

# Succinct Discrete Time Approximations of Distributed Hybrid Automata

P. S. Thiagarajan<sup>\*</sup>  
School of Computing, NUS, Singapore  
thiagu@comp.nus.edu.sg

Shaofa Yang<sup>†</sup>  
UNU-IIST, Macao, China  
ysf@iist.unu.edu

## ABSTRACT

We consider a network of hybrid automata that observe and control a plant whose state space is determined by a finite set of continuous variables. We assume that at any instant, these variables are evolving at (possibly different) constant rates. Each automaton in the network controls—i.e. can switch the rates of—a designated subset of the continuous variables without having to reset their values. These mode changes are determined by the current values of a designated subset of the variables that the automaton can observe. We require the variables controlled—in terms of effecting mode changes—by different hybrid automata to be disjoint. However, the same variable may be observed by more than one automaton.

We study the discrete time behavior of such networks of hybrid automata. We show that the set of global control state sequences displayed by the network is regular. More importantly, we show that one can effectively and succinctly represent this regular language as a product of local finite state automata.

## Categories and Subject Descriptors

F.4.0 [Theory of Computation]: Mathematical Logic and Formal Languages—*General*

## General Terms

Theory, Verification

## Keywords

distributed hybrid automata, discrete time approximation

<sup>\*</sup>Supported by the Ministry of Education Grant T208A2104.

<sup>†</sup>Supported by Macao FDCT under the PEARL project, grant number 041/2007/A3.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HSCC'10, April 12–15, 2010, Stockholm, Sweden.

Copyright 2010 ACM 978-1-60558-955-8/10/04 ...\$10.00.

## 1. INTRODUCTION

We study the discrete time behavior of a network of hybrid automata. The network models the behavior of a set of controllers that interact with a plant in a distributed fashion. We associate a finite set of continuous variables  $\mathcal{X}$  with the plant. Each controller can observe a subset of  $\mathcal{X}$ . Based on these observations, it can actuate—i.e. effect changes to the rates of—a subset of  $\mathcal{X}$ . An important feature is that the mode changes to the continuous variables are effected without requiring the current values of these variables to be reset to a pre-determined range of values. We show that the discrete time behavior of such a network of hybrid automata can be effectively and succinctly represented as a network of finite state automata.

In the present study, there is no explicit communication between the hybrid automata. However, there will be information flow between them due to the state space of the plant acting as a shared memory. Specifically, the automaton  $\mathcal{A}$  will use the values of  $Obs_{\mathcal{A}}$ , the variables it can observe, to guard the transitions affecting mode changes to  $Ctl_{\mathcal{A}}$ , the variables it controls. For a different automaton  $\mathcal{A}'$  in the network,  $Obs_{\mathcal{A}'}$  may intersect with  $Ctl_{\mathcal{A}}$ . This will result in coordination between the behaviors of the two automata. To ensure that at every instant, a variable has a well-defined rate governing its evolution, we require that the variables controlled by the automata are pairwise disjoint. In general,  $Obs_{\mathcal{A}}$  will be different from  $Ctl_{\mathcal{A}}$ . This is so since a controller may be able to sense (actuate) a variable without being able to actuate (sense) it.

Figure 1 displays an example of a plant with three controllers. An arc from the plant to controller  $i$  labelled with  $x$  indicates that  $i$  can observe the value of  $x$ . Thus, the set of variables that controller 2 observes is  $\{x_1, x_3\}$ . An arc from the controller  $i$  labelled with an  $x$  to the plant signifies that  $x$  is controlled by  $i$ .

Though we do not handle it here, there *is* a case to be made for letting the automata communicate with each other externally. In practice they will share a communication platform and exchange the values of the private variables they use to compute the control functions that are being realized. This external computational activity of the controllers and the plant dynamics they control will strongly influence each other. However, as we point out in the concluding section, this is a topic worth an independent study. Hence, we shall focus here on studying just the plant behavior determined by the network of controllers.

We assume that at any instant, each variable is evolving at a constant rate and that the guards associated with the

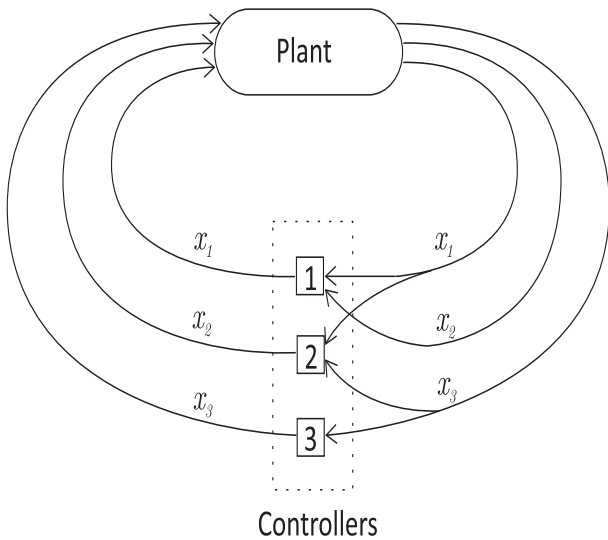


Figure 1: A plant interacting with three controllers

transitions of the hybrid automata are rectangular. Both these restrictions are standard, well-motivated and extensively studied [13, 15, 12, 14]. However, even with these restrictions, the continuous time behavior of a hybrid automaton will often be intractable [4, 15]. One usually imposes two very different types of additional restrictions to ensure tractability. One type of restriction is to require that the values of the continuous variables be reset to fall within a pre-determined range of values whenever there is a mode change affecting the rate of this variable [3]. Clearly, 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 a natural restriction since the controllers—almost always realized as digital devices—will sense and actuate only at discrete time points determined by their internal clocks.

In the presence of the discrete time assumption, one will not require the resetting assumption. This was established in [13] and later extended to much richer settings [1, 2]. The basic result in this setting is that the discrete time behavior of the plant-controller combine is a regular language and one can effectively compute a finite state automaton representing this language. As a result, a variety of verification and controller synthesis problems can be studied and solved using classical methods.

In fact, one can exploit the techniques developed in [1, 2] to show that in the present setting too, the discrete time behavior of the network of hybrid automata is a regular language and that this language can be effectively represented as a finite state automaton. However, the size of this automaton will be exponential in the number of variables and in the number of controllers. Hence this automaton can be prohibitively large.

Our main result is that one can instead represent this language succinctly as a network of finite state automata. The overall size of this representation will be linear in the number of variables and in the number of controllers but exponential in the number of variables that a controller can

observe and control. We however expect this latter number to be small relative to the total number of variables. Naturally, to analyze this representation as a network of finite state automata, one will have to deal with the state explosion problem. However, a variety of techniques have been developed by the formal verification community including symbolic representation methods [6] and partial order verification methods [11] 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 [9]). An informative survey of research on networked control systems for instance is provided in [16]. The main objective in this line of research is to minimize the impact of distribution and communication on the control task being implemented; not 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 with [17] being a representative study. 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 terms of a transition system. In section 4 we establish our main results. We do so in a restricted setting to avoid notational clutter from obscuring the key ideas. In section 5 we then sketch how these restrictions can be relaxed and our results can be extended in various ways. In the concluding section we summarize and briefly discuss the prospects for future research.

