Proceedings of the 40th IEEE
Conference on Decision and Control
Orlando, Florida USA, December 2001



# Compositional Abstractions of Hybrid Control Systems[1]

Paulo Tabuada[2] George J. Pappas[3] Pedro Lima[2]

[2]Instituto de Sistemas e Robótica
Instituto Superior Técnico
1049-001 Lisboa - Portugal
{tabuada,pal}@isr.ist.utl.pt

[3]Department of EE
University of Pennsylvania
Philadelphia, PA 19104
pappasg@ee.upenn.edu

**Abstract**

Abstraction is a natural way to hierarchically decompose the analysis and design of hybrid systems. Given a hybrid control system and some desired properties, one extracts an abstracted system while preserving the properties of interest. Abstractions of purely discrete systems is a mature area, whereas abstractions of continuous systems is a recent activity. In this paper we present a framework for abstraction that applies to abstract control systems capturing discrete, continuous, and hybrid systems. Parallel composition is presented in a categorical framework and an algorithm is proposed to *construct* abstractions of hybrid control systems. Finally, we show that our abstractions of hybrid systems are compositional.

## 1 Introduction

Networked, embedded systems are compositions of many complicated subsystems. The analysis and design of such systems is currently limited by their complexity. In order to tackle complexity, these systems are organized in a distributed or hierarchical manner. This structure must be exploited in the analysis and design of such systems in order to scale our models, methods, and tools to real life examples. Hybrid systems have been used to model these kinds of large scale complex systems and they usually come equipped with composition operators which compose subsystems in order to form larger systems [4]. This modeling formalism also contains abstraction operators which hide irrelevant details resulting in simpler, higher level models.

The notions of composition and abstraction are mature in theoretical computer science, and, in particular, in the areas of concurrency theory [6] [11], and computer aided verification [5]. This has resulted in formal and very meaningful notions of abstraction. Given a discrete system, an *abstraction* is simply a quotient system that preserves some properties of interest while ignoring detail. Language equivalence, simulation, and bisimulation are established notions of abstraction for discrete systems that preserve properties expressed in various temporal logics. For purely continuous systems, the notions of simulation, and bisimulation had no counterparts. Recently, similar notions were introduced in [7] and this research resulted in automatic constructions of abstractions for linear and nonlinear analytic control systems [8], while characterizing abstracting maps that preserve properties of interest such as controllability. Based on these results, in [10], we took the first steps towards *automatically* constructing abstractions of hybrid systems while preserving timed languages. Even though *only* the continuous part of the system was abstracted, the important property that needed to be preserved in this abstraction was the detectability of the discrete switching conditions. Related but orthogonal work considers purely discrete abstractions of hybrid systems [2, 9].

The similarities between notions of abstraction for discrete, continuous, and hybrid systems immediately raise the question of a more unified theory of abstraction. In this paper, we begin addressing this issue. We start by first considering a more unified and abstract model for control systems. Our abstract control system is inspired by categorical definitions of systems that are as old as [1] and as recent as [11]. We show that purely discrete, continuous, and hybrid systems can be easily captured by our abstract model. Furthermore, at this level of abstraction, one can show many useful properties regarding abstraction or composition that are independent of the discrete, continuous, or hybrid structure of the system. In particular, we use the abstract composition operators defined in [11], and show that system abstraction is compositional. As a result, when abstracting a subsystem of a larger system, we obtain an abstraction of the overall system. We also present an algorithm for the *automatic* abstraction of hybrid systems.

[1]This work was partially performed while the first author was visiting the University of Pennsylvania. The authors would like to thank Esfandiar Haghverdi for extremely stimulating discussions on category theory, and its use for hybrid systems. This research is partially supported by DARPA under grant F33615-00-C-1707, the University of Pennsylvania Research Foundation, and by Fundação para a Ciência e Tecnologia under grant PRAXIS XXI/BD/18149/98.

0-7803-7061-9/01/$10.00 © 2001 IEEE

## 2 Abstract Control Systems

In order to capture continuous, discrete, and hybrid systems under a unified model, we need an abstract definition of control systems. The essence of a control system is reflected into two different aspects: a notion of evolution, and the ability to control the evolution. These two fundamental aspects are captured in the following definition.

