The Concept of Deadlock and Livelock in Hybrid Control Systems

Alessandro Abate
Alessandro D'Innocenzo
Giordano Pola
Maria Domenica Di Benedetto
S. Shankar Sastry

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-181
http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-181.html

December 17, 2006

The Concept of Deadlock and Livelock in Hybrid Control Systems

Alessandro Abate[1], Alessandro D'Innocenzo[2], Giordano Pola[2,3],
Maria Domenica Di Benedetto[2], and Shankar Sastry[1]

[1] Department of Electrical Engineering and Computer Sciences,
University of California, at Berkeley - USA
{aabate,sastry}@eecs.berkeley.edu
[2] Department of Electrical Engineering and Computer Science,
Center of Excellence DEWS, University of L'Aquila - Italy
{adinnoce,pola,dibenede}@ing.univaq.it
[3] Department of Electrical Engineering,
University of California, at Los Angeles - USA
pola@ee.ucla.edu

Abstract. This paper introduces a formal definition of the concepts of Deadlock and Livelock for a general class of Hybrid Control Systems (HCS). Such a characterization hinges on three important aspects: firstly, the concept of composition of HCS; secondly, the general concept of specifications and their composition for HCS; finally, the dynamical structure and behaviors for HCS. The first aspect is introduced in a novel manner, including both aspects from the literature of discrete transition systems, and accounting for concepts such as feedback inteconnections of dynamical systems; the second point accounts for general properties that are of interest from a system theory and control perspective; the third part encompasses and categorizes between the diverse and possibly pathological behaviors that are characteristic to HCS. We investigate the problems of Deadlock and Livelock Verification.

1 Introduction

The concept of deadlock and its close relative, that of livelock, have been widely investigated in the literature of various branches of computer science [16]. Deadlock, in particular, has often been regarded as a pathology and associated with the deficiency of a liveness specification, that of forward progress [17][18]. Much interesting work has been focused in verifying the presence of deadlock situations in programs, or ensuring its absence upon their composition [3][5].

Hybrid Systems are rather general mathematical models that connect between discrete, logical, synchronous systems and continuous, real-time, asynchronous ones [1][2]. It has often been observed that they present behaviors or are endowed with properties that are "at the limit" between classical transition systems and dynamical models [2].

Motivated by a number of case studies, this work aims at "exporting" the notions of deadlock and livelock to the Hybrid Control Systems (HCS) case. More precisely, the objective has been that of first introducing a mathematically rigorous definition of the phenomena and providing a clear characterization of them; secondly, the task has been that of presenting some verification procedures for their existence; finally, the practical topic of resolution of these behaviors has also been looked at.

In this paper, we shall focus on the definition and characterization of deadlock and livelock. It is interesting to stress that this can be given only considering three important parts/components: first, that of *composition* of HCS, which we reformulate inspired by different approaches in the literature; second, that of *specifications*- and their composition- which we reinterpret from a control-meaningful perspective; finally, a purely dynamical level, which is characteristic of HCS, as opposed to transition systems for instance. It is possible to reinterpret known dynamical behaviors that are characteristic to HCS within this new characterization. We stress that for all the levels the introduced concepts nicely taylor back to similar ones that can be found in the literature of, respectively, discrete and continuous systems.

We develop a simple motivational case study to describe how to categorize the notion of deadlock within the theoretical framework we have developed. For simplicity, we shall focus on an interconnection of three controlled dynamical systems, rather than hybrid ones.

2 Deterministic Hybrid Control Systems: the Model

Let us start by introducing the concept of HCS, which is done in a rather detailed way for the sake of clarity of the definitions that shall be given in the sequel of the manuscript.

Definition 1 (Deterministic Hybrid Control System). *A* Deterministic Hybrid Control System *is a tuple $\mathcal{H} = (X, U, Y, F, T)$, where:*

- *X is the hybrid state space, composed of*
  - *$Q \subseteq \mathbb{N}$, a finite set of discrete states, or modes;*
  - *$D = \{D_i\}_{i \in Q}$, the set of domains associated with each mode; D_i is a subset of $\mathbb{R}^{n_i}$, where n_i may vary for different domains;*

  The space is specified as the disjoint union of the domains, i.e. $X = \cup_{i \in Q}\{q_i\} \times D_i$;
- *U is the control space, $U \subseteq \mathbb{R}^{n_U}$ and bounded; the control is a function defined on the set of nonnegative reals and with values in U, $u : \mathbb{R}_0^+ \to U$. We assume it is cadlag, i.e. piecewise-continuous from the right and with left limit, lying in $\mathbb{R}^{n_U}$;*
- *Y is the observation space, $Y \subseteq \mathbb{R}^{n_Y}$. The outputs will be determined by a static "observation" function $h : X \to Y$;*
- *$F = \{f_i\}_{i \in Q} : D \times U \to \mathcal{T}D$ is the set of vector fields. Each f_i is assumed to be continuous w.r.t. time and Lipschitz continuous with respect to the dependent variables; f_i characterizes the ODE $\dot{x} = f_i(x, u)$*[4]*, where $u(t) \in U$, for any $t \in \mathbb{R}_0^+$ and with initial condition $x_0 \in D_i$;*
- *T is the set of transition relations, composed of*
  - *$E \subseteq Q \times Q$, a set of edges; each edge $e \in E$ is an ordered pair of modes, the first component of which is the source and is denoted by s(e), while the second is the target and denoted by t(e);*

[4] For the sake of precision, we remark that the state and the input are different signals; one is defined over the hybrid time set (Def. 2 below), the second over the real time, which is a subset of the first. This is not disruptive and can be extended to time-dependent vector fields.

- $G = \{G_e\}_{e \in E}$, *a set of guards, where* $G_e : U \to 2^{D_{s(e)}}$*; the guards are "domain characteristic". We assume that* $\forall e', e'' \in E, e' \neq e'', s(e') = s(e''), \forall u \in U, G_{e'}(u) \cap G_{e''}(u) = \emptyset$*, i.e. the guards do not intersect in a domain; they are considered to be "forcing" the jumps;*
- $R = \{R_e\}_{e \in E}$, *where* $\forall e \in E, R_e : D_{s(e)} \to D_{t(e)}$*, a set of reset functions.*

In general the set of initial conditions (often called *Init* in the literature) is a subset of the hybrid state space X.

Remark 1 (Determinism of the Model). The assumptions of Lipschitz continuity of the vector fields[5], of non-intersection between guards[6] and their "forcing" feature, on the structure of the reset functions are sufficient for ensuring the *determinism* of the model for all initial conditions in *Init*, both in the continuous dynamics and in the discrete jumps. We allowed for resets onto a guard set, which translates into admitting multiple events at the same time; this does not exclude determinism, as also stressed in [2]. □

Remark 2 (Non-blocking Conditions). For $i \in Q$, let $G_i(u) = \bigcup_{e:s(e)=i} G_e(u)$, and define $\partial D_i = cl(D_i) \backslash int(D_i)$, where $cl(D_i)$ is the closure of D_i, while $int(D_i)$ is the interior of D_i. If the following holds:

$$(\forall i \in Q, \forall u \in U), \quad \partial D_i \cap D_i \subseteq G_i(u) \;\wedge\; \partial D_i \backslash D_i \subseteq cl(G_i(u)), \tag{1}$$

then this says that any trajectory never escapes a domain without hitting a guard, which is intended to be forcing the jump. The condition says that the boundary has to belong to the guard set, if it is part of the domain, or else to the closure of the guard set; it can be weakened, as long as we realize that the guard set is by definition a subset of the domain. Notice that the resets are always within a domain. □

Recall that we have introduced a dependence of the guards on the continuous control, but the resets are uncontrolled (they would be controlled if we had defined them as $R_e : G_e \to D_{t(e)}$); it is possible to extend this definition without affecting the determinism of the system, and avoiding to hamper the non-blocking feature by assuming that, $\forall e \in E, \forall u \in U, R_e : G_e \to D_{t(e)}$.

To define the executions of $\mathcal{H}$, we introduce the following classical notion [1]:

Definition 2 (Hybrid Time Set). *A hybrid time set* $\tau = \{I_k\}_{k \geq 0}$ *is a finite or infinite sequence of intervals* $I_k = [t_k, t'_k] \subseteq \mathbb{R}$ *such that*

1. I_k *is closed if* τ *is infinite;* I_k *might be right-open if it is the last interval of a finite sequence* τ*;*
2. $t_k \leq t'_k$ *for* $k > 0$ *and* $t'_{k-1} = t_k$ *for* $k > 1$.

[5] In particular, we stress the assumption with respect to the control input, which in general is allowed to be Lebesgue measurable, in particular piecewise-continuous, as in our case.

[6] Notice that, if the guards are control-dependent, this condition can be quite restrictive, but nevertheless necessary. In the uncontrolled case, it is instead quite natural to require this property.

The length $t'_k - t_k$ of every interval I_k denotes the dwelling time in a discrete location of the hybrid flow, while the extrema t_k, t'_k specify the switching instants. Let us stress that the above set is ordered; hence, it makes sense to use notations such as $t_k \leq t'_k$, as we shall do throughout the paper. A *hybrid trajectory*, or *hybrid flow*, is a pair (x, τ), where the first component is the hybrid state $x = (q, v) \in X$, that describes the evolution of the continuous part v and the discrete part q by means of functions defined on the hybrid time set τ and having value on X. Finally, an *hybrid execution* is a pair (τ, x) which can be algorithmically described as follows:

Algorithm 1 (Hybrid Execution):

1. *Pick $(q(t_0), v(t_0)) \in Init$, set $k = 0$, $\tau = \emptyset$;*
2. *Evolve the continuous trajectory $v(t)$ according to the vector field dependent on the exogenous control $u|_{[0,t)}$ and with initial condition $v(t_k)$ until a guard is hit: namely until time $t'_k \in [t_k, \infty)$ such that $v(t'_k) \in G_e\left(\lim_{t \to t'^{-}_k} u(t)\right)$, where $s(e) = q(t_k)$;*
3. *If $t'_k = \infty$, add $I_k = [t_k, \infty)$ to τ and exit the algorithm;*
4. *Else add $I_k = [t_k, t'_k]$ to τ. Define $q(t_{k+1}) = d(e)$ and $v(t_{k+1}) = R_e(v(t'_k))$. Increment k and go to line 2.*

Remark 3. Notice that, for generality sake (and in order to further taylor the model into that of discrete transition systems), in Def. 1 we have assumed a dependency of the guard on the control signal; the control is a piecewise right-continuous function: in order to rule out problems related to its discontinuity points, we have introduced a dependence on the limit from the left in the semantical definition of an event that coming from the intersection of a spatial guard. This assumption will play a role in ruling out "cycling conditions" on the composition of two HCS (see Remark 6). □

Within the set of hybrid executions, we shall carefully study the following subset:

Definition 3 (Zeno Executions). *Zeno executions are hybrid trajectories which are characterized by an infinite number of jumps in a finite amount of elapsed time. The hybrid time set of a Zeno trajectory has infinite cardinality and satisfies the following property:*

$$\sum_{k=0}^{\infty} (t'_k - t_k) < \infty$$

They can be of two kinds [15]: chattering and genuine. The first case happens when $\exists k^* : \forall k \geq k^*, t_k = t'_k = t_{k^*}$; the second happens if $\forall k \geq 0, \exists k^* \geq k : (t'_{k^*} - t_{k^*}) > 0$.

The output of the hybrid system is, for each execution, a function from the hybrid time set to the output space: it is not necessarily a function of the real time.

Since our purpose is to set up a notion of input-output interconnection, in the spirit of [5][6], we suppose that the *interconnected* output of hybrid systems considered is a physical signal expressed by means of a function of the real time, rather than of the hybrid time basis. This assumption is motivated by the aim of giving a notion of interconnection that is asynchronous and disregards the dynamics at the level of the internal states of the hybrid system.

Recalling the notion of $\tau = \{I_k\}_{k\geq 0}$, where $I_k = [t_k, t'_k]$, consider another set $\{J_k\}_{k\geq 0} \in \mathbb{R}^+_o$, where $J_k = [t_k, t'_k)$. Similarly to [21], we introduce:

Definition 4 (Interconnectable Outputs). *Given an HS and any output execution $y \in Y$, the* interconnectable output *associated with y is given by a right-continuous signal $\tilde{y}$, which is a function of the real time basis $\mathbb{R}^+_o$, defined by the action on y of a function $l : \tau \to \mathbb{R}^+_o$, such that, for any interval $[a, b] \in \tau, a \leq b$, consider the first $k \geq 0$ such that $[a, b] = \{[a, t'_k], \ldots, [t_{k+p}, b]\}$, where $p \geq 0$, then $l\Big(\{[a, t'_k], I_{k+1}, \ldots, I_{k+p-1}, [t_{k+p}, b]\}\Big) = \{[a, t'_k), J_{k+1}, \ldots, J_{k+p-1}, [t_{k+p}, b)\} \in \mathbb{R}^+_o$.*

Notice that the obtained "physical" signal $\tilde{y}(t)$ is cadlag. When dealing with HCS interconnections, we shall refer to the output spaces and implicitly assume to be working with these physical signals.

The model is a melange between the classic hybrid automaton [1] and the HIOA in [5][6]. In particular, we remark that we intend the state space to be an *internal* state, while the output and the control ones to be *interface/external* states: the model then is similar in structure to the hybrid automaton at an internal level, while it relates to the HIOA at a higher, external level. We avoid, motivated by a control interpretation of the model, to introduce the notion of *action* (either internal, or external): an action is simply the dynamic outcome of a state jump, or it can be possibly influenced by the exogenous control we have introduced above. We also do not distinguish a-priori between internal and external variables; in fact, we have implicitly assumed that there is a relationship between the state and the output spaces by the introduction of the observation function h; being this a function, determinism is retained. If the system is *fully-observable*, for instance, we take the continuous output space Y to concide with the largest of the domains D_i, once we have embedded all of them in $\mathbb{R}^{\max_i\{n_i\}}$; h performs this embedding. In this case it is unnecessary to define the state space as the disjoint union of the single domains, as they all coincide.