## 2. A CLASS OF DISTRIBUTED HYBRID AUTOMATA

We associate  $n$  continuous variables  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$  as well as a set of *locations* (*names*)  $\mathcal{P} = \{p_1, p_2, \dots, p_m\}$  with the plant. We let  $p, q$  range over  $\mathcal{P}$ . We will assign one controller—modeled as a hybrid automaton  $\mathcal{A}_p$ —to each location  $p$ .

Each automaton  $\mathcal{A}_p$  can observe a subset of  $\mathcal{X}$  denoted  $Obs_p$  and control a subset of  $\mathcal{X}$  denoted  $Ctl_p$ . By “control” we mean that at suitable times, it can effect mode changes to the variables in  $Ctl_p$ . We require  $Ctl_p \cap Ctl_q = \emptyset$  if  $p \neq q$  and  $\bigcup_p Ctl_p = \mathcal{X}$ . To highlight the main ideas and lighten the notation, we will first formulate our model with the restriction  $|Ctl_p| = 1$  for every  $p$ . In other words, each automaton will control exactly one continuous variable and we let  $x_p$  denote the variable controlled by the automaton  $\mathcal{A}_p$ .

$\mathbb{R}$  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 which the variables can attain lie within the interval  $[B_{min}, B_{max}]$ . By convention, the plant will get stuck whenever the value of some continuous variable goes outside this interval. For convenience we have assumed uniform lower and upper bounds for the values of the variables.

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  $x \in X$ . Let  $Grd(X)$  denote the collection of guards over  $X$ . By an  $X$ -valuation, we shall mean a mapping from  $X$  to  $\mathbb{R}$ . The notion of an  $X$ -valuation satisfying an  $X$ -guard is defined in the obvious way.

A *Distributed Hybrid Automaton* (“dha” for short) is a  $\mathcal{P}$ -indexed family of hybrid automata  $\{\mathcal{A}_p\}_{p \in \mathcal{P}}$ , where each  $\mathcal{A}_p = (S_p, s_p^{in}, Obs_p, x_p, \rightarrow_p, Init_p, \rho_p)$  is as follows:

- $S_p$  is a finite set of *control states*.
- $s_p^{in} \in S_p$  is the initial control state.
- $Obs_p \subseteq \mathcal{X}$  is the set of variables observed by  $p$  and  $x_p$  is the variable controlled by  $p$ . For convenience, we set  $Var_p = Obs_p \cup \{x_p\}$ .
- $\rightarrow_p \subseteq S_p \times Grd(Obs_p) \times S_p$  is the transition relation such that if  $(s_p, \varphi, s'_p) \in \rightarrow_p$  then  $s_p \neq s'_p$ .
- $Init_p$  will assign an interval  $[d_{min}^x, d_{max}^x]$  of initial values to each variable being observed or controlled by  $p$ . We require both  $d_{min}^x$  and  $d_{max}^x$  to be rational numbers such that  $B_{min} \leq d_{min}^x \leq d_{max}^x \leq B_{max}$ . We further require that if  $x \in Var_p \cap Var_q$  then  $Init_p(x) = Init_q(x)$ .
- $\rho_p : S_p \rightarrow \mathbb{Q}$  where  $\rho_p(s)$  is the rate of evolution of  $x_p$  when  $\mathcal{A}_p$  resides in the control state  $s$ .

Figure 2 displays a dha consisting of three automata  $\mathcal{A}_{p1}$ ,  $\mathcal{A}_{p2}$ ,  $\mathcal{A}_{p3}$  following the usual graphical notations. To reduce clutter, we have not shown all the guards.

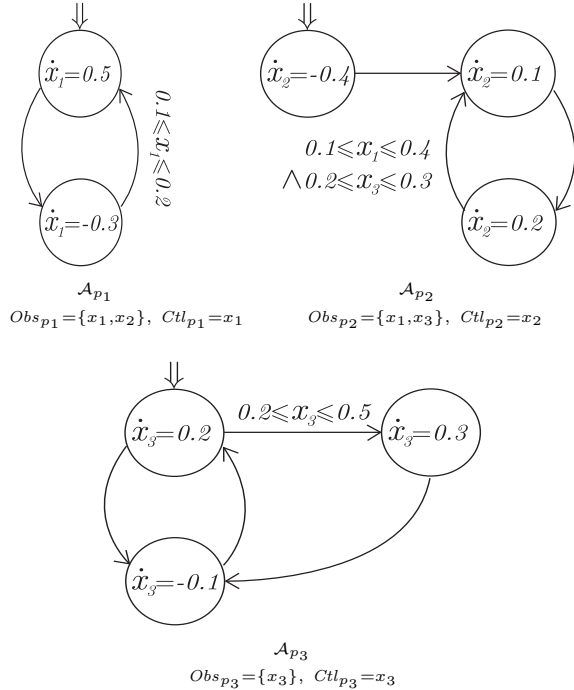


Figure 2: A dha

We shall study the discrete time behavior of this model and accordingly fix a unit of time at a suitable level of granularity. At time  $t_0 = 0$ , each  $\mathcal{A}_p$  will be in its initial control state and each variable  $x$  will have its initial value lying in the interval  $[d_{min}^x, d_{max}^x]$ . Each variable  $x_p \in \mathcal{X}$  will start evolving at rate  $\rho_p(s_p^{in})$ .

Suppose at time  $t_k$  the automaton  $\mathcal{A}_p$  is in the control state  $s_p$  and the valuation  $V_k$  specifies the current value of each variable and each  $x_p$  is evolving at rate  $\rho_p(s_p)$ . Then one unit of time will pass at the end of which the following actions occur instantaneously.

- The plant will transmit the current value of the variable  $x$  at time  $t_{k+1}$  to each  $\mathcal{A}_p$  for which  $x \in Obs_p$ . Each automaton  $\mathcal{A}_p$  will then do the following.
- Based on the received values of the variables in  $Obs_p$ ,  $\mathcal{A}_p$  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  $\mathcal{A}_p$  will non-deterministically choose one of the enabled transitions and move to a new control state, say,  $s'_p$ . In this case, it will transmit the new rate, namely  $\rho_p(s'_p)$  to the plant. As a result, the variable  $x_p$  will now start to evolve at this transmitted rate starting from  $t_{k+1}$ .
- $\mathcal{A}_p$  will stay put in its current control state in case no outgoing transition is enabled. In this case, no mode change will be effected and  $x_p$  will continue to evolve at the rate  $\rho_p(s_p)$  starting from  $t_{k+1}$ .

One could permit  $\mathcal{A}_p$  to stay in its current mode even if an outgoing transition is 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  $x \in Obs_p$  and  $c, c' \in \mathbb{Q}$ . We would then demand that at  $t_{k+1}$ , the automaton  $\mathcal{A}_p$  may choose to stay in  $s_p$  provided the state invariant associated with  $s_p$  is not violated according to the new values of the variables in  $Obs_p$  that have been received from the plant at  $t_{k+1}$ . Our results will easily go through in the presence of such invariants.

It will be more realistic to assume that the controllers and the plant have different units of time according to which they react. There could also be delays associated with the transmission of messages between the plant and the controllers. However these complications can be factored in without affecting the main results. Hence, in what follows, we will assume a particularly simple and rigid style of operation for the network of the plant and the controllers. In section 5, we will describe how various extensions can be incorporated into the model.