**Definition 2.1 (Abstract Control System)** *Let $S$ be a set, $\mathcal{M}$ a monoid and $A$ a fibering relation*[1] *on $S \times \mathcal{M}$ with base space $S$ such that $A_s = \pi_S^{-1}(s)$ is a prefix closed subset of $\mathcal{M}$ containing the identity for every $s \in S$. An abstract control system over $S$ is a map $\Phi : A \rightarrow S$ respecting the monoid structure, that is $\Phi_s : A_s \rightarrow S$ verifies:*

- **Identity:** $\Phi_s(\varepsilon) = s$
- **Semi-group:** $\Phi_s(a_s a_{s'}) = \Phi_{\Phi_s(a_s)}(a_{s'})$

Intuitively, we can think of the set $S$ as the state space, and the fiber bundle $A$, also called in this work a fibering monoid, as the set of possible actions, that depend on the base point. The map $\Phi$ assigns to each point $s \in S$ a function from $A_s$ to $S$ representing all the input choices that can be made at the point $s$. To get a better understanding of the above definition we will see how it applies to three classes of systems.

### 2.1 Discrete Control Systems as Abstract Control Systems

The usual model for discrete control systems are automata however it will be enough to work with transition systems. Let $(Q, \Sigma, \delta)$ be a transition system, where $Q$ is a finite set of states, $\Sigma$ is a finite set of input symbols, and $\delta : Q \times \Sigma \rightarrow Q$ is the transition function. Let us denote by $\Sigma^*$ the set of all finite strings obtained by concatenating elements in $\Sigma$. In particular the empty string $\varepsilon$ also belongs to $\Sigma^*$. With concatenation as a monoid operation, $\Sigma^*$ can be taken as the monoid $\mathcal{M}$. The state space is naturally $S = Q$. The transition function $\delta$ defines a *unique* partial map from $Q \times \Sigma^*$ to $Q$ which is just an abstract control system $\Phi : (S \times \mathcal{M})|_R = A \rightarrow S$, where $R$ is the relation given by $R = \{(s, m) \in S \times \mathcal{M} \ : \ \Phi(s, m) \text{ is defined}\}$.

### 2.2 Continuous Control Systems as Abstract Control Systems

For simplicity of presentation, we consider only time-invariant control systems, although the construction to be presented is generalizable to time varying systems. Let $U$ be the space of admissible inputs. Define the set $U^t$ as:

$$U^t = \{u : [0, t[ \rightarrow U \quad | \quad [0, t[ \subseteq \mathbb{R}_0^+ \} \tag{2.1}$$

[1] We say that $R \subset A \times B$ is a fibering relation with base space $A$ if $\pi_A(R) = A$ where $\pi_A$ is the natural projection on $A$.

An element of $U^t$ is denoted by $u^t$, and represents a map from $[0, t[$ to $U$. Consider now the set $U^*$ which is the disjoint union of all $U^t$ for $0 \leq t < \infty$:

$$U^* = \bigcup_{0 \leq t < \infty} U^t \tag{2.2}$$

The set $U^*$ can be regarded as a monoid under the operation of concatenation, that is if $u^{t_1} \in U^{t_1} \subset U^*$ and $u^{t_2} \in U^{t_2} \subset U^*$ then $u^{t_1} u^{t_2} = u^{t_1 + t_2} \in U^{t_1 + t_2} \subset U^*$ with concatenation given by:

$$u^{t_1} u^{t_2}(t) = \begin{cases} u^{t_1}(t) & \text{if} \quad 0 \leq t < t_1 \\ u^{t_2}(t - t_1) & \text{if} \quad t_1 \leq t < t_1 + t_2 \end{cases}$$

The identity element is given by the empty input, that is $\varepsilon = u^0$. We now show how this monoid is used to describe any smooth control system as an abstract control system. Let $\dot{x} = f(x, u)$ be a smooth control system, where $x \in M$, a smooth manifold and $u \in U$, the set of admissible inputs. Choosing an admissible input trajectory $u^t$, $f(x, u^t)$ is a well defined vector field and as such it induces a flow which we denote by $\gamma_x : [0, t[ \rightarrow M$, such that $\gamma_x(0) = x$. We can then cast any smooth control system in out framework by defining:

$$\begin{aligned} \Phi : M \times U^* &\rightarrow M \\ (x, u^t) &\mapsto \gamma_x(t) \end{aligned} \tag{2.3}$$

It is not difficult to see that $\Phi$ is in fact a well defined abstract control systems since $\Phi(x, \varepsilon) = \gamma_x(0) = x$ and $\Phi(x, u^{t_1} u^{t_2}) = \gamma_x(t_1 + t_2) = \gamma_{\gamma_x(t_1)}(t_2) = \Phi(\Phi(x, u^{t_1}), u^{t_2})$. In general the set of admissible control inputs may change with the point $x$ so that the domain of $\Phi$ will be in fact a fiber bundle over $M$.

### 2.3 Hybrid Control Systems as Abstract Control Systems

Hybrid control systems also fit in the abstract control system framework. The state space of an hybrid control system is usually described as $Q \times M$, where $Q$ is a finite set of states and $M$ a smooth manifold. However it will be convenient to relax this concept and look at the state space as a fiber bundle. Instead of considering the same manifold $M$ for every $q \in Q$ we consider a set of smooth manifolds $X_q$ parameterized by the discrete states, denoted by $X = \{X_q\}_{q \in Q}$. The discrete set $Q$ is thought as the base space, and for each base point $q \in Q$ we attach a fiber $X_q$. A point in $X$ is represented by the pair $(q, x)$. As action monoid we will use the set:

$$\mathcal{M} = \bigcup_{t \in \{1, 2, \ldots, n\}} (U^* \cup \Sigma^*)^{\{1, 2, \ldots, t\}} \tag{2.4}$$

assuming that $U^* \cap \Sigma^* = \{\varepsilon\}$. Let us elaborate on the product operation on $\mathcal{M}$. This operation is defined as the usual concatenation and therefore it requires finite length strings. To accommodate this requirement and

still be able to have an infinite number of concatenations of elements in $U^*$ we proceed as follows. Suppose that we want to show that $\sigma_1 u^{t_1} u^{t_2} \dots u^{t_n} \dots \sigma_2$ belongs to $\mathcal{M}$, where $t_n$ is a convergent series. Instead of regarding each element in the string as an element in $\mathcal{M}$, which would not allow us to define the last concatenation since it would happen after $\infty$ we regard $\sigma_1, \sigma_2$ as elements of $\mathcal{M}$ and $u^{t_1} u^{t_2} \dots u^{t_n} \dots = u^{t'}$ as an element of $U^*$ and consequently as an element of $\mathcal{M}$, where $t' = \lim_{n \longrightarrow \infty} t_n$. This string is then regarded as the map $u : \{1,2,3\} \rightarrow \mathcal{M}$ defined by $u(1) = \sigma_1$, $u(2) = u^{t'}$ and $u(3) = \sigma_3$. The product in $\mathcal{M}$ is then the usual concatenation on reduced strings, that is, strings where all consequent sequences of elements of $U^*$ or $\Sigma^*$ have been replaced by their product in $U^*$ or $\Sigma^*$, respectively. Hybrid control systems are now cast into the abstract control systems framework as:

**Definition 2.2 (Hybrid Control System)** *An hybrid control system $H = (X, A_X, \Phi_X)$ consists of:*

- *The state space $X = \{X_q\}_{q \in Q}$.*
- *A subset $A_X$ of $X \times \mathcal{M}$ defined by $A_X = \{((q,x),m) \in X \times \mathcal{M} : \Phi_X((q,x),m)$ is defined$\}$.*
- *A map $\Phi_X : A_X \rightarrow X$ respecting the monoid structure such that for all $q \in Q$, there is a set $Inv_q \subseteq X_q$ and for all $x \in Inv_q$, $A_{(q,x)} \cap U^* \neq \{\varepsilon\}$ and $\Phi((q,x),u^t) \in Inv_q$ for all $u^t \in A_{(q,x)}$.*

The semantics associated with the evolution from $(q,x)$ governed by $\Phi$ and controlled by $a \in A_{(q,x)}$ is the standard transition semantics of hybrid automata. Suppose that $a = u^{t_1} \sigma_1 \sigma_2 u^{t_2}$, then $\Phi((q,x),a) = (q',x')$ means that the system starting at $(q,x)$ evolves during $t_1$ units of time under continuous input $u^{t_1}$, jumps under input $\sigma_1$ and them jumps again under $\sigma_2$. After the two consecutive jumps, the system evolves under the continuous control input $u^{t_2}$ reaching $(q',x')$ $t_2$ units of time after the last jump. From the hybrid system construction we can clearly extract the purely discrete case ($X_q$ is a singleton and $U_q = \varnothing$) as well as the purely continuous case ($Q$ is a singleton and $\Sigma = \varnothing$).

### 2.4 Control System Abstractions

We now consider simulation relations, and in particular abstractions, between the general systems considered in Definition 2.1. Although for discrete and smooth systems a notion of simulation based on a map between fibering monoids is able to model the relevant concepts and constructions, that will not be the case for hybrid control systems. A map between fibering monoids turns out to be too restrictive and one is forced to look into more general notions of simulation. The link between the fibering monoids will be provided by a *relation*[2] which is general enough for our purposes.

[2]In fact it was by means of a relation that the notion of bisimulation was introduced in [6]

A notion of simulation will involve a relation between fibering monoids that respectes the control structure given by the map $\Phi$. This is formalized as follows:

**Definition 2.3 (Simulation)** *Let $\Phi_X$ and $\Phi_Y$ be two abstract control systems over $X$ and $Y$ with fibering monoids $A_X$ and $A_Y$, respectively. Let $R \subseteq A_X \times A_Y$ be a fibering monoid respecting relation*[3]*. Then $\Phi_Y$ is a simulation of $\Phi_X$ with respect to $R$ or a $R$-simulation if and only if:*

$$\forall_{x \in X} \ (x,y) \in R_B \ \Rightarrow \ \forall_{(x,a_x) \in dom(R)} \ \exists_{(x,a_x,y,a_y) \in R} \ (\Phi_X(x,a_x), \Phi_Y(y,a_y)) \in R_B \qquad (2.5)$$

This definition slightly generalizes the usual notions of morphisms between transition systems as in [11], since we allow the control inputs to depend on the state space and since we use relations instead of functions. This notion of simulation can be reformulated in terms of the notion given in [6]. This is accomplished by embedding the category of abstract control systems in the category of transition systems. It is not difficult to see that abstract control systems and relations satisfying condition (2.5) form a category, that we call the abstract control systems category. The notion of abstraction naturally follows:

**Definition 2.4** *Let $\Phi_X$ and $\Phi_Y$ be abstract control systems over $X$ and $Y$ with fibering monoids $A_X$ and $A_Y$, respectively. If $R \subseteq A_X \times A_Y$ is a fibering monoid respecting relation we say that $\Phi_Y$ is a $R$-abstraction of $\Phi_X$ iff $\Phi_Y$ is a $R$-simulation of $\Phi_X$ and $R$ is a surjective relation.*

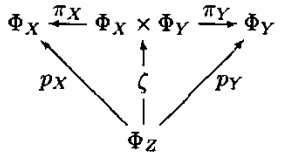
### 2.5 Compositional Abstractions

Following [11] the first step of composition combines two abstract control systems into a single one by forming their product. Given two abstract control systems $\Phi_X : A_X \rightarrow X$ and $\Phi_Y : A_Y \rightarrow Y$ we define their product to be the abstract control system $\Phi_X \times \Phi_Y : (A_X \times A_Y) \rightarrow (X \times Y)$, where the fibers of $(A_X \times A_Y)$ are subsets of the direct product monoid $\mathcal{M}_X \otimes \mathcal{M}_Y$. The trajectories of the product control system consist of all possible combinations of the trajectories of the initial control systems. The product can also be defined in a categorical manner.

**Definition 2.5 (Product of abstract systems)** *Let $\Phi_X : A_X \rightarrow X$ and $\Phi_Y : A_Y \rightarrow Y$ be two abstract control systems. The product of these abstract control systems is a triple $(\Phi_X \times \Phi_Y, \pi_X, \pi_Y)$ where $\Phi_X \times \Phi_Y$ is an abstract control system and $\pi_X \subseteq (X \times Y) \times X$ and $\pi_Y \subseteq (X \times Y) \times Y$ are projection relations*

[3]We say that a relation $R \subseteq F_1 \times F_2$ between fibering monoids is fibering monoid respecting iff satisfies: **Identity:** $(x_1,x_2) \in R_B \ \Rightarrow \ ((x_1,\varepsilon),(x_2,\varepsilon)) \in R$; **Semigroup:** $(m_{x_1}, m_{x_2}), (m'_{x_3}, m'_{x_4}) \in R$ and $m_{x_1} m'_{x_3} \in F_1$ then $(m_{x_1} m'_{x_3}, m_{x_2} m'_{x_4}) \in R$, where $R_B$ is the relation on the base spaces of $F_1$ and $F_2$ indced by $R$.

*such that $\Phi_X$ is a $\pi_X$-simulation of $\Phi_X \times \Phi_Y$, $\Phi_Y$ is a $\pi_Y$-simulation of $\Phi_X \times \Phi_Y$, and for any other triple $(\Phi_Z, p_X, p_Y)$ of this type there is one and only one relation $\zeta \subseteq Z \times (X \times Y)$ such that $\Phi_X \times \Phi_Y$ is a $\zeta$-simulation of $\Phi_Z$, and the following diagram commutes:*

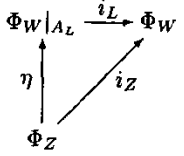



(2.6)

The relations $\pi_X$ and $\pi_Y$ are in fact those induced by the canonical projection maps $\pi_X : X \times Y \to X$, $\pi_Y : X \times Y \to Y$ and the relation $\zeta$ is easily seen to be given by $\zeta = (p_X, p_Y)$. In the product system we capture all possible trajectories of both systems and consequently several non physically meaningful trajectories. One allows for example input trajectories of the form $(\varepsilon, u^t)$ where no time elapses in system $\Phi_X$ and $t$ units of time elapse in system $\Phi_Y$. These trajectories need to be removed from the product system in order to faithfully model a physical system. Another reason to remove transitions from the product system comes from the fact that in the product system, the behavior of one system does not influence the behavior of the other system. Since in general the behavior of a system composed of several subsystems depends strongly on the interaction between the subsystems one tries to capture this interaction by removing undesired evolutions from the product system $\Phi_X \times \Phi_Y$, through the operation of restriction.

Given a subbundle[4] $A_L \subseteq A_W$ we define the restriction of control system $\Phi_W : A_W \to W$ to $A_L$ as a new control system $\Phi_W|_{A_L} : A_L \to L$ which is given by $\Phi_W|_{A_L}(x,a) = \Phi_W(x,a)$ iff $(x,a) \in A_L$ and $\Phi_W(x,a')$ belongs to $L$ for any prefix $a'$ of $a$. If the subbundle $A_L$ has the same base space as $A_W$ but "smaller" fibers, then restriction is modeling synchronization of both systems on the control inputs. If on the other hand the fibers are equal but the base space of $A_L$ is "smaller" then the base space of $A_W$ then both systems are being synchronized on the state space. Synchronization on inputs and states is also captured by the operation of restriction by choosing a subbundle with "smaller" fibers and base space. This operation also admits a categorical characterization.

**Definition 2.6 (Restriction of abstract systems)** *Let $\Phi_W : A_W \to W$ be an abstract control system and let $A_L$ be a subbundle of $A_W$. The restriction of $\Phi_W$ to $A_L$ is a pair $(\Phi_W|_{A_L}, i_L)$ where $\Phi_W|_{A_L}$ is an abstract control system and $i_L \subseteq L \times W$ is an inclusion relation such that $\Phi_W$ is a $i_L$-simulation of $\Phi_W|_{A_L}$, and for any other pair $(\Phi_Z, i_Z)$ of this type with $i_Z(A_Z) = i_L(A_L)$ there is one and only one relation $\eta$ such that $\Phi_W|_L$ is a $\eta$-simulation of $\Phi_Z$, and the following diagram commutes:*




(2.7)

The inclusion relation $i_L$ is in fact the map $i_L : A_L \hookrightarrow A_W$ sending $l \in A_L$ to $i_L(l) = l \in A_W$, and consequently the relation $\eta$ is trivially given by $\eta = i_Z$. With the notions of products and restriction at hand, we can now define a general operation of parallel composition with synchronization.

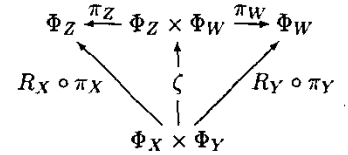
**Definition 2.7 (Parallel Composition)** *Let $\Phi_X : A_X \to X$ and $\Phi_Y : A_Y \to Y$ be two abstract control systems and consider a subbundle $A_L \subseteq A_X \times A_Y$. The parallel composition of $\Phi_X$ and $\Phi_Y$ with synchronization over $A_L$ is the abstract control system denoted by $\Phi_X \|_{A_L} \Phi_Y$ and defined as:*

$$\Phi_X \|_{A_L} \Phi_Y = (\Phi_X \times \Phi_Y)|_{A_L} \tag{2.8}$$

Any theory of abstraction only makes sense if it is compositional. Since the system being analyzed is given by the parallel composition of several smaller subsystems, one can perform abstraction of individual subsystems, resulting in abstractions of the overall system. This will be asserted in the next theorem, but first we need to introduce some notation. Given relations $R_1 \subseteq A_1 \times B_1$, $R_2 \subseteq A_2 \times B_2$ and a subset $L \subseteq A_1 \times A_2$ we define the new relations $R_{1\times 2} = R_1 \times R_2 = \{((a_1,a_2),(b_1,b_2)) \in (A_1 \times A_2) \times (B_1 \times B_2) : (a_1,b_1) \in R_1 \wedge (a_2,b_2) \in R_2\}$ and $R_{1\times 2}|_L = \{((a_1,a_2),(b_1,b_2)) \in R_{1\times 2} \ : \ (a_1,a_2) \in L\}$.

**Theorem 2.8 (Compositionality of Simulations)** *Given abstract control systems $\Phi_X$, $\Phi_Z$ (which is a $R_X$-simulation of $\Phi_X$), $\Phi_Y$, $\Phi_W$ (which is a $R_Y$-simulation of $\Phi_Y$) and the subbundle $A_L \subseteq A_X \times A_Y$, the parallel composition of the simulations $\Phi_Z$ and $\Phi_W$ with synchronization over $R_{X\times Y}(A_L)$ is a $R_{X\times Y}|_{A_L}$-simulation of the parallel composition of $\Phi_X$ with $\Phi_Y$ with synchronization over $A_L$.*

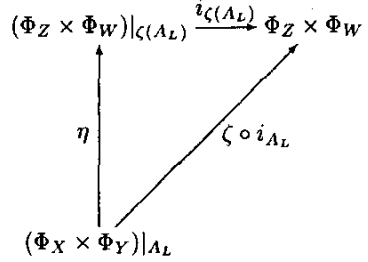
**Proof:** Consider the product system $(\Phi_Z \times \Phi_W, \pi_Z, \pi_W)$ and the triple $(\Phi_X \times \Phi_Y, R_X \circ \pi_X, R_Y \circ \pi_Y)$. By definition of product we know that there is one and only one relation $\zeta$ such that:




commutes and this relation is given by $\zeta = (R_X, R_Y) = R_{X\times Y}$, meaning that $\Phi_Z \times \Phi_W$ is a $R_{X\times Y}$-simulation

[4] A subbundle is understood as a fiber bundle such that the inclusion morphism (in this case a relation) is fiber preserving.

of $\Phi_X \times \Phi_Y$. Consider now the following diagram:




(2.9)

One sees that the unique relation $\eta$ is given by $\eta = \zeta \circ i_{A_L} = R_{X\times Y} \circ i_{A_L}$, that is, $\eta$ is the relation $R_{X\times Y}$ restricted to the subbundle $A_L$. From this we conclude that $\Phi_Z \parallel_{R_{X\times Y}(A_L)} \Phi_W$ is a $R_{X\times Y}|_{A_L}$-simulation of $\Phi_X \parallel_{A_L} \Phi_Y$ as desired. ∎

The above result was stated for parallel composition of two abstract control systems but it can be easily extended to any finite number of abstract control systems. The relevance of the result lies in the fact that, in general, it is much easier to abstract each individual subsystem and by parallel composition obtain an abstraction of the overall system.

## 3 Compositional Abstractions of Hybrid Control Systems

Simulations of hybrid control systems are a simple instantiation of the previously introduced notion of simulation for abstract control systems. However, hybrid control systems usually come equipped with a set of initial conditions which must also be related with the set of initial conditions of its simulation. The proper relation is expressed as follows:

**Definition 3.1 (Simulations of Hybrid Systems)** *Let $H_X = (X_0, X, A_X, \Phi_X)$ and $H_Y = (Y_0, Y, A_Y, \Phi_Y)$ be two hybrid control systems over $X$ and $Y$ respectively and let $R \subseteq A_X \times A_Y$ be a fiber respecting relation. $H_Y$ is a $R$-simulation of $H_X$ iff:*

1. $R_B(X_0) \subseteq Y_0$.
2. $\forall_{x\in X}\,(x,y) \in R_B \Rightarrow \forall_{(x,a_x)\in dom(R)} \exists_{(x,a_x,y,a_y)\in R}$ $(\Phi_X(x,a_x), \Phi_Y(y,a_y)) \in R_B$.

The goal of obtaining algorithmic procedures for computing abstractions guide us to more amenable characterizations of hybrid control systems. A first step in this direction is given by the next proposition (whose proof we omit for space reasons) characterizing hybrid control systems in terms of its generators.

**Proposition 3.2 (Hybrid Generators)** *A set of initial conditions $X_0 \subseteq X$, a finite set of symbols $\Sigma_X$, a family of smooth fiber bundles $\pi_X^q : U_X^q \to X_q$, a partially defined map $\delta_X : X \times \Sigma_X \to X$ and a family of smooth control systems $F_X = \{F_X^q\}_{q\in Q}$, $F_X^q : U_X^q \to TX_q$ defined on a open subset of $X_q$ for each $q \in Q$ uniquely define a hybrid control system $H_X$. The maps $\delta_X$ and $F_X$ are called the discrete and continuous generators of $H_X$, respectively.*

This result tells us that it is enough to work with vector fields and single event jumps, which is how hybrid automata are usualy defined. In the light of this result we will also denote an hybrid control system by the tuple $H_X = (X, X_0, \Sigma_X, U_X, \delta_X, F_X)$. This representation of hybrid control systems will allow constructive methods to generate abstractions by combining discrete and continuous abstraction methodologies.

We have already introduce all the required tools to extract an abstraction of any given hybrid control system $H_X$ with respect to some classes of relations. These relations will aggregate several states from $H_X$ to its abstractions, in particular we abstract continuous states to discrete ones. This feature will be fundamental to reduce the complexity of hybrid control systems since by aggregating continuous states into discrete ones we are also trading continuous dynamics by discrete dynamics thereby considerably simplifying the analysis and synthesis processes. In this paper we will only consider *admissible* relations defined as:

**Definition 3.3 (Admissible Relation)** *Given a hybrid control system $H_X$ and:*

- *A finite covering $\Gamma_q = \{\Gamma_q^i\}_{i\in I}$ by pairwise disjoint sets of $dom(F_X^q)$ for every $q \in Q$ inducing a well defined transition system across adjacent covering sets*[5].
- *A family of smooth surjective fiber preserving submersions $\varphi_{qp} : U_X^q \to U_Y^p$.*
- *A trivial bundle surjective map $\varphi_D = (\phi_Q, \phi_\Sigma)$, $\phi_Q : X \to P$ and $\phi_\Sigma : \Sigma_X \to \Sigma_Y$.*

*the induced admissible relation $R \subseteq A_X \times A_Y$ is defined for:*

- *Continuous flows remaining inside a single covering set and starting on a interior point:*

$$\big((q,x,u_x^t),(\phi_Q(q,x),\varphi_{q\phi_Q(q,x)}(x,u_x^t))\big) \in R$$
$$\text{iff} \quad \forall_{0\le t'<t} \quad \Phi_X(q,x,u_x^{t'}) \in \Gamma_q(x)$$
$$\wedge \quad (q,x) \in int(\Gamma_q(x))$$

- *Continuous flows remaining inside a single covering set and starting on a boundary point:*

$$\big((q,x,u_x^t),(p_i,\varphi_{qp_i}(x,u_x^t))\big) \in R$$
$$\big((q,x,\varepsilon),(p_j,y_j,\varepsilon)\big) \in R \qquad \forall_{j\in K}$$
$$\big((q,x,\varepsilon),(p_j,y_j,\sigma_{p_jp_i})\big) \in R \qquad \forall_{j\in K, j\neq i}$$

*for $\sigma_{p_jp_i} \in \Sigma_Y$ such that $\Phi_Y(p_j,y_j,\sigma_{p_jp_i}) = (p_i,y_i)$ iff the following holds:*

$$(q,x) \in \cap_{k\in K} cl(\Gamma_q^k) \;\wedge\; \forall_{0<t'<t} \quad \Phi_X(q,x,u_x^{t'}) \in \Gamma_q^i$$
$$\wedge\; \phi_Q|_{\Gamma_q^k} = p_k \;\wedge\; \phi_{qp_k}(x) = y_k \quad \forall_{k\in K}$$

[5] We do not provide more specific conditions since this is still an open problem. Sufficient conditions involving subanalytic stratifications can be found, for example, in [3].

- *Discrete jumps:*

$$((q,x,\sigma),(\phi_Q(q,x),\phi_{q\phi_Q(q,x)}(x),\phi_\Sigma(\sigma))) \in R$$

Admissible relations allow us to effectively compute abstractions of hybrid control systems. A conceptual algorithm may be formulated as follows:

**Algorithm 3.4 (Abstracting Algorithm)**
**Input data:** $H_X = (X_0, X, \Sigma_X, U_X, \delta_X, F_X)$
$R \subseteq A_X \times A_Y$

1. $Y := R_B(X)$
2. $Y_0 := R_B(X_0)$
3. $\Sigma_Y := \phi_\Sigma(\Sigma_X) \cup \{\sigma : \exists((q,x,\varepsilon),(p,y,\sigma)) \in R\}$
4. $U_Y = \{U_Y^p\}_{p\in P}\ U_Y^p = \varphi_{pq}(U_X^q)$
5. $J = \{(p,y,\sigma_{pp'},p',y') : \exists(q,x) \in \cap_{k\in K} cl(\Gamma_q^k)\ \exists u \in U_X^q(x)$ *such that* $F_X^q(u)$ *is transversal to the boundary of* $\Gamma_q^i$, *points to* $\Gamma_q^i$, $((q,x),(p,y)) \in R_B$, $p \neq \phi_Q|_{\Gamma_q^i}$ *and* $((q,x),(p',y')) \in R_B$, $p' = \phi_Q|_{\Gamma_q^i}\}$
6. $\delta_Y := (\phi_Q, \phi_{q\phi_Q}, \phi_\Sigma, \phi_Q, \phi_{q\phi_Q})(\delta_X) \cup J$ *where* $\delta_X$ *is regarded as the set* $\delta_X \subseteq X \times \Sigma_X \times X$.
7. $F_Y^p :=$ *is the* $\varphi_{qp}$*-abstraction of* $F_X^q$ *with domain* $\varphi_{qp}(dom(F_X^q \cap \Gamma_q(x)))$.

**Output data:** $H_Y = (Y_0, Y, \Sigma_Y, U_Y, \delta_Y, F_Y)$

Intuitively the above algorithm can be described as follows. Steps 1 and 2 simply define $Y$ and $Y_0$ as the image under $R_B$ of $X$ and $X_0$, respectively. In step 3 the set of labels $\Sigma_Y$ is computed as the image under $\phi_\Sigma$ of $\Sigma_X$ and all the symbols $\sigma_{pp'}$ created when the continuous flows crosses the boundary between adjacent covering sets. In step 4 the continuous control bundle is computed as the image of $U_X^q$ under each map $\varphi_{qp}$. In step 5 the set $J$ is computed to be used on the next step. Step 6 determines $\delta_Y$ in a way that can be described as follows: for every transition $(q,x) \xrightarrow{\sigma} (q',x')$ defined by $\delta_X$ there will be a transition $(\phi_Q(q,x), \phi_{q\phi_Q(q,x)}(x)) \xrightarrow{\phi_\Sigma(\sigma)} (\phi_Q(q',x'), \phi_{q'\phi_Q(q',x')}(x'))$ expressed by the set $(\phi_Q, \phi_{q\phi_Q}, \phi_\Sigma, \phi_Q, \phi_{q\phi_Q})(\delta_X)$, where $\delta_X$ is regarded as a subset of $X \times \Sigma_X \times X$. Furthermore every time a continuous flