuration of its transition system is feasible. It is easy to see that the global control states of $\mathcal{B}$ can be augmented with the finitely quantized values of the variables. The resulting automaton will still be finite state and effectively computable. This leads to:

Proposition 6. *It is decidable whether a DHA is well-behaved.*

Next let $\mathcal{H}$ be a well-behaved DHA. Then our representation of its discrete time behavior enjoys the following basic property.

Proposition 7. *Every σ in $L(\mathcal{B})$ is the prefix of a complete firing sequence.*

Consequently, for well-behaved DHAs, there is no need to "filter" the behavior of $\mathcal{B}$.

As observed above, the global control states can be augmented with the finitely quantized values of the variables and the resulting language will still be regular. Consequently one can also reason – in terms of intervals of values – about the quantitative behavior of the plant.

4.6. The marked graph representation of communication graphs

A useful fact is that the communication graph $CG = (\Upsilon, A)$ can be viewed as the underlying graph of a marked graph [16]. As the name suggests a marked graph is a directed graph together with an initial marking. The marking denotes a distributed state and is a function which assigns a non-negative integer to each arc. If M is a marking and a is an arc then one says that a carries $M(a)$ tokens at M.

A node can fire at the marking M iff all its input arcs carry at least one token at M. When a node fires, one token is removed from each of its input edges and one token is added to each of its output edges and this will result in a new marking. The reachable markings are the ones that are reached, starting from the initial marking through repeated node firings.

In the present setting both the variable nodes and controller nodes can fire when enabled. Further, the initial marking is the one which places exactly one token on edges of the form (p, x) where $x \in W_p$. All other edges are left unmarked.

By augmenting A with all the complementary arcs (i.e. add the arc (η, η') if (η', η) is in A) and augmenting the initial marking suitably, one can obtain a live and safe marked graph [16]. By "live", we mean that for every node, starting from any reachable marking, we can reach a marking at which the node becomes enabled. By "safe" we mean that at any reachable marking an arc will carry at most one token.

Following these ideas, the connected graph shown in Fig. 3(i) will give rise to the live and safe marked graph shown in Fig. 3(ii). The dotted arcs are the "complement" arcs that have been added to the communication graph. The initial marking is indicated by the tokens placed on some of the arcs. The firing of the node η in this marked graph will correspond to a move of the automaton $\mathcal{B}_\eta$. In this sense, the firing sequences of this marked graph will be an abstraction of the firing sequences of $\mathcal{B}$.

Live and safe marked graphs have a rich and well-understood theory which can be exploited to study the behavior of $\mathcal{B}$. For instance, we note that at the initial marking of the marked graph shown in Fig. 3(ii), every variable node is enabled. Further, if (η, η') is an arc in this marked graph then the firings of η and η' will have to alternate. One can also use the acyclic path carrying a maximal number of tokens, namely the path $px_2qx_3rx_1$ to determine that $K = 3$ for this system where K is the constant asserted in Proposition 3(ii). More importantly, the marked graph representation can be the basis for devising partial order based reduction methods including finite unfoldings [17] to efficiently verify the behavior of $\mathcal{B}$.

5. Extensions

We next discuss how the main result can be extended to more general settings.

5.1. Multi-speed clocks

Theorem 5 goes through even when the controllers are driven by clocks running at different but rationally related frequencies. To bring this out, we associate with each controller $\mathcal{A}_p$ a positive rational number T_p called its period. $\mathcal{A}_p$ receives at each $k \cdot T_p$, the observed values of the variables in R_p and updates – if possible and if it chooses to do so – the rates of variables in W_p. As before we introduce delay parameters $\hat{\delta} = \{\delta^W_{min}, \delta^W_{max}, \delta^R_{min}, \delta^R_{max}\}$ which are rationals in $[0, 1]$ with $\delta^W_{min} \leq \delta^W_{max} \leq \delta^R_{min} \leq \delta^R_{max}$. The interpretation is that at each $k \cdot T_p$, the observed value of each x in R_p received by p is the value of x that held at some time in $[((k-1)+\delta^R_{min}) \cdot T_p,\ ((k-1)+\delta^R_{max}) \cdot T_p]$. And the rate change of x in W_p induced by a mode change of p will kick in at some time in $[(k+\delta^W_{min}) \cdot T_p,\ (k+\delta^W_{max}) \cdot T_p]$. Note that the delay in sensing a variable x is with regard to the period of p for which $x \in R_p$. As before, we have introduced uniform delay parameters only for notational convenience.

Let Δ be the largest rational which integrally divides every rational in $\{c \cdot T_p \mid p \in \mathcal{P},\ c \in \{1\} \cup \hat{\delta}\}$. Then the dynamics of the plant and the controllers can be captured by a transition system in which Δ units of time will pass between successive moves. For each p, one keeps a modulo T_p/Δ counter. This counter will be incremented after each passage of Δ units of time. A transition in $\mathcal{A}_p$ will be taken only when its counter is zero.

For each $x \in R_p$, its observed value is the actual value that held when the value of the counter of p is between $\delta^R_{min} \cdot T_p/\Delta$ and $\delta^R_{max} \cdot T_p/\Delta$. Thus one guesses non-deterministically in which Δ-size interval x has been observed, and calculates the observed value depending on the time instant at which x was measured. The observed value is then kept until it is used in evaluating a guard when the counter of p reaches zero. Similarly, for each $x \in W_p$, its rate change will kick in when the value of the counter of p is between $\delta^W_{min} \cdot T_p/\Delta$ and $\delta^W_{max} \cdot T_p/\Delta$. Thus, one guesses non-deterministically at which Δ-size interval the rate change of x kicks in and updates the actual value of x depending on the time instant in this interval that the rate change kicks in.

Thus a configuration will now track the values held by the T_p/Δ counters too. A transition from one configuration to another will first let Δ units of time pass followed by updating the actual and observed values of each x, increasing the counter value by 1. This will be followed by possible mode transitions provided the corresponding counter values are 0 and the chosen guards are is satisfied. The details are tedious but straightforward and can be easily derived from the transition system semantics presented in Section 3.

As in Section 4, one can construct a network of finite state automata $\{\mathcal{B}_\eta\}_{\eta \in \mathcal{P} \cup \mathcal{X}}$ such that $L(\mathcal{H}) = L(\mathcal{B})$. The key difference is that each move of $\mathcal{B}_\eta$ would correspond to the passage of just Δ time units. The moves of the automata $\{\mathcal{B}_\eta\}$ will be coordinated using a parity bit protocol as in Section 4.

A state of $\mathcal{B}_p$ will keep track of the control states at $k \cdot T_p$, $(k-1) \cdot T_p$, as well as a modulo T_p/Δ counter. The control states kept track by $\mathcal{B}_p$ will be updated only when its counter value of $\mathcal{B}_p$ reaches zero.

A state of $\mathcal{B}_x$ will track the quantized value of x at $k \cdot T_p$ and the quantized observed value reported at $k \cdot T_p$, and also whether it has switched to the rate prescribed by the control state of $\mathcal{B}_p$ at $k \cdot T_p$. Suppose x is controlled by $\mathcal{A}_p$. Then it will read the counter of p. When this counter value lies between $\delta^W_{min} \cdot T_p/\Delta$ and $\delta^W_{max} \cdot T_p/\Delta$, $\mathcal{B}_x$ will non-deterministically switch to the rate prescribed by the control state of $\mathcal{B}_p$ at $k \cdot T_p$. Similarly, if x is sensed by $\mathcal{A}_q$, then by tracking the counter of q, $\mathcal{B}_x$ will update its quantized observed value. Only when the counter of $\mathcal{B}_q$ reaches zero, automaton $\mathcal{B}_q$ will use the quantized observed value of $\mathcal{B}_x$ in evaluating a guard associated with a transition of $\mathcal{A}_q$.

Again the details are straightforward and the proof that the product of this network of automata accepts $L(\mathcal{H})$ can be established as in Section 4.

5.2. Communication between controllers

As observed earlier, there is implicit information flow between the controllers since variable controlled by one controller may be sensed by another one. We can in fact extend the model by specifying explicit point-to-point communication channels between the controllers.

We assume a finite communication alphabet, and allow the controllers to transmit messages via point-to-point channels (p, q) with $p \neq q$. Transitions of $\mathcal{A}_p$ are now augmented with the sending and receiving of a specific message. To capture delays in message transmissions we introduce parameters ϑ_{min}, ϑ_{max} which are positive rationals. The interpretation is that

a message sent at $k \cdot T_p$ by p on the channel (p, q) will reach q at some time in $[k \cdot T_p + \vartheta_{min},\ k \cdot T_p + \vartheta_{max}]$. This message may be consumed by $\mathcal{A}_q$ only at the discrete time instants of the form $k' \cdot T_q$.

We assume that if two messages sent out by p both arrive before $\mathcal{A}_q$ consumes any one of them, then the message which reaches q later will overwrite the one which reaches q earlier. It then follows that at any time, the number of messages in transit from p to q is bounded by $\lceil \vartheta_{max}/T_p \rceil$. As in Section 4 we can construct a network of automata that accepts $L(\mathcal{H})$. The family of automata $\{\mathcal{B}_\eta\}_{\eta \in \mathcal{P} \cup \mathcal{X}}$, will be augmented with one automaton $\mathcal{C}_{pq}$ for each point-to-point channel (p, q). The communication graph will be expanded with nodes of the form (p, q) corresponding to the channels and there will be an arc from p to (p, q) as well as from (p, q) to q. We choose Δ so that it also integrally divides ϑ_{min}, ϑ_{max}. A channel automaton $\mathcal{C}_{pq}$ will not keep track of the exact duration that a message has been in transit, but only in which Δ-size interval this duration lies. Using this expanded network of finite state automata, Theorem 5 can then be established. At present we have not worked out the details for the case where there is bounded buffering of messages at the receiver's side.

5.3. Variables observed by more than one controller

In practice, it is realistic to assume that each variable is observed – with the help of a sensor – by only one controller. If necessary, this information can be communicated to other controllers. However, in theory, one can permit more than multiple controllers to independently observe a variable. Interestingly, our results will still go through but with an additional computational price. In particular, if x is observed by a collection of controllers Q, then the automaton $\mathcal{B}_x$ will also need to track the observed quantized values of x as measured by every $p \in Q$. And these values have to be updated *simultaneously* because they all depend on the time at which the rate change of x kicks in. Thus, in computing the transitions of $\mathcal{B}_x$, one has to enumerate all possible ways of updating simultaneously these quantized values. To do so one must add a single time of rate change and for each p in Q one measurement time. Consequently, the number of states of $\mathcal{B}_x$ as well as the complexity of computing transitions of $\mathcal{B}_x$ will be additionally exponential in the size of $|Q|$.

5.4. Finite precision

We have assumed that variables can be measured with perfect precision and that the guards are rectangular. Following the techniques in [18], our results can be extended to a setting where the variables are assumed be measured with finite precision but the guards are allowed to be polynomial constraints. The key observation is that the finite precision condition allows one to transform polynomial guards (in fact computable guards) into rectangular guards on the *actual* values of variables as detailed in [18].

6. Conclusion

We have studied here a network of hybrid automata capturing the interactions between a plant and its distributed set of controllers. We have shown that the discrete time behavior of this model is not only regular but that it can be succinctly represented as a network of finite state automata. We have also described how our main result can be extended in a number of interesting ways.

The case where the rate of a continuous variable is specified as $\frac{dx}{dt} \in [c, c']$ for rational constants c, c' is worth considering next. We conjecture that our main result will through in this setting but the details need to be worked out. The case where exponential rates are allowed via differential equations of the form $\frac{dx}{dt} = c \cdot x(t)$ is harder. Even if each controller actuates just one variable, it is not clear how to quantize the value space. Hence we are unable to venture a conjecture in this setting.