### 3. THE TRANSITION SYSTEM SEMANTICS

Through this section, we fix a distributed hybrid automaton *DHA* as described in the previous section with the associated notations. We note that in the way we have set up the model, for each  $x \in \mathcal{X}$ , there exists a unique  $p$  such that  $x = x_p$ . This fact will be implicitly used in what follows.

We shall define the discrete time dynamics of *DHA* in terms of a transition system  $TS_{DHA}$ , while often dropping the subscript *DHA*. States of *TS* will be termed *configurations*. A configuration is a map  $\xi$  which assigns a control state  $\xi(p) \in S_p$  to each  $p \in \mathcal{P}$  and a value  $\xi(x) \in \mathbb{R}$  to each variable in  $\mathcal{X}$ . Let  $Conf$  denote the collection of configurations. The set of *initial* configurations  $Conf^{in}$  is given by:  $\xi$  is in  $Conf^{in}$  iff  $\xi(p) = s_p^{in}$  and  $\xi(x_p) \in Init_p(x_p)$

for each  $p$ . A configuration  $\xi$  is *feasible* if for every  $x$ ,  $B_{min} \leq \xi(x) \leq B_{max}$ . Clearly, every initial configuration is feasible.

We define the transition relation  $\Longrightarrow \subseteq Conf \times Conf$  via:  $\xi \Longrightarrow \xi'$  iff the following conditions hold.

- $\xi$  is feasible.
- For each  $p$ ,  $\xi'(x_p) = \xi(x_p) + \rho_p(s_p)$ , where  $s_p = \xi(p)$ .
- For each  $p$ , let  $s_p = \xi(p)$ ,  $s'_p = \xi'(p)$ . If  $s_p \neq s'_p$ , then there exists a transition  $(s_p, \varphi, s'_p) \in \longrightarrow_p$  such that the  $Obs_p$ -valuation  $V'$  given by  $V'(x) = \xi'(x)$  for each  $x \in Obs_p$ , satisfies  $\varphi$ .

On the other hand, if there exists no transition  $(s_p, \varphi, s'_p)$  in  $\longrightarrow_p$  such that the  $Obs_p$ -valuation  $V'$  given by  $V'(x) = \xi'(x)$  for each  $x \in Obs_p$ , satisfies  $\varphi$  then  $s'_p = s_p$ .

Now we define the transition system  $TS$  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  $p$ .

A *run* of  $DHA$  is a finite sequence of configurations  $\xi_0 \xi_1 \dots \xi_k$  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  $\bigcup S_p$  such that  $s(p) \in S_p$  for each  $p$ . The global control state induced by the configuration  $\xi$  is denoted as  $st(\xi)$ . It is the global control state  $s$  satisfying  $s(p) = \xi(p)$  for each  $p$ .

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

Based on the results in [1], it is easy to show that  $L(DHA)$  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 is that this state-explosion problem in the *representation* of  $L(DHA)$  can be avoided. We shall show that  $L(DHA)$  can be succinctly represented as the language accepted by the product of a family of finite state automata. Further, 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  $|\mathcal{P}|$ .

It is worth noting that 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  $DHA$ .

## 4. THE REPRESENTATION RESULT

In this section, we establish our main result, namely, that the behavior of a dha can be represented succinctly as the

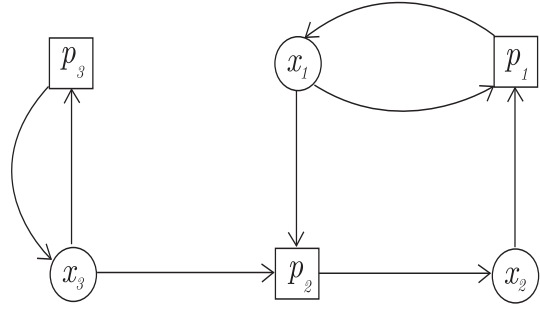


Figure 3: Communication graph

product (parallel composition) of a family of finite state automata that operate asynchronously. We fix a distributed hybrid automaton  $DHA$  as described above with the associated notations. The corresponding network of finite state automata we construct will communicate with each other in the manner of asynchronous cellular automata ([8]). However, no prior knowledge of asynchronous cellular automata will be assumed or needed.

In the rest of this section we let  $\Upsilon = \mathcal{P} \cup \mathcal{X}$  and let  $\eta, \eta'$  range over  $\Upsilon$ . There will be one finite state automaton  $\mathcal{B}_p$  for each  $p$  and one automaton  $\mathcal{B}_x$  for each  $x$ . To show that this family  $\{\mathcal{B}_\eta\}$  represents  $L(DHA)$ , we will define  $\mathcal{B}$ , the product of  $\{\mathcal{B}_\eta\}$  in the standard way.  $\mathcal{B}$  will contain a richer set of behaviors than what is needed. Hence we will extract from  $\mathcal{B}$ —through a simple operation—a finite state automaton that will accept  $L(DHA)$ .

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  $Var_p = Obs_p \cup \{x_p\}$ ). On the other hand, the moves of the automaton  $\mathcal{B}_x$  will depend on its current state and on the current states of the automata in  $\{\mathcal{B}_p\}_{p \in loc(x)}$  where  $loc(x) = \{p \mid x \in Var_p\}$ .

The flow of information between the automata in  $\{\mathcal{B}_\eta\}$  can be conveniently represented in terms of the *communication graph* of  $DHA$  denoted  $CG_{DHA}$ . As before, we will often drop the subscript  $DHA$ . Then  $CG = (\Upsilon, A)$  where  $A = \{(p, x) \mid x = x_p\} \cup \{(x, p) \mid x \in Obs_p\}$ . Thus  $CG$  will be bipartite with arcs going from variable nodes to location nodes and from location nodes to variable nodes.

Figure 3 displays the communication graph of the dha in figure 2. We have used circles to denote the variables and boxes to denote locations merely to emphasize the bipartite nature of the communication graph. No association with Petri nets is intended at this stage.

$\mathcal{B}_p$  will track the current state of  $\mathcal{A}_p$  and thus the current rate of  $x_p$ . The automaton  $\mathcal{B}_x$  will track the current value of  $x$ . Since  $\mathcal{B}_x$  is required to be finite state, it can keep only a bounded amount of information about the current value of  $x$ . To facilitate this, we will quantize the feasible range of values  $[B_{min}, B_{max}]$  into finitely many sub-intervals and represent a value of a continuous variable by the sub-interval it lies in. It will turn out that the evolution of values of variables in  $\mathcal{X}$  can be represented in terms of these sub-intervals.

### 4.1 The quantization of the value space

We begin with the quantization of  $[B_{min}, B_{max}]$ . Let  $\Omega$  be the least set of rational numbers given by:

- $B_{min}, B_{max} \in \Omega$ .

- Suppose  $Init_p(x) = [d_{min}^x, d_{max}^x]$ . Then  $d_{min}^x, d_{max}^x \in \Omega$ .
- $\rho_p(s) \in \Omega$  for every  $p$  and every  $s \in S_p$ .
- Suppose  $(s_p, \varphi, s'_p) \in \longrightarrow_p$  and  $c$  appears in  $\varphi$ . Then  $c \in \Omega$ .