3 Hybrid Systems Composition

Abstractly, the concept of systems *composition* may be introduced in many ways, depending on the characteristics and properties of the systems that are considered (discrete or continuous, causal or non-causal, to name a few), the structure of the operation (parallel [9] or product [11], for instance), and the particular properties that we may want to check for (synchronicity or sequentiality, for example). In this work we consider an operation that may be interpreted as a form of *parallel composition*. A similar concept has been introduced, among the many places, in [3][4][9][10]. Notice that the introduction of a model structure with internal and external components, similar to that in [5][6], allows to conceive the system at the level of its hidden/internal variables (the hybrid state space with its vector fields and transition relations, as in Sec. 2) as a black box and only focus on the external components (the interface variables in Sec. 2) when performing the interconnection; this is the advantage of the interpretation as an I/O system. Unlike previous work, which simply performed parallel compositions or crude variables "sharing", inspireded here by a more control theoretical perspective we allow the connections between inputs and outputs of the systems to depend on general functions endowed with some properties: we naturally introduce an *output feedback* framework. We are in particular interested in a definition which may further taylor into known operations in the purely discrete case (transition systems) or dynamical instance (feedback interconnection). We suppose that:

Assumption 1. *The input spaces $U_i, i = 1, 2$ are rectangular sets, i.e. $U_i = [\underline{u}_1, \bar{u}_1] \times [\underline{u}_2, \bar{u}_2] \times ... \times [\underline{u}_m, \bar{u}_m]$, where $m = dim(span(U_i))$, $\underline{u}_j, \bar{u}_j \in \mathbb{R}$ and $\underline{u}_j \leq \bar{u}_j$.* □

The need to introduce this absence of mutual dependence between input components comes from the need to perform projections on their spaces, as it will become clear in the sequel. Before introducing the notion of composition of two hybrid systems $\mathcal{H}_1$ and $\mathcal{H}_2$, we raise the following condition:

Assumption 2 (Compatibility Conditions). *Upon composing two HCS $\mathcal{H}_1$ and $\mathcal{H}_2$, we assume that the two systems have no shared input or output variables. We say that $\mathcal{H}_1$ and $\mathcal{H}_2$ are compatible.*

The first assumption is raised for consistency with the definition below; the second assumption is to avoid the imposition of conditions on the dynamics of the two systems once they get interconnected. In the literature this structural requirement is a subset of the known *compatibility* conditions, [4][6]. We nevertheless recall that, as discussed at the end of Sec. 2, in general we allow for the (partial) coincidence of internal and external variables of two systems, unlike [5][6].

Given a set W, subset of a subspace $\mathcal{W} \subseteq \mathbb{R}^n$ with $dim(\mathcal{W}) = m$, let $M_{\mathcal{W}}$ be an invertible matrix such that for any $w \in \mathcal{W}$, $M_{\mathcal{W}} w = [\bar{w}^T, 0^T]^T$. Assume W is rectangular; then $\mathcal{W}$ is also rectangular, that is of the form $\mathcal{W} = a_1 span\{w_1\} \times a_2 span\{w_2\} \times ... \times a_n span\{w_n\}$, where $a_i \in \{0, 1\}$ and w_i are the canonical base vectors of $\mathbb{R}^n$. A (parallel) *composition procedure* between two systems is introduced as follows:

Definition 5 (HCS Composition). *Given two compatible HCS $\mathcal{H}_1 = (X_1, U_1, Y_1, F_1, T_1)$ and $\mathcal{H}_2 = (X_2, U_2, Y_2, F_2, T_2)$, a composition procedure $\|_\Sigma$ is specified by $\Sigma = \{\mathcal{W}_1 \times \mathcal{W}_2, g_1 \times g_2\}$, i.e. a set of rectangular subspaces $\mathcal{W}_i \subseteq span(U_i)$ and the maps $g_i : Y_{3-i} \rightarrow \pi|_{\mathcal{W}_i}(U_i), i = 1, 2$. The operation $\|_\Sigma$ yields $\mathcal{H}^\Sigma = \mathcal{H}_1 \|_\Sigma \mathcal{H}_2 = (X^\Sigma, U^\Sigma, Y^\Sigma, F^\Sigma, T^\Sigma)$, which is a new HCS made up of the following:*

- *$X^\Sigma = X_1 \times X_2$, $Init^\Sigma = Init_1 \times Init_2$ are the hybrid state space and initial conditions;*
- *$U^\Sigma = \pi|_{(\mathcal{W}_1^\perp \times \mathcal{W}_2^\perp)}(U_1 \times U_2)$ is the input space;*
- *$Y^\Sigma = Y_1 \times Y_2$ is the output space;*
- *$F^\Sigma = \{f^\Sigma_{(1,2)}\}_{(1,2) \in (Q_1 \times Q_2)}$ is the vector field; for any $(x_1, x_2) \in X^\Sigma$ and $(u_1, u_2) \in U^\Sigma$, $f^\Sigma_{(1,2)}((x_1, x_2), (u_1, u_2)) = f_1(x_1, M^{-1}_{\mathcal{W}_1}[u_1^T, g_1(h_2(x_2))^T]^T \times f_2(x_2, M^{-1}_{\mathcal{W}_2}[u_2^T, g_2(h_1(x_1))^T]^T$;*
- *T^Σ is the set of transition relations, composed of*
  - *$E^\Sigma = E_1 \times E_2$, is the set of edges;*
  - *$G^\Sigma = \{G^\Sigma_e\}_{e \in E^\Sigma}$, is the set of guards, where for any $(u_1, u_2) \in U^\Sigma$ and any $(e_1, e_2) \in E^\Sigma$ the guard set $G^\Sigma_{(e_1,e_2)}$ is implicitly defined by $(x_1, x_2) \in G^\Sigma_{(e_1,e_2)}$, where $(x_1, x_2) \in (G_1)_{e_1}(M^{-1}_{\mathcal{W}_1}[u_1^T, g_1(h_1(x_2))^T]^T) \times (G_2)_{e_2}(M^{-1}_{\mathcal{W}_2}[u_2^T, g_2(h_2(x_1))^T]^T)$;*
  - *$R^\Sigma = \{R_e\}_{e \in E}$, is the reset function, where $\forall e = (e_1, e_2) \in E^\Sigma, R^\Sigma_e = (R_1)_{e_1} \times (R_2)_{e_2}$.*

Remark 4 (On the Composition). The interconnecting functions g_1, g_2 turn a transformation of part of the original output spaces into part of the original input spaces. Notice that we have not assumed that the input and output spaces have the same dimension: the dimensionality is handled directly by the interconnecting functions; in particular, the interconnection maps the output signal to a subset of the input space. More precisely, the functions g_i are defined on a subset of the vector spaces, $dom(g_i) \subseteq Y_{3-i}$; its dimension tells the number of signals employed in the interconnection, while the dimension of the codomain $dim(\mathcal{W}_i)$ defines the number of signals actually connected. The way they get connected to the input signals is further specified by the subspaces $\mathcal{W}_i$ in an intuitive way. More general compositions, that allow the output signals to be "shuffled" (rather than just ordinately clustered), can be directly defined by appropriately redefining matrices $M_{\mathcal{W}_i}$. The new input set is the projection of the cartesian product of the two original input sets onto the space of the "unused signals", i.e. of those inputs that have accepted no feedback. Clearly, in the full-feedback case, the codomains of the g_i shall be the whole input spaces U_i, thus yielding a fully dynamic (uncontrolled) composition. Finally, let us stress that it is possible to express $g = g_1 \times g_2$, i.e. as a cartesian product between functions.