An important extension from a practical point of view would be to explicitly model the computations carried out by the controllers. In such an extension, for determining the mode transitions, each controller will execute a task to compute the control law based on the values received from the plant as well as internal variables. Further, the controllers will exchange the results of these computations through a shared bus. Due to resource-bounds there will be a complex interplay between the continuous behavior of the plant and the discrete behavior of the controllers. Specifically, the worst case execution times of the tasks and end-to-end delays in the communications between the controllers will impact on controlled behavior of the plant. On the other hand, safety and quality-of-service requirements placed on the plant will require the computational platform to meet stringent performance requirements. It will be particularly interesting to study this interplay between the plant dynamics and the performance of the computational platform in a Time-Triggered Architecture setting [19].

Acknowledgement

The second author's work was supported by Macao FDCT under the PEARL project, grant number 041/2007/A3, and by Institute for Mathematical Sciences of the National University of Singapore.

References

[1] T. Henzinger, The theory of hybrid automata, in: Proc. of 11th LICS, IEEE Press, 1996, pp. 278–292.
[2] T. Henzinger, P. Kopke, A. Puri, P. Varaiya, What's decidable about hybrid automata? J. Comput. System Sci. 57 (1998) 94–124.

[3] T. Henzinger, Hybrid automata with finite bisimulations, in: Proc. of 22nd ICALP, in: LNCS, vol. 944, Springer, 1995, pp. 324–335.
[4] T. Henzinger, P. Kopke, Discrete-time control for rectangular hybrid automata, Theoret. Comput. Sci. 221 (1999) 369–392.
[5] E. Asarin, O. Bournez, T. Dang, O. Maler, Reachability analysis of piecewise-linear dynamical systems, in: Proc. of 3rd HSCC, in: LNCS, vol. 1790, Springer, 2000, pp. 20–31.
[6] R. Alur, T. Henzinger, G. Lafferriere, G. Pappas, Discrete abstractions of hybrid systems, Proc. IEEE 88 (2000) 971–984.
[7] M. Agrawal, P. Thiagarajan, Lazy rectangular hybrid automata, in: Proc. of 7th HSCC, in: LNCS, vol. 2993, Springer, 2004, pp. 1–15.
[8] V. Diekert, G. Rozenberg (Eds.), The Book of Traces, World Scientific Publishing, Singapore, 1995.
[9] O. Clarke, E.M. Grumberg, D. Peled, Model Checking, MIT Press, 1999.
[10] P. Godefroid, Partial-Order Methods for the Verification of Concurrent Systems—An Approach to the State-Explosion Problem, in: LNCS, vol. 1032, Springer, 1996.
[11] M. Donkers, L. Hetel, W. Heemels, N. van de Wouw, M. Steinbuch, Stability analysis of networked control systems using a switched linear systems approach, in: Proc. of HSCC 2009, in: LNCS, vol. 5469, 2009, pp. 150–164.
[12] J. Hespanha, P. Naghshtabrizi, Y. Xu, A survey of recent results in networked control systems, Proc. IEEE 95 (1) (2007) 138–162.
[13] S. Tripakis, Decentralized control of discrete event systems with bounded or unbounded delay communication, IEEE Trans. Automat. Control 49 (9) (2004) 1489–1501.
[14] P.S. Thiagarajan, S. Yang, Succinct discrete time approximations of distributed hybrid automata, in: Proc. of 13th HSCC, ACM Press, 2010, pp. 1–10.
[15] G.B. Dantzig, B.C. Eaves, Fourier–Motzkin elimination and its dual, J. Combin. Theory, Ser. A 14 (3) (1973) 288–297.
[16] F. Commoner, A. Holt, S. Even, A. Pnueli, Marked directed graphs, J. Comput. Syst. Sci. 5 (1971) 511–523.
[17] J. Esparza, K. Heljanko, Unfoldings—A partial-order approach to model checking, in: EATCS Monographs in Theoretical Comp. Sci, Springer, 2008.
[18] M. Agrawal, P. Thiagarajan, The discrete time behaviour of lazy linear hybrid automata, in: Proc. of 8th HSCC, in: LNCS, vol. 3414, Springer, 2005, pp. 55–69.
[19] P. Caspi, A. Curic, A. Maignan, C. Sofronis, S. Tripakis, P. Niebert, From Simulink to SCADE/Lustre to TTA: a layered approach for distributed embedded applications, in: Proc. of LCTES 2003, ACM Press, 2003, pp. 153–162.