Now let  $\Gamma$  be the largest rational which integrally divides every number in  $\Omega$ . 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  $\mathbb{R}$ , the set of real numbers into finitely many intervals  $(-\infty, N_{min} \cdot \Gamma), [N_{min} \cdot \Gamma, N_{min} \cdot \Gamma], (N_{min} \cdot \Gamma, (N_{min} + 1) \cdot \Gamma), [(N_{min} + 1) \cdot \Gamma, (N_{min} + 1) \cdot \Gamma], \dots, ((N_{max} - 1) \cdot \Gamma, N_{max} \cdot \Gamma), [N_{max} \cdot \Gamma, N_{max} \cdot \Gamma], (N_{max} \cdot \Gamma, \infty)$ .

Let  $INT$  be the collection of these open intervals and closed singleton intervals. Clearly,  $INT$  is a finite set. Now we define the map  $\|\cdot\| : \mathbb{R} \rightarrow INT$  via:  $\|v\| = I$  iff  $v \in I$ .

Next suppose  $X \subseteq \mathcal{X}$  and  $V$  is an  $X$ -valuation. Then  $\|V\|$  is the map  $\|V\| : X \rightarrow INT$  given by:  $\|V\|(x) = \|V(x)\|$  for each  $x$ .

We note some simple facts which will be used to show that the above quantization is sufficiently fine to capture the dynamics of  $DHA$ .

LEMMA 1. *The following assertions hold:*

- (i) *Suppose  $X \subseteq \mathcal{X}$ . Let  $\varphi$  be an  $X$ -guard and  $V, V'$  be  $X$ -valuations such that  $\|V\| = \|V'\|$ . Then  $V$  satisfies  $\varphi$  iff  $V'$  satisfies  $\varphi$ .*

*Further, given a collection of  $X$ -indexed intervals  $\{I_x\}_{x \in X}$  where  $I_x \in INT$ , one can effectively check whether there exists an  $X$ -valuation  $U$  such that  $U(x) \in I_x$  for each  $x$  and  $U$  satisfies  $\varphi$ .*

- (ii) *Let  $v, v' \in \mathbb{R}$  with  $\|v\| = \|v'\|$ . Then  $v \in [B_{min}, B_{max}]$  iff  $v' \in [B_{min}, B_{max}]$ . Further, if  $v \in [B_{min}, B_{max}]$  and  $c = \rho_p(s)$  for some  $p$  and some  $s \in S_p$ , then  $\|v + c\| = \|v' + c\|$ .*

PROOF. First, we prove (i). Let  $\varphi = \bigwedge_{x \in X} c_x \leq x \leq c'_x$ . We fix an  $x \in X$ . Let  $\|V\|(x) = I_x$  and suppose  $I_x = (j \cdot \Gamma, (j + 1) \cdot \Gamma)$  with  $N_{min} \leq j \leq N_{max} - 1$ . Since  $c_x, c'_x$  are multiples of  $\Gamma$ , it follows that  $c_x \leq V(x) \leq c'_x$  (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  $V$  satisfies  $\varphi$  iff  $V'$  satisfies  $\varphi$ , and the second part of (i) also holds.

Similarly, (ii) can be established by considering cases according to whether  $\|v\|$  is an open interval or a singleton interval.  $\square$

## 4.2 The construction of the family of local automata

We now turn to the construction of the collection of automata  $\{\mathcal{B}_\eta\}$ . We first recall that a move in  $DHA$  at  $t_{k+1}$  consists of (i) the value of each  $x$  being updated—from the previous value at  $t_k$ —according to its rate of evolution at time  $t_k$  (ii) the controller  $p$  reading the updated values of each  $x \in Obs_p$ , determining if there is to be mode change and if so, a new control state and rate of evolution of  $x_p$ . Such a move in  $DHA$  will be simulated by the automata in  $\{\mathcal{B}_\eta\}$  in two stages.

For explaining this and for later use it will be convenient to define the notion of the *neighbors* of a node  $\eta$ , denoted

$Nbr(\eta)$ , in the communication graph  $CG = (\Upsilon, A)$ . It is given by,  $Nbr(\eta) = \{\eta' \mid (\eta', \eta) \in A\} \cup \{\eta'' \mid (\eta, \eta'') \in A\}$ .

Clearly there is some redundancy between the notions  $Var_p$ ,  $loc(x)$  and  $Nbr(\eta)$ . We have introduced this redundancy for convenience.

Each automaton  $\mathcal{B}_x$  will read—but not alter—the current states of automata  $\{\mathcal{B}_q\}_{q \in Nbr(x)}$ . Using this information, it will update the quantized value of  $x$  using its current quantized value and the rate of its evolution prescribed by the current state of  $\mathcal{B}_p$  where  $x_p = x$ . Thus the information it reads from the other automata in  $\{\mathcal{B}_q\}_{q \in Nbr(x)}$  is only for proper coordination as will become evident below.

Each  $\mathcal{B}_p$  will read—but not alter—the states of the automata  $\{\mathcal{B}_x\}_{x \in Nbr(p)}$  to obtain the updated quantized values of the variables  $x \in Obs_p$  and determine the new control state. Again, the state of the automaton  $\mathcal{B}_{x_p}$  is read (in case  $x_p \notin Obs_p$ ) only for proper coordination.

To coordinate the moves of automata in  $\{\mathcal{B}_\eta\}$ , each state of  $\mathcal{B}_x$  and  $\mathcal{B}_p$  will also maintain a parity bit. Initially, every automaton will be in its initial state with parity 0. We will require that  $\mathcal{B}_x$  can make a move only when its parity is the same as that of  $\mathcal{B}_p$  for every  $p \in Nbr(x)$ . And  $\mathcal{B}_x$  will flip its parity whenever it makes a move. On the other hand,  $\mathcal{B}_p$  will be allowed to 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. As mentioned earlier, the moves of the automata will be made asynchronously. Thus different automata in  $\{\mathcal{B}_\eta\}$  may have made different number of moves at any given time and hence may have different views of how much global time has passed. We will however show that the automata that belong to the same connected component of the communication graph of  $DHA$  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. As a consequence, every automaton that lies in the same connected component of the communication graph will get stuck within a bounded amount of time.

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

A state of  $\mathcal{B}_p$  will be of the form  $(s_p, \beta)$  where  $s_p \in S_p$  and  $\beta \in \{0, 1\}$  is the parity bit. We let  $W_p$  be the set of states of  $\mathcal{B}_p$ . Thus  $W_p = S_p \times \{0, 1\}$ . The set of initial states of  $\mathcal{B}_p$  denoted  $W_p^{in}$  is a singleton set and consists of  $(s_p^{in}, 0)$ .

Clearly  $W_\eta$  is a finite set for every  $\eta$ .

To define the transition relations, we will make use of the notion of  $Q$ -states relative to the family of sets of states  $\{W_\eta\}$ . Suppose  $Q \subseteq \Upsilon$ . Then a  $Q$ -state is a map which assigns to every element  $\eta$  in  $Q$  a state in  $W_\eta$ . We let  $W_Q$  be the set of  $Q$ -states. In case  $Q = \{\eta\}$  is a singleton, we will say  $\eta$ -state instead of  $\{\eta\}$ -state.