Remark 5 (On the Events of the composed system). The events of the composed system are specified "asynchronously"; in other words, even though the dynamics of the two models reciprocally affect each other, the events happen within the single system, possibly at the same time. □

Remark 6 (Asynchronicity of the Composition, Absence of Cyclic Constraints). Both $\mathcal{H}_1$ and $\mathcal{H}_2$ are asynchronous by definition and their composition $\mathcal{H}^c$ is asynchronous as well: it can indeed be viewed from the outside as an interconnection of two dynamical systems. The reader should realize that this does not exclude the presence of pathological events (Zeno or blocking, for instance), which arises at an internal level. For now, we do not impose any a-priori condition to exclude this, unlike previous literature, where the focus was on ensuring infinite forward progress of time under systems' composition [3][4][5][6]. As discussed, the discrete events affecting the variables of each system are allowed to happen without any imposed syncronization, even if we have allowed the internal (state) and external (outputs) variables to possibly partially overlap; this is due to the semantics of the events, which depend on the limit value of the control, rather than its value at a certain time. In the literature [5][6], this "cyclic constraints" have been artificially avoided by splitting up internal and external variables, which we do not want to assume here; another method to prevent these conditions has been that of imposing some ordering, or sequentiality, between the signals in the loop. This would prevent the introduction of the concept of dynamic feedback, and hence we avoid raising such assumption. □

The following can be derived from Remark 1 and the way the composition in Def. 5 is performed. The easy proof is omitted for space limitations.

Proposition 1 (Determinism of the Composition). *Given a pair of deterministic HCS $\mathcal{H}_1$ and $\mathcal{H}_2$ and a composition $\Sigma = \{\mathcal{W}_1 \times \mathcal{W}_2, g_1 \times g_2\}$, if the maps g_i are almost-everywhere continuous, then the hybrid system $\mathcal{H}^{\Sigma} = \mathcal{H}_1 \|_{\Sigma} \mathcal{H}_2$ is deterministic.*

Remark 7 (Commutativity and Associativity of the Composition). The composition is trivially *commutative* by the way it was defined. The *associative property* is verified in the following sense: given three HCS $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$, if it is legitimate to compose them as $(\mathcal{H}_1\|_{\Sigma_1}\mathcal{H}_2)\|_{\Sigma_2}\mathcal{H}_3$, then it holds that $(\mathcal{H}_1\|_{\Sigma_1}\mathcal{H}_2)\|_{\Sigma_2}\mathcal{H}_3 = \mathcal{H}_1\|_{\Sigma_1}(\mathcal{H}_2\|_{\Sigma_2}\mathcal{H}_3)$. We have not specified the details of the composition procedures, but just assumed that there exist interconnecting maps that respect the domain dimensionality and bounds so that the composition of the three systems can be performed in one way. Showing that it is possible, and indeed equivalent, to perform the composition in the other order is only a matter of calculations. We remark that this property ensures the generality of Def. 5 in that it can be repeated more than once without worrying about the order; this will also be exploited when reasoning about composing specifications in the ensuing sections. Another issue that precedes the above property is the seek of conditions that ensure that the composition between a number of HCS is structurally allowed. A necessary condition for this fact to hold for a composition between a set $\mathcal{H}_i, i \in \mathcal{I}$ through $\Sigma = \{\Sigma_{j,k}, j, k \in \mathcal{I}\}$, is that the codomains[7] of all the interconnecting maps to any particular input space do not intersect : $\forall i \in \mathcal{I}, \forall j, k : \{\Sigma_{j,i}, \Sigma_{k,i}\} \in \Sigma, \mathcal{W}_j \cap \mathcal{W}_k = \emptyset$. □

We conclude the Section by remarking that more general compositions can be described with slight changes to the setup given in this Section. We mention the possibility to define feedbacks with "self-loops" within a single HCS. Futhermore, it is also interesting to look at time-varying interconnections, which introduce a "switching" composition level that may introduce interesting, but possibly pathologic, phenomena, such as Zeno [13].

4 Composing Hybrid Systems Specifications

In this section we consider rather general specifications defined on trajectories on the observation space (or, possibly, in the fully-observable case, on the state space); they may be defined, for instance, via temporal logic formulas for real-time systems. Furthermore, we shall also introduce an explicit dependence on the control signals: this would allow to express specifications that are general enough to cover the most important problems in control theory. Let $\mathcal{M} \subseteq Y$ and recall that U is the set of control inputs for $\mathcal{H}$. We set up the following problems as paradigms for our study:

1. *reachability*: $\varphi_R(U, \mathcal{M}) := \exists u^* : \mathbb{R}_0^+ \to U, \exists t^* < \infty : y(t^*) \in \mathcal{M}$;
2. *attractivity*: $\varphi_A(U, \mathcal{M}) := \forall u : \mathbb{R}_0^+ \to U, \exists t^* < \infty : y(t^*) \in \mathcal{M}$;
3. *invariance*: $\varphi_I(U, \mathcal{M}) := \forall u : \mathbb{R}_0^+ \to U, \forall t \geq 0, y(t) \in \mathcal{M}$;
4. *viability*: $\varphi_V(U, \mathcal{M}) := \exists u^* : \mathbb{R}_0^+ \to U : \forall t \geq 0, y(t) \in \mathcal{M}$.

The known *liveness* property, which is quite important and object of much investigation in the literature on transition systems, can be reinterpreted within the first two of the above properties; in particular, the requirement of *forward progress* has been object of investigation [3][4][16][17] for its obvious practical implications. The last two properties can be thought of as *safety* specifications. The first and fourth instances can be

[7] This is stronger than a similar requirement on the ranges of these functions.

intended as *control synthesis* problems, while the second and third as *verification* problems. Another relevant specification we shall be referring to is that of *time progression*, which entails the study of phenomena like blocking or Zeno: this can be conceived as a liveness property at large.

Notice that the above definitions have been given in generality with respect to the signal $y(t) \in Y$, output of the control-dependent dynamics of $\mathcal{H}$, and which is defined on the hybrid time set ($t \in \tau$, according to Def. 2). Because of the hypothesis for the determinism for the model (see Remark 1), given an initial condition and a time-dependent control profile, the generated hybrid trajectory is unique. We write $x_0 \models_{\mathcal{H}} \varphi(U, \mathcal{M})$ (and say that x_0 satisfies $\varphi(U, \mathcal{M})$ for $\mathcal{H}$) if that specification (which also depends on the set $\mathcal{M}$ and is further quantified over the controls in U) is satisfied by executions with initial condition $x_0 \in X$. We are then interested in the set of initial conditions that satisfy a given specification: more formally, we introduce the following subset of the initial conditions: $\mathcal{X}_{\mathcal{H}} = \{x_0 \in Init : x_0 \models_{\mathcal{H}} \varphi(U, \mathcal{M})\}$.