The transition relation of each  $\mathcal{B}_x$ , denoted  $\rightsquigarrow_x$ , is a subset of  $W_x \times W_{Nbr(x)} \times W_x$  defined as follows.

Let  $w = (I, \beta)$  and  $w' = (I', \beta')$  be  $x$ -states and  $z$  be a  $Nbr(x)$ -state with  $z(p) = (s_p, \beta_p)$  for every  $p \in Nbr(x)$ . Then  $(w, z, w') \in \rightsquigarrow_x$  iff the following conditions are satisfied:

- $\beta = \beta_p$  for each  $p$  and  $\beta' = 1 - \beta$ .
- $I$  is contained in  $[B_{min}, B_{max}]$ .
- There exists  $v \in I$  such that  $v+c \in I'$  where  $c = \rho_q(s_q)$  and  $x_q = x$ .

Following lemma 1, the above transition relation is well-defined in the sense that the last condition does not depend on the specific choice of  $v$ . Further,  $\rightsquigarrow_p$  can be effectively computed.

The first condition in the definition of  $\rightsquigarrow_p$  ensures that  $\mathcal{B}_x$  can make a move only when its parity is the same as that of each  $\mathcal{B}_p$  with  $p \in Nbr(x)$ . Further,  $\mathcal{B}_x$  flips its parity at the end of the move. The second condition dictates the current quantized value of  $x$  is within the value range. The last condition ensures that the quantized value of  $x$  is updated according to the current rate of  $\mathcal{B}_q$  with  $x = x_q$ .

The transition relation  $\rightsquigarrow_p$  of  $\mathcal{B}_p$ , is the subset of  $W_p \times W_{Nbr(p)} \times W_p$  defined as follows.

Let  $w_p = (s, \beta)$  and  $w'_p = (s', \beta')$  be  $p$ -states and  $z$  be a  $Nbr(p)$ -state. Assume  $z(x) = (I_x, \beta_x)$  for every  $x$  in  $Nbr(p)$ . Then  $(w_p, z, w'_p) \in \rightsquigarrow_p$  iff the following conditions are satisfied:

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

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  $\{\mathcal{B}_\eta\}$ .

### 4.3 The product of the family of local automata

We now wish to define the product (parallel composition) of  $\{\mathcal{B}_\eta\}$ . The resulting finite state automaton will be denoted as  $\mathcal{B}$ . It is important to note that  $\mathcal{B}$  is being constructed only to establish that  $\{\mathcal{B}_\eta\}$  captures the behavior  $L(DHA)$ . However, verification problems concerning  $L(DHA)$  are intended to be addressed in terms of  $\{\mathcal{B}_\eta\}$ .

In each transition of  $\mathcal{B}$ , one component  $\eta$  will make a move while all other components stay put. Anticipating later developments, it will be convenient to associate action labels with the transitions of  $\mathcal{B}$ . Accordingly, we define  $\Sigma_x = INT \times \{x\} \times INT$  for each  $x$ . A letter  $(I, x, I')$  in  $\Sigma_x$  will be used to label a transition of  $\mathcal{B}$  in which  $\mathcal{B}_x$  makes a move and changes its interval from  $I$  to  $I'$ . For each  $p$ , we define  $\Sigma_p = S_p \times \{p\} \times S_p$ . The letter  $(s, p, s') \in \Sigma_p$  will be used to record a move of  $\mathcal{B}$  in which  $\mathcal{B}_p$  moves from control state  $s$  to  $s'$ . We set  $\Sigma = \bigcup_{\eta \in \Upsilon} \Sigma_\eta$  and let  $e, e'$  range over  $\Sigma$ .

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

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

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

Thus  $\mathcal{B}$  is a finite state automaton which captures the global asynchronous interleaved behavior of  $\{\mathcal{B}_\eta\}$ . The reader familiar with asynchronous cellular automata [8] will notice the similarity in terms of the way in which the components of  $\{\mathcal{B}_\eta\}$  interact with each other.

Our goal is to show that  $\mathcal{B}$  captures in a natural way  $L(DHA)$ , the control state sequence language of  $DHA$ . To start with, we will assume that the communication graph of  $DHA$  is connected (but not necessarily strongly connected) in the usual graph-theoretic sense. Later we will argue how the main result can be lifted to the general case.

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

$FS_{\mathcal{B}} \subseteq \Sigma^*$  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 as follows. 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  $w \xrightarrow{e} u$  instead of  $(w, e, u) \in \hookrightarrow$ .

- The null sequence  $\epsilon$  is in  $FS$ . And  $w \xrightarrow{\epsilon} w$  for each  $w \in W^{in}$ .
- Suppose  $\sigma \in FS$  and  $w^{in} \xrightarrow{\sigma} u$  where  $w^{in} \in W^{in}$ . If  $u \xrightarrow{e} z$ . Then  $\sigma e \in FS$  and  $w^{in} \xrightarrow{\sigma e} z$ .

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

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

Using the definitions of  $\{\mathcal{B}_\eta\}$  and  $\mathcal{B}$  it is tedious but straightforward to establish the following result:

**PROPOSITION 2.** (i) Let  $\sigma$  be a firing sequence and  $x \in Nbr(p)$ . Then  $\#(\sigma, p) \leq \#(\sigma, x) \leq 1 + \#(\sigma, p)$ .

(ii) There exists a non-negative integer  $K$  which depends only on the communication graph  $CG$  such that for every firing sequence  $\sigma$  and every  $\eta, \eta'$  in  $\Upsilon$ ,  $\#(\sigma, \eta) - \#(\sigma, \eta') \leq K$ .

The second part of the proposition needs the assumption that the communication graph is connected. The proof of the proposition merely 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. As we will explain in section 4.5, one can also augment the communication graph suitably to obtain a live and safe marked graph [7] and use basic results of marked graphs to establish proposition 2. Note that the last part of the proposition bounds the amount by which the automata in  $\{\mathcal{B}_\eta\}$  can get away from each other in terms of the number of moves they can make.

We can now extract a control state sequence from a complete firing sequence. To this end, let  $\sigma$  be a complete firing sequence and  $m = \#(\sigma, \eta)$  for some  $\eta$ . By the definition of a complete firing sequence,  $m$  does not depend on the choice of  $\eta$ . We now define  $s_0 s_1 \dots s_m$  to be the control state sequence induced by  $\sigma$  where the global control states  $s_0, s_1, \dots, s_m$  are obtained as follows.

For each  $p$  and for each  $j \in \{0, 1, \dots, m\}$  fix a prefix  $\tau_j^p$  of  $\sigma$  such that  $\#(\tau_j^p, p) = j$ . Let  $z_j^{in} \xrightarrow{\tau_j^p} z_j^p$  for each sequence where  $z_j^{in} \in W^{in}$ . Then for  $0 \leq j \leq m$ ,  $s_j$  is the global control state given by:  $s_j(p) = z_j^p(p)$ .

According to our definition there is a great deal of choice when it comes to fixing the prefixes  $\tau_j^p$ . 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  $L(DHA) = L(\mathcal{B})$ .

For proving the main result, it will be convenient to introduce a Mazurkiewicz trace alphabet ([8]) and group firing sequences into equivalence classes induced by the Mazurkiewicz trace alphabet. A Mazurkiewicz trace alphabet is a pair  $(\Theta, I_\Theta)$  where  $\Theta$  is a finite alphabet and  $I_\Theta \subseteq \Theta \times \Theta$  is an irreflexive and symmetric relation. We call  $I_\Theta$  the independence relation. The relation  $D_\Theta = \Theta \times \Theta - I_\Theta$  is called the dependence relation. Clearly,  $D_\Theta$  is reflexive and symmetric.

Recall  $CG_{DHA} = (\Upsilon, A)$ . We first observe that there is a natural dependence relation  $D_\Sigma \subseteq \Sigma \times \Sigma$  by:  $e1 \ D_\Sigma \ e2$  if one of the following holds:

- (i)  $e1 = e2$ .
- (ii) Let  $e1 = (I, x, I')$ ,  $e2 = (s, p, s')$ . Then  $(x, p) \in A$  or  $(p, x) \in A$ .
- (iii) Let  $e1 = (s, p, s')$ ,  $e2 = (I, x, I')$ . Then  $(p, x) \in A$  or  $(x, p) \in A$ .

We set the independence relation  $I_\Sigma$  to be  $\Sigma \times \Sigma - D_\Sigma$ . Thus  $(\Sigma, I_\Sigma)$  is a Mazurkiewicz trace alphabet which induces the equivalence relation  $\approx \subseteq \Sigma^* \times \Sigma^*$ , where  $\approx$  is the least equivalence relation satisfying:

Suppose  $\sigma e e' \sigma'$ ,  $\sigma e' e \sigma'$  are in  $\Sigma^*$  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  $\sigma$  and often drop the subscript  $\approx$ . Again, using our definitions and basic Mazurkiewicz trace theory, one can easily establish the following facts.

**PROPOSITION 3.** (i) Suppose  $\sigma$  is a firing sequence. Then  $[\sigma] \subseteq FS$ .

(ii) Suppose  $\sigma$  and  $\sigma'$  are firing sequences and  $\sigma \approx \sigma'$ . Then the control state sequence induced by  $\sigma$  is the

same as that induced by  $\sigma'$ . Further,  $\sigma$  is complete iff  $\sigma'$  is complete.

(iii) Let  $\sigma$  be a complete firing sequence and  $\#(\sigma, e_\eta) = m$  for some  $\eta$  with  $m > 0$ . Recall that  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ ,  $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ . Then there exists  $\hat{\sigma} \in [\sigma]$  such that  $\hat{\sigma}$  can be expressed as  $\hat{\sigma} = \tau_1 \tau_2 \dots \tau_m$ , where each  $\tau_j$  is of the form  $e_{x_1} e_{x_2} \dots e_{x_n} e_{p_1} e_{p_2} \dots e_{p_n}$  with  $e_\eta \in \Sigma_\eta$  for each  $\eta$ .

## 4.4 The main result

At last, we can prove our main result.

**THEOREM 4.** Let  $DHA$ ,  $\mathcal{B}$  be as described above. Then  $L(DHA) = L(\mathcal{B})$  and  $L(DHA)$  is regular.

**PROOF.** First, we show that  $L(DHA) \subseteq L(\mathcal{B})$ . Let  $\hat{s}_0 \hat{s}_1 \dots \hat{s}_k \in L(DHA)$  be a control state sequence induced by the run  $\sigma = \xi_0 \xi_1 \dots \xi_k$  of  $DHA$ . We shall construct a complete firing sequence  $\tau$  of  $\mathcal{B}$  which simulates the run  $\sigma$ . The control state sequence induced by  $\tau$  will be  $\hat{s}_0 \hat{s}_1 \dots \hat{s}_k$ .

The proof proceeds by induction on  $k$ . The base case  $k = 0$  is clear. Note that for each  $x$  we also have that  $\|\xi_0(x)\|$  is equal to the first component of  $w^{in}(x)$  for some  $w^{in} \in W^{in}$ , where for  $\tau = \epsilon$ , we also have  $w^{in} \xrightarrow{\epsilon} w^{in}$ . So assume inductively that there is a complete firing sequence  $\tau$  such that the control state sequence induced by  $\sigma = \xi_0 \xi_1 \dots \xi_k$  is identical to the control state sequence induced by  $\tau$  and that  $\|\xi_k(x)\|$  is equal to the first component of  $w(x)$  for every  $x$  where  $w^{in} \xrightarrow{\tau} w$  for some  $w^{in} \in W^{in}$ . Now suppose  $\xi_k \implies \xi_{k+1}$ . In each  $\mathcal{B}_x$ , we choose the transition that will update the current quantized value of  $x$  using the current rate of  $x$  obtained from  $\mathcal{B}_p$  with  $x = x_p$ . Suppose this move takes  $\mathcal{B}_x$  from  $(I, \beta)$  to  $(I', \beta')$ . Then by lemma 1 and the induction hypothesis, we will have  $\|\xi_{k+1}(x)\| = I'$ . Thus we can extend  $\tau$  via  $\tau' = \tau e_{x_1} e_{x_2} \dots e_{x_n}$  (recall that  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ ), such that  $e_x \in \Sigma_x$  for each  $x$ . And for some suitable  $u$ , we have  $w^{in} \xrightarrow{\tau'} u$  with the property that the first component of  $u(x)$  will agree with  $\|\xi_{k+1}(x)\|$  for every  $x$ . It is also easy to show that  $\tau'$  is indeed a firing sequence.

Next we consider  $\mathcal{B}_p$  for some  $p$  and note that the change from  $\xi_k(p)$  to  $\xi_{k+1}(p)$ , if any, can be mimicked by a suitable move in  $\mathcal{B}_p$ . Suppose this move takes  $\mathcal{B}_p$  from  $(s, \beta)$  to  $(s', \beta')$ , then again, using the definitions, lemma 1 and the induction hypothesis one can ensure that the chosen move is such that  $\xi_{k+1}(p) = s'$ . We now extend  $\tau'$  to  $\tau'' = \tau' e_{p_1} e_{p_2} \dots e_{p_n}$  (recall that  $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ ), such that  $e_p \in \Sigma_p$  for each  $p$ . Further we can find a suitable  $z \in W$  such that  $w^{in} \xrightarrow{\tau''} z$  such that  $\xi_{k+1}(p)$  agrees with the first component of  $z(p)$  for every  $p$ . It is now easy to verify that  $\tau''$  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  $\sigma$  be a complete firing sequence of  $\mathcal{B}$ . To start with, in view of proposition 3, assume that  $\sigma$  is of the form  $\sigma = e_{x_1} \dots e_{x_n} e_{p_1} \dots e_{p_n}$  such that  $e_\eta \in \Sigma_\eta$  for each  $\eta$ . Assume further that  $w^{in} \xrightarrow{\sigma} w$  where  $w^{in} \in W^{in}$  and  $w \in W$ . Then by repeating the arguments developed in the first half of this proof, we can find an initial configuration  $\xi_0$  and a configuration  $\xi_1$  such that the following conditions hold:

- $\|\xi_0(x)\| = I_0^x$  for every  $x$  where  $w^{in}(x) = (I_0^x, 0)$  for every  $x$ .
- $\|\xi_1(x)\| = I_1^x$  for every  $x$  where  $w(x) = (I_1^x, 1)$  for every  $x$ .
- For each  $p$ , the first component of  $w(p)$  agrees with  $\xi_1(p)$ .

By repeated applications of the last part of the previous proposition we now have the required inclusion.  $\square$

Lastly, we argue that  $L(DHA)$  is regular. This already follows from results of [1]. 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. Due to the bound  $K$  established in proposition 2, only a finite amount of information will have to be maintained to check this and hence indeed the required finite state automaton can be constructed. This establishes that the language of complete firing sequences is regular and hence  $L(\mathcal{B})$  is also regular.

We note that the global control states can be augmented with the current finitely quantized values of the variables. The resulting languages of augmented global control states and hence its projection to just to the quantized values components will be regular. Consequently one can also reason—in terms of intervals of values—about the quantitative behavior of the plant.

We conclude by observing that the case where the communication graph is not connected can be easily dealt with. Clearly, 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 with some notational complications, it is easy to once again establish the main result.

## 4.5 The marked graph connection

A potentially useful fact is that the communication graph  $CG = (\Upsilon, A)$  can be viewed as the underlying directed graph of a marked graph [7].

We recall that in a marked graph, a node can fire iff all its input edges carry at least one token. 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 lead to a new marking. The reachable markings are the ones that are reached, starting from the initial marking through repeated node firings. Thus in the present setting both the variable nodes and location nodes can fire when enabled. Further, the initial marking is the one which places exactly one token on edges of the form  $(p, x_p)$ . All other edges are left unmarked.

By augmenting  $A$  with all the complementary arcs (i.e. add the arc  $(\eta, \eta')$  if  $(\eta', \eta)$  is in  $A$ ) and augmenting the initial marking suitably, one can obtain a live and safe marked graph [7]. 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 figure 3 will give rise to the live and safe marked graph shown in figure 4. The dotted arcs are the “complement” arcs that have been added to the communication graph. The (safe)

initial marking is indicated by the tokens (darkened circles) placed on some of the arcs. The key point is that a firing of the node  $\eta$  in this marked graph will correspond to a transition in  $\mathcal{B}$  caused by 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 in abstract terms the behavior of  $\mathcal{B}$ . For instance, we note that at the initial marking of the marked graph shown in figure 4, every variable node is enabled. Further, if  $(\eta, \eta')$  is an arc in this marked graph then the firings of  $\eta$  and  $\eta'$  will have to alternate. One can also use the acyclic path carrying a maximal number of tokens, namely the path  $p_3x_3p_2x_2p_1x_1$ , to determine that  $K = 3$  for this system where  $K$  is the constant asserted in Proposition 2.

More generally, this marked graph representation of the communication graph can be exploited to develop partial order based reduction methods including finite unfoldings [10] to efficiently verify the behavior of  $\mathcal{B}$ .

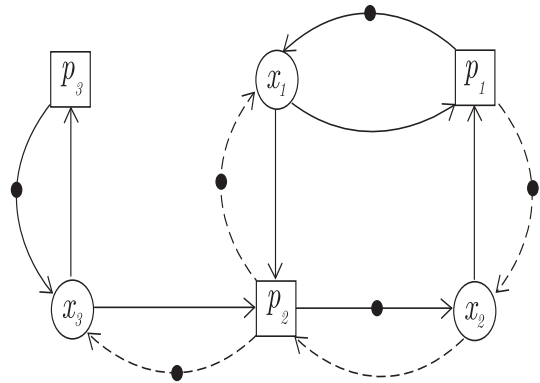


Figure 4: Marked graph representation

## 5. EXTENSIONS

Clearly the restriction that  $Ctl_p$  is a singleton for each  $p$  can be easily dropped. Each  $\rho_p(s)$  will now be a vector of rationals with one component for each variable in  $Ctl_p$ . We now turn to more interesting extensions.

### 5.1 Multi-Time Units and Communication Delays

We may associate with the plant a period  $T_{Plant}$  and with each controller  $p$  a period  $T_p$ , where  $T_{Plant}, T_p$  are positive rationals. We also fix a lower bound  $\theta_{min}$  and an upper bound  $\theta_{max}$  for the transmission times between each controller and the plant, where  $\theta_{min} < \theta_{max}$  are positive rationals. It will become clear in the sequel that one can also associate separate lower and upper bounds for each transmission link; from the plant to a controller, and from a controller to the plant.

A transmission of a value of variable from the plant at instant  $k \cdot T_{Plant}$  will reach a controller at some time in the interval  $[k \cdot T_{Plant} + \theta_{min}, k \cdot T_{Plant} + \theta_{max}]$ . Similarly, a transmission of rate update from a controller at instant  $k \cdot T_p$  will arrive at the plant at some time in the interval  $[k \cdot T_p + \theta_{min}, k \cdot T_p + \theta_{max}]$ .

Let  $\Delta$  be the largest rational which integrally divides every rational in  $\{T_{Plant}, \theta_{min}, \theta_{max}\} \cup \{T_p \mid p \in \mathcal{P}\}$ .

Then the dynamics of the plant and the controllers can be captured by a transition system of which each move will consist of letting  $\Delta$  time units pass, updating the variables, updating the modes of the controllers and updating the messages in transit between the plant and each controller. In particular, for each transmission of value  $v_x$  in transit from the plant to a controller, we will not keep track of the exact amount of time  $t$  that  $v_x$  has been in transit, but only which  $\Delta$ -size interval  $[m \cdot \Delta, (m + 1) \cdot \Delta]$  this time  $t$  lies in. Similar remarks apply to transmission of rate update from a controller to the plant.

To construct a distributed representation of the control state sequences of the plant and the controllers, we build a family of finite state automata as in section 4. However, each automaton  $\mathcal{B}_x$  will now simulate moves of controller  $p$  at every  $\Delta$  time unit. Similarly for each automaton  $\mathcal{B}_p$ .

The quantization of  $[B_{min}, B_{max}]$  will have to be changed as follows. We will now define  $\Omega$  to be the least set of rationals which satisfies the following conditions:

- $B_{min}, B_{max}$  are in  $\Omega$ .
- $\rho_p(s) \cdot \Delta \in \Omega$  for every  $p$  and every  $s \in S_p$ .
- Suppose  $(s_p, \varphi, s'_p) \in \longrightarrow_p$  and  $c$  appears in  $\varphi$ . Then  $c \in \Omega$ .

Thus we need to quantize using  $\rho_p(s) \cdot \Delta$  instead of setting  $\Delta = 1$ . In addition, each automaton  $\mathcal{B}_x$  will keep track of quantized values of values of  $x$  that are in transit from the plant to the controller. We note that at any instant, there will be at most  $\lceil \theta_{max}/T_{Plant} \rceil$  such values in transit between the plant and each controller. Similarly, each automaton  $\mathcal{B}_p$  will additionally keep track of (*boundedly many*) rate updates in transit from controller  $p$  to the plant. It is not difficult to extend the proofs of theorem 4 to this general setting.

## 5.2 Laziness

We can allow laziness of the plant in observing values of variables and in updating rates of variables. We fix positive rationals  $\delta_{min}^{ob}, \delta_{max}^{ob}, \delta_{min}^{up}, \delta_{max}^{up}$ , where  $\delta_{min}^{ob} < \delta_{max}^{ob}, \delta_{min}^{up} < \delta_{max}^{up}$ , with the following interpretation. Each value of  $x$  sent out by the plant at instant  $k \cdot T_{Plant}$  is the value that held at some instant in  $[k \cdot T_{Plant} - \delta_{max}^{ob}, k \cdot T_{Plant} - \delta_{min}^{ob}]$ . Each rate update  $\rho$  that the plant applies to variable  $x$  at instant  $k \cdot T_{Plant}$  kicks in at some instant in  $[k \cdot T_{Plant} + \delta_{min}^{up}, k \cdot T_{Plant} + \delta_{max}^{up}]$ .

The dynamics of the plant and the controllers can be defined in terms of a transition system  $\widehat{TS}$ , similar to the earlier setting of different periods of the controller and the plant, with time being quantized into units of  $\Delta$  as defined earlier. Each move of  $\widehat{TS}$  corresponds to the passage of  $\Delta$  time units.

On the other hand, to construct the distributed representations  $\mathcal{B}_x, \mathcal{B}_p$ , we shall first quantize time as well as the value range of variables at a finer level of granularity. Let  $\widehat{\Delta}$  be the largest rational which integrally divides every rational in the finite set  $\{T_{Plant}, \theta_{min}, \theta_{max}, \delta_{min}^{ob}, \delta_{max}^{ob}, \delta_{min}^{up}, \delta_{max}^{up}\} \cup \{T_p \mid p \in \mathcal{P}\}$ . Let  $\widehat{\Gamma}$  be the largest rational which integrally divides every rational in the finite set  $\{B_{min}, B_{max}\} \cup \{\rho_p(s) \cdot \Delta \mid p \in \mathcal{P}, s \in S_p\} \cup C$  where  $C$  is the collection of rationals which appear in guards of the transitions of the hybrid automata  $\{\mathcal{A}_p\}$ . As before, we then quantize  $[B_{min}, B_{max}]$  into finitely many sub-intervals of size  $\widehat{\Gamma}$ .

The constructions of  $\mathcal{B}_x, \mathcal{B}_p$  are similar as in the setting of the plant and the controllers having different periods. Each  $\mathcal{B}_x$  will keep track of the quantized interval in which the value of  $x$  lies. Each move of  $\mathcal{B}_x$  and  $\mathcal{B}_p$  will simulate the change of information of the plant, the controllers and messages in transit, due to passage of  $\widehat{\Delta}$  time units.

## 5.3 Finite Precision

We have assumed that variables can be measured with perfect precision and guards are rectangular. Following the techniques in [2], our results can also be extended to above generalized settings whereby variables are assumed be measured with finite precision and guards are allowed to be consisting of polynomial constraints that are to be evaluated against the measured values of variables. The key observation is that the finite precision condition allows one to transform polynomial guards (in fact any effectively computable guards) into rectangular guards on the *actual* values of variables.

## 6. CONCLUSION

We have formulated here a model consisting of a network of hybrid automata to study the behavior of a family of distributed controllers interacting with a plant. We have shown that the discrete time behavior of our model in terms of the control state sequences 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.

At present we have not considered differential inclusions. More precisely, the case where the rate of a continuous variable is specified as  $\frac{dx}{dt} \in [c, c']$  for rational constants  $c, c'$ . It is likely that the main result will go through but the details need to be worked out. It is also not clear if we can allow exponential rates induced by simple differential equations of the form  $\frac{dx}{dt} = c \cdot x(t)$ . An intriguing possibility is that in the restricted case where each hybrid automaton controls just one variable, the main result might go through. However, a good deal of coordination can still be achieved with the help of the guards and hence we do not wish to venture a conjecture at present.

As discussed earlier, an important extension would be to design mechanisms for the controllers to communicate with each other. Some obvious means would be to synchronize on common actions or to fix a message alphabet and use point-to-point bounded buffers to communicate asynchronously. A more elaborate approach will be to assign tasks to the controllers which can compute functions based on the values received from the plant as well as internal variables and exchange with each other the results of these computations, say, through a shared bus. Since the computations and communications will both be resource-bounded there will be complex interplays between the continuous behavior of the plant and the discrete behavior of the controllers. For instance, worst case execution times of the tasks and end-to-end delays in the communications between the controllers will impact on the trajectories that the plant will be allowed

to exhibit. We feel it will be particularly interesting to study this interplay in a Time-Triggered Architecture setting [5].

## 7. REFERENCES

- [1] M. Agrawal and P.S. Thiagarajan. Lazy rectangular hybrid automata. In *7th HSCC, LNCS 2993*, pages 1–15. Springer, 2004.
- [2] M. Agrawal and P.S. Thiagarajan. The discrete time behaviour of lazy linear hybrid automata. In *8th HSCC, LNCS 3414*, pages 55–69. Springer, 2005.
- [3] R. Alur, T.A. Henzinger, G. Lafferriere, and G.J. Pappas. Discrete abstractions of hybrid systems. *Proc. of the IEEE*, 88:971–984, 2000.
- [4] E. Asarin, O. Bournez, T. Dang, and O. Maler. Reachability analysis of piecewise-linear dynamical systems. In *HSCC 2000, LNCS 1790*, pages 20–31. Springer, 2000.
- [5] P. Caspi, A. Curic, A. Maignan, C. Sofronis, S. Tripakis, and P. Niebert. From Simulink to SCADE/Lustre to TTA: a layered approach for distributed embedded applications. In *Proc. of LCTES'03*, 2003.
- [6] O. Clarke, E.M. Grumberg and D.A. Peled. *Model Checking*. MIT Press, 1999.
- [7] F. Commoner, A.W. Holt, S. Even, and A. Pnueli. Marked directed graphs. *J. Computer and System Sciences*, 5:511–523, 1971.
- [8] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, 1995.
- [9] M.C.F. Donkers, L. Hetel, W.P.M.H. Heemels, N. van de Wouw, and M. Steinbuch. Stability analysis of networked control systems using a switched linear systems approach. In *HSCC 2009, LNCS 5469*, pages 150–164, 2009.
- [10] J. Esparza and K. Heljanko. *Unfoldings—A Partial-Order Approach to Model Checking*. Springer, 2008. EATCS Monographs in Theoretical Computer Science.
- [11] P. Godefroid. *Partial-Order Methods for the Verification of Concurrent Systems—An Approach to the State-Explosion Problem*. Springer, 1996. LNCS 1032.
- [12] T.A. Henzinger. Hybrid automata with finite bisimulations. In *22nd ICALP, LNCS 944*, pages 324–335. Springer, 1995.
- [13] T.A. Henzinger. The theory of hybrid automata. In *11th LICS*, pages 278–292. IEEE Press, 1996.
- [14] T.A. Henzinger and P.W. Kopke. Discrete-time control for rectangular hybrid automata. *Theoretical Comp. Sci.*, 221:369–392, 1999.
- [15] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What’s decidable about hybrid automata? *J. of Comp. and Sys. Sci.*, 57:94–124, 1998.
- [16] J. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proc. of the IEEE*, 95(1):138–162, 2007.
- [17] S. Tripakis. Decentralized control of discrete event systems with bounded or unbounded delay communication. *IEEE Trans. on Automatic Control*, 49(9):1489–1501, 2004.