We write $\mathcal{H} \models \varphi(U, \mathcal{M})$ (and say that $\mathcal{H}$ satisfies $\varphi(U, \mathcal{M})$) if $\mathcal{X}_{\mathcal{H}} = Init$. Let us now consider two hybrid systems $\mathcal{H}_1$ and $\mathcal{H}_2$, corresponding specifications $\varphi_1(U_1, \mathcal{M}_1)$, $\varphi_2(U_2, \mathcal{M}_2)$ and a composition procedure Σ. We are interested in composing the two specifications φ_1, φ_2. Similar to [12], we define

Definition 6 (Composition of Specifications). *Given the two HCS $\mathcal{H}_1, \mathcal{H}_2$ with specifications $\varphi_1(U_1, \mathcal{M}_1)$, $\varphi_2(U_2, \mathcal{M}_2)$ and a composition procedure Σ, the composed specification $\varphi^{\Sigma}(U^{\Sigma}, \mathcal{M}_1 \times \mathcal{M}_2) = \varphi_1(U_1, \mathcal{M}_1) \|_{\Sigma} \varphi_2(U_2, \mathcal{M}_2)$ for the HCS $\mathcal{H}^{\Sigma}$ is the conjunction $\varphi_1(U_1, \mathcal{M}_1) \wedge \varphi_2(U_2, \mathcal{M}_2)$, modulo proper substitutions of variables, according to the composition maps g_1, g_2 that characterize Σ* [8].

4.1 Specification Retention on the HCS Composition

The composed system $\mathcal{H}^{\Sigma}$ can be investigated from the perspective of the existence of initial conditions that yield to the verification of the composed specification:

$$\mathcal{X}_{\mathcal{H}^{\Sigma}} = \{(x_1^0, x_2^0) \in Init_1 \times Init_2 : (x_1^0, x_2^0) \models_{\mathcal{H}^{\Sigma}} \varphi^{\Sigma}\}. \tag{2}$$

Rather than verifying a certain specification in generality on the HCS composition as in Eqn. (2), we may be interested in checking whether some verified properties on the original systems are retained through a certain composition; this is motivated by the introduction of a definition of the concepts of deadlock and livelock, which, as it shall be motivated in the sequel, is associated with pathological behaviors at the level of the composition. More precisely, we look at the following set:

$$\bar{\mathcal{X}}_{\mathcal{H}^{\Sigma}} = \{(x_1^0, x_2^0) \in \mathcal{X}_{\mathcal{H}_1} \times \mathcal{X}_{\mathcal{H}_2} : (x_1^0, x_2^0) \models_{\mathcal{H}^{\Sigma}} \varphi^{\Sigma}\}.$$

It should be clear that the following containment relationships hold: $\bar{\mathcal{X}}_{\mathcal{H}^{\Sigma}} \subseteq \mathcal{X}_{\mathcal{H}^{\Sigma}} \subseteq Init_1 \times Init_2$. Note that $\mathcal{X}_{\mathcal{H}^{\Sigma}}$ is the subset of initial conditions $Init_1 \times Init_2$ of $\mathcal{H}^{\Sigma}$ that satisfy the composed specification. We are interested in $\bar{\mathcal{X}}_{\mathcal{H}^{\Sigma}}$, that is the subset of the

[8] In [12], a space embedding is also necessary in the case of composition of specifications defined on heterogeneous spaces; Def. 1 and Def. 5 render this unnecessary.

initial conditions $\mathcal{X}_{\mathcal{H}_1} \times \mathcal{X}_{\mathcal{H}_2}$ that separately satisfy the single specifications φ_1, φ_2 for $\mathcal{H}_1, \mathcal{H}_2$, and also satisfy the composed specification φ^{Σ} for $\mathcal{H}^{\Sigma}$. We are finally able to define an even more interesting set:

$$\tilde{\mathcal{X}}_{\mathcal{H}^{\Sigma}} = (\mathcal{X}_{\mathcal{H}_1} \times \mathcal{X}_{\mathcal{H}_2}) \backslash \bar{\mathcal{X}}_{\mathcal{H}^{\Sigma}}.$$

This is the set of initial conditions that, taken separately i.e. within the original system, would satisfy the corresponding properties for that single system, but which do not satisfy the composed property. Notice that, if the specifications φ_1, φ_2 are expressed with the universal quantifier on the control variable (namely $\forall u_i \in U_i, i = 1, 2$), then $\bar{\mathcal{X}}_{\mathcal{H}^{\Sigma}} = \mathcal{X}_{\mathcal{H}_1} \times \mathcal{X}_{\mathcal{H}_2}$, and thus $\tilde{\mathcal{X}}_{\mathcal{H}^{\Sigma}} = \emptyset$: in fact, the interconnection restricts the set of available input signals for the composed system; then, if a property holds for all the control input signals, it also holds for a subset of them. This motivates to restrict the attention to the case where either φ_1 or φ_2 handle the variable u_i with an existential quantifier (i.e. $\exists u_i \in U_i, i = 1, 2$). In other words, we shall focus on control synthesis problems.

The elements of $\tilde{\mathcal{X}}_{\mathcal{H}^{\Sigma}}$ are initial states of $\mathcal{H}^{\Sigma}$ that are associated with "pathological" behaviors coming out of the interconnection, when looked from the perspective of the verification of the composed specification: this can be due to different reasons for the same initial condition in $\tilde{\mathcal{X}}_{\mathcal{H}^{\Sigma}}$, i.e. to different control profiles that generate different "bad" trajectories. Within the set $\tilde{\mathcal{X}}_{\mathcal{H}^{\Sigma}}$ we shall be looking for deadlock and livelock executions: to this aim, we categorize these initial conditions into three possibly overlapping subsets:

$$\tilde{\mathcal{X}}_{\mathcal{H}^{\Sigma}} = \tilde{\mathcal{X}}^{d}_{\mathcal{H}^{\Sigma}} \cup \tilde{\mathcal{X}}^{l}_{\mathcal{H}^{\Sigma}} \cup \tilde{\mathcal{X}}^{e}_{\mathcal{H}^{\Sigma}}. \tag{3}$$

In the passing, we remark that the above three sets are non overlapping if the composition is fully dynamic, namely if $U^{\Sigma} = \emptyset$. The categorization in Eqn. (3) hinges on the dynamical properties of the generated executions, as described in the next Section.

5 Formal Definition of Deadlock and Livelock for HCS

Let us start by introducing the following concept:

Definition 7 (Hybrid Invariant Set). *For a HCS $\mathcal{H}$ and a hybrid trajectory (x, τ) originating from a particular control profile, a hybrid set $\mathcal{I} \subseteq X$ is defined to be* invariant *if the following holds: if $x(t^*) \in \mathcal{I}$ for $t^* \in \tau$, then $x(t) \in \mathcal{I}, \forall t \geq t^*$.*

From a dynamical standpoint, the concepts of deadlock, or livelock, are intrinsically related to the idea of a trajectory being "constrained" or "stalled" somewhere in the state space. This locking condition is then further specified with regards to the presence or absence of indefinite motion within the region. We then distinguish the pathological trajectories associated with Eqn. (3) as follows:

Definition 8 (Deadlock and Livelock for HCS). *The items in $\tilde{\mathcal{X}}_{\mathcal{H}^{\Sigma}}$ belong to either of the following:*

- *$\tilde{\mathcal{X}}^{e}_{\mathcal{H}^{\Sigma}}$, the set of initial conditions that correspond to trajectories that do not encounter an invariant set in X^{Σ};*

- $\tilde{\mathfrak{X}}^d_{\mathcal{H}\Sigma} \cup \tilde{\mathfrak{X}}^l_{\mathcal{H}\Sigma}$, *the complement of the above set in* $\tilde{\mathfrak{X}}_{\mathcal{H}\Sigma}$, *the set of initial conditions associated with trajectories that enter an invariant set* $\mathfrak{I}$.

The second set is further composed of:

- $\tilde{\mathfrak{X}}^d_{\mathcal{H}\Sigma}$, *the set of initial conditions associated with* deadlock *executions: these are defined by absence of motion in finite time ("stalling" situation);*
- $\tilde{\mathfrak{X}}^l_{\mathcal{H}\Sigma}$, *the set of initial conditions associated with a* livelock *situation: these are characterized by endless motion, either in their continuous or discrete component.*

Notice that the definition above hinges on a purely *dynamical* level; this represents the last point, after that of *composition* and that of *specification*, which is regarded as necessary to introduce the notions of deadlock and livelock in the framework of HCS. Let us study how special types of dynamics are categorized within the above definition:

- Deadlock Situations:
  - *blocking conditions*: states for which no "next" state is defined.
  - *stable equilibria in finite time*: equilibria in the continuous dynamics that are reached by a reset operation.
  - *chattering Zeno*: the discrete component infinitely jumps instantaneously between different domains, while the continuous component is still (Def. 3).
  - *genuine Zeno*: the hybrid trajectory performs an infinite number of transitions in a finite amount of time (see Def. 3).
- Livelock Situations:
  - *stable equilibria in infinite time*: equilibria for the continuous dynamics.
  - *limit cycles*, both in the continuous and the discrete dynamics.

It is easy to add to the above that the set $\tilde{\mathfrak{X}}^e_{\mathcal{H}\Sigma}$ is associated with *diverging trajectories*: such trajectories are characterized by finite or infinite escape time, either in the continuous, or the discrete component.

Remark 8 (On Zeno Phenomena). Zeno behaviors are peculiar phenomena that occurr to HCS models and which truly highlight their structural characteristics. Even in this particular instance, it is interesting to stress that their characterization is "in between" that of deadlock and livelock. They are in fact similar to a blocking condition, in that they are not defined for all the (hybrid) time set, while adhering to the second group as they are endowed with an "infinite motion". We categorized them within the deadlock situations also because of the practical outcome involving their presence, that of "stalling" a program that simulates them. □

6 Deadlock Verification

From the above discussion, it comes at no surprise that the next obligatory step after that of defining and characterizing the notion of deadlock and livelock for HCS is that of looking at ways to detect it; furthermore, it would be desirable to derive some conditions to prevent it a-priori, or synthesize actions in order to resolve it. It has been motivated that the new notions depend on a dynamical level, on the notion of composition, and on the specifications that are attached to the problem: we shall then look at all these levels and leverage on each of them for the following tasks. Recall that we discussed about the last point at the beginning of Sec. 4.

Detecting particular behaviors through the use of sufficient conditions for their existence is in general hard (see for instance [15] for the Zeno cases). Secondly, model checking for certain properties requires a simulation-based approach and is in general doomed to be undecidable. It appears to be easier to come up with conditions ensuring the absence of these pathological conditions.

To begin with, according to Def. 7, let us focus our attention on the presence of invariant sets on the composed HCS: this is a computationally heavy task. We can resort to the back-propagation of equilibria or $\omega-$limit sets. In the hybrid case the search is indeed not easy [19]. The reason for this can be traced back to the variety of possible behaviors, especially switching ones; this difficulty is in fact related to the hardness in finding conditions for stability in HCS [20].

Regarding the "other side" of the set of pathological trajectories, divergent trajectories can be ruled out by proper Lipschitz conditions (for the finite-time, continuous-space case), reset assumptions (no resets on guards, for the finite-time, discrete-space case) and domain boundedness conditions (for infinite-time case).

The next step involves looking into the specific types of locking behaviors. For simplicity, let us raise the following

Assumption 3. *Let the assumptions invoked in Proposition 1 be in order.* □

Deadlock Avoidance.

Proposition 2. *Under condition in Eqn. (1) in Remark 2 on $\mathcal{H}_1, \mathcal{H}_2$ and considered the composition procedure Σ, HCS $\mathcal{H}^\Sigma$ does not generate blocking executions.*

Let $\mathcal{E} = \{x^* = (q^*, v^*)\}$ be the set of equilibria points on X^Σ, the state space of $\mathcal{H}^\Sigma$; if the composition is purely dynamical, i.e. an autonomous system, the definition is known; if, instead, $\mathcal{H}^\Sigma$ is controlled, then the points in $\mathcal{E}$ are defined to be such for all the possible applicable controls in U^Σ.

Proposition 3. *Given $\mathcal{H}^\Sigma$, if range(R^Σ_e) $\cap\, \mathcal{E} = \emptyset$ for all $e \in E^\Sigma$, then $\mathcal{H}^\Sigma$ does not generate deadlock executions coming from finite-time stable executions.*

Definition 9 (Cycle). *Given a system $\mathcal{H}$, a collection of edges $c = \{e_k\}_{k=0\cdots s}, e_k \in E$ is a cycle of length $|c| = s + 1$ if $d(e_k) = s(e_{k+1}), \forall k < s$ and $s(e_1) = d(e_s)$.*

Proposition 4. *Given $\mathcal{H}^\Sigma$, let $\mathcal{C}$ be the set of all cycles on T^Σ. If there exists $c \in \mathcal{C}$ such that range($R^\Sigma_{e_k}$) $\cap\, G^\Sigma_{e_{(k+1) mod(|c|)}} = \emptyset$ for all $k = 0 \cdots |c| - 1$, then $\mathcal{H}^\Sigma$ does not generate deadlock executions coming from chattering Zeno, as intended in Def. 3.*

Let us now investigate conditions to prevent genuine Zeno. We shall follow an approach already presented in [5][6], and in part inspired by older work [3][4]. In Assumption 2, conditions for the *compatibility* between HCS for their composition have been introduced. More interestingly, we may look for conditions that ensure that the composed system, when control-dependent, accepts any possible input trajectory, given the compatibility of the two original systems: the notion of *strong compatibility* has been introduced in [5][6] to formalize this concept. The following holds:

Proposition 5. *The composition $\mathcal{H}^{\Sigma}$ of two compatible HCS $\mathcal{H}_1$ and $\mathcal{H}_2$ is strongly compatible. A particular instance is the case of a purely dynamical connection $\mathcal{H}^{\Sigma}$.*

The above claim, which is not proven here because of space constraints, is related to the "Lipschitz HIOA" case in [5]. In that work, the notion of *strategy* is further introduced, that is of deterministic refinement of a HCS, which in our case coincides with the system $\mathcal{H}^{\Sigma}$ itself (see Assumption 3). We are in particular interested in seeking *progressive strategies*, that is of a system $\mathcal{H}^{\Sigma}$ for which time progresses to infinity (no more complicated liveness specifications are needed). In our case, progressiveness is postulated by definition of the HCS models $\mathcal{H}_1, \mathcal{H}_2$, in fact they are assumed to be blocking-free; hence, to get a progressive strategy it is further important to make sure that the absence of Zeno is preserved by composition.

Proposition 6. *Reinstate the non-blocking conditions in Remark 2 (Eqn. (1)), which ensure that the composition $\mathcal{H}^{\Sigma}$ is non-blocking. Then, according to Thm. 7.4 in [5], $\mathcal{H}^{\Sigma}$ is a progressive strategy.*

Now, defining "receptive" a HCS with progressive strategy, combining the hypotheses of Propositions 5 and 6, the following holds: "strong compatibility", to have a *receptive* HCS as a composition, with a progressive stategy.

Corollary 1. *Under condition in Eqn. (1) in Remark 2 for two compatible HCS $\mathcal{H}_1, \mathcal{H}_2$, we conclude that $\mathcal{H}^{\Sigma}$ is receptive, and hence free of genuine-Zeno, as in Def. 3.*

A more structural approach to the issue is the following: the Lipschitz property of the vector fields of $\mathcal{H}^{\Sigma}$ entails a bound on the "speed" of the trajectories dwelling in X^{Σ}; this can be exploited postulating that the codomain of the reset maps is bounded away from the guard set, i.e. assuming that it will take at least a minimum, non-zero amount of time for the trajectories to get to the boundary. This, according to Def. 3, shall rule out genuine Zeno:

Proposition 7. *Given $\mathcal{H}^{\Sigma}$, if $dist(cod(R^{\Sigma}), G^{\Sigma}) > \delta > 0$ then $\mathcal{H}^{\Sigma}$ does not generate deadlock trajectories coming from genuine Zeno, as intended in Def. 3.*

Livelock Avoidance.

Proposition 8. *Given $\mathcal{H}^{\Sigma}$, consider the set $\mathcal{E}$ of $\mathcal{H}^{\Sigma}$: if this set is unreachable from $\mathcal{X}_{\mathcal{H}_1} \times \mathcal{X}_{\mathcal{H}_2}$, then $\mathcal{H}^{\Sigma}$ will have no Livelock behavior coming from infinite-time stability.*

The approach can be extended in spirit to include detection of limit cycles.

7 Example: Three cars crossing an intersection

Consider the interconnection of three models $H_i, i \in \{0, 1, 2\}$, each of which describes the dynamics of a car along a road with a stop sign. The dynamical model describes the dynamics associated with the car's position, and is hence one dimensional. The intersection is assumed to be positioned at the origin. The i^{th} car is assumed to be controlled through its velocity: we distinguish five operation regions in its domain. In the first (leftmost), the vehicle speeds toward the intersection, until it reaches the second region, where it is assumed to decelerate until hitting the intersection; there, it may move on, and thereafter accelerate until gettin out of a buffer zone, where the control is assumed to steer the car away from the intersection. Quantitatively, the model is the following:

$$\dot{x}_i = \begin{cases} u_i + \delta & \text{, if } x_i < -c \\ -\frac{u_i}{c} x_i + \delta & \text{, if } -c \le x_i < 0 \\ \frac{u_i}{d} x_i & \text{, if } 0 \le x_i < d \\ u_i + \delta & \text{, if } x_i \ge d \end{cases}$$

Here we assume that $c, d > 0$, that the initial conditions $x_i^o \in Init_i = (-\infty, -c]$, and that $\delta > u_i$ (so that the car decelerates, yet arrives at the intersection in finite time). The continuous control u_i belongs to a set $U_i = [0, N], N < \infty$; assume the control is cadlag. The control is space is then U_i. The output is the state – we are in the case of full observability. Notice that the structure of the vector fields, with their simple dependence on the control signals, which are piecewise-continuous from the right, as well as the absence of guards, defines the state trajectory, and hence the output, on the real time. It is hence not necessary to introduce the maps l, which turn output signals to interconnectable ones.

We specify for each system a reachability (and liveness) property, which denotes the possibility of the car to reach a certain point "way ahead of the stop sign":

$$\varphi_R(U_i, K) := \exists u_i^* : [0, \infty) \to U_i, \exists\, 0 \le t^* < \infty : x_i(t^*) = K > d.$$

It should be clear that for all the three systems, taken by themselves, for each initial condition there exists a control (discrete and continuous) such that the system satisfies the liveness specification. More precisely,

$$\forall u_i \in (0, N], \forall x_i^o \in Init_i, x_i^o \models_{H_i} \varphi_R, \text{ hence } H_i \models \varphi_R.$$

Hence, we can say that $\mathcal{X}_{H_i} = Init_i$. Notice that the left closure of the sets U_i make a similar "attractivity" specification not be verified.

We interconnect the three systems in order to model the intersection of a three-way stop (see Figure 1). Furthermore, as customary for the majority of real world traffic codes, we impose a "yield to the right" rule. Introduce the indicator function

$$\mathbf{1}_{[-c,0]}(x) = \begin{cases} 1, & \textit{if } x \in [-c, 0]; \\ 0, & \textit{else.} \end{cases}$$

The interconnection between $H_{(i+1)mod3}$ and H_i is specified by $\Sigma_{i,(i+1)mod3} = \{\mathbb{R}, g_{i,(i+1)mod3}\} = \{\mathbb{R} \times \emptyset, g_{i,(i+1)mod3} \times \tilde{g}\}, i \in \{0, 1, 2\}$, where $\tilde{g} = g_{(i+1)mod3,i}$ is undefined because $\mathcal{W}_{(i+1)mod3,i}$ is the empty set. The following function defines the structure of the interconnected inputs:

$$u_i = g_{i,(i+1)mod3}(y_{(i+1)mod3}) = \mathbf{1}_{[-c,0]^C}\,(x_{(i+1)mod3}),$$

where we have denoted with $[-c, 0]^C$ the complement of $[-c, 0]$. It is easy to realize that, given the structure of the $\Sigma_{i,(i+1)mod3}, i \in \{0, 1, 2\}$, the interconnection is associative;

hence, we naturally compose the three models with no particular order. The pairwise composed system is controlled in that $U^{\Sigma_{i,(i+1)mod3}} = U_{(i+1)mod3}$.

Define finally $H^{\Sigma} = H_1||_{\Sigma_{2,1}}H_2||_{\Sigma_{3,(2,1)}}H_3||_{\Sigma_{1,(3,(2,1))}}H_1$; the output of the composition is purely dynamical (see Figure 2).

The reader should easily convince himself that, for a particular set of the model parameters, there exist a combination of initial conditions for the three systems which yield to a condition that has the three cars stopped at the intersection, each of them “waiting” for the next on its right to proceed. This is intuitively a *deadlock* situation. Such a situation is obtained, for example, when the initial condition $x_i^0 \in [-c, 0], i \in \{0, 1, 2\}$. If we in fact look at the system in its entirety, we understand that from a dynamical standpoint the obtained deadlock condition corresponds to finite-time equilibrium.

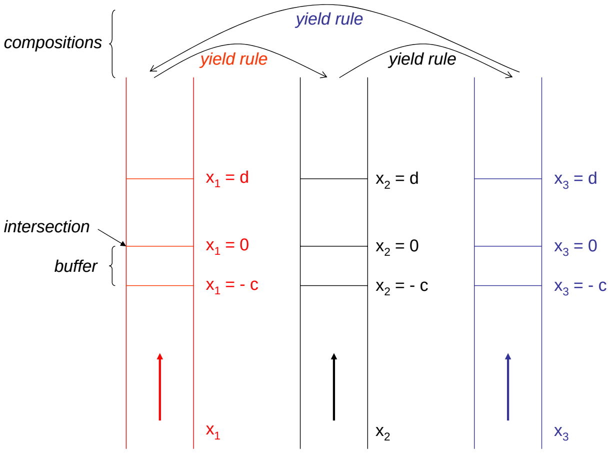

Fig. 1. Composition of the three dynamical control systems according to specific rules.

8 Conclusions and Future Work

It is interesting to notice that the ideas introduced in the paper taylor to know one from the literature on distrete transition systems, HIOA, or dynamical systems.

Deadlock and Livelock *resolution* is another topic that does not find space in the present paper; this take-away point is to add noise to the dynamics of the model in order to probabilisitically reduce the likelihood of deadlock executions.

The authors are working on many enticing extensions of the presented results. The concept of composition is prone to be generalized, and the issue of “deep composition”, i.e. of a composition procedure preserving certain properties, clearly connects with the above effort when the absence of deadlock or livelock is the specification to be exported.

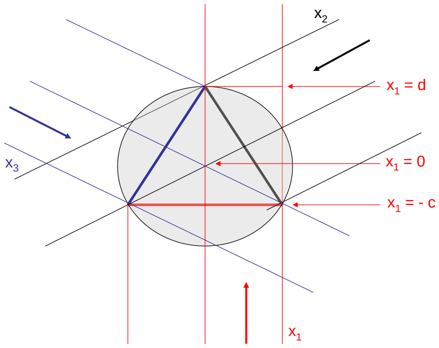

Fig. 2. Three cars crossing an intersection.

References

1. John Lygeros: *Lecture Notes on Hybrid Systems.* ENSIETA, 2-6/2/2004.
2. John Lygeros, Karl Henrik Johansson, Slobodan N. Simic, Jun Zhang, Shankar Sastry : *Dynamical Properties of Hybrid Automata.* IEEE Transactions On Automatic Control, VOL. 48, NO. 1, January 2003.
3. Rajeev Alur, Thomas Henzinger: *Reactive Modules.* Proceedings of the 11th IEEE Symposium on Logics in Computer Science (LICS), pages 207-218, 1996.
4. Rajeev Alur, Thomas Henzinger: *Modularity for Timed and Hybrid Systems.* Proceedings of the 8th International Conference on Concurrency Theory (CONCUR 97), LNCS 1243, pp. 74-88, 1997.
5. Nancy Lynch, Roberto Segala, Frits Vaandrager: *Hybrid I/O Automata.* Information and Computation, 185(1):105-157, 2003.
6. Nancy Lynch, Roberto Segala, Frits Vaandrager: *Hybrid I/O Automata Revisited.* in Proceedings of the 4th Hybrid Systems Computation and Control (HSCC) 2001, Rome, Italy, LNCS2034, pages 403-417, March 2001.
7. Mikhail Bernardsky, Raman Sharykin, Rajeev Alur: *Structured Modeling of Concurrent SHS.* Joint Conference on Formal Modeling and Analysis of Timed Systems and Formal Techniques in Real-Time and Fault Tolerant Systems, 2004.
8. Stefan Strubbe: *Compositional Modelling of Stochastic Hybrid Systems.* PhD Thesis, University of Twente, 2005.
9. Sebastien Bornot, Joseph Sifakis: *On the Composition of Hybrid Systems.* Hybrid systems: Computation and Control, Berkeley, April 1998, invited talk, LNCS 1386, pp. 69-83.
10. Joseph Sifakis: *The Compositional Specification of Timed Systems.* CAV 1999 Trento, July 1999.
11. Paulo Tabuada, George Pappas, Pedro Lima: *Compositional Abstractions of Hybrid Control Systems.* Discrete Events Dynamic Systems: Theory and Applications, 14, 203-238, 2004.
12. Martin Abadi, Leslie Lamport: *Composing Specifications.* REX Workshop on Stepwise Refinement of Distributed Systems, Mook, NL, May 1989.
13. Michael Heymann, Feng Lin, George Meyer, Stefan Resmerita: *Analysis of Zeno Behaviors in Hybrid Systems.* in Procedings of the 41st Decision and Control Conference, Las Vegas, NV, Dec. 2001.

14. Holger Hermanns: *Interactive Markov Chains, and the Quest for Quantified Quality.* Lecture Notes in Computer Science 2428, 2002.
15. Aaron Ames, Alessandro Abate and Shankar Sastry: *Sufficient Conditions for the Existence of Zeno Behavior in Hybrid Systems.* Proceedings of the 44th IEEE Conference on Decision and Control, Seville, SP, Dec. 2005.
16. Ekkart Kindler: *Safety and Liveness Properties: A Survey.* Bulletin of the European Association for Theoretical Computer Science, vol. 53, pp. 268–272, 1994.
17. Martin Abadi *et al.*: *Preseving Liveness: Comments on "Safety and Liveness from a Methodological Point of View",* 1991.
18. Bowen Alpern and Fred Schneider: *Defining Liveness.* Information Processing Letters, vol. 21, pp. 181-185, 1985.
19. Alessandro Abate and Ashish Tiwari: *Box Invariance of Hybrid and Switched systems.* Proceedings of the 2nd IFAC Conference on Analysis and Design of Hybrid Systems, Alghero, IT. June 2006.
20. Michael Branicky: *Stability of switched and hybrid systems.* Proceedings of the 33rd Conference on Decision and Control. Dec. 1994.
21. Elena De Santis, Maria Domenica Di Benedetto, Giordano Pola: *Detectability based state space reductions for hybrid systems.* 17th International symposium on Mathematical Theory of Network and Systems, Kyoto, Japan, July 2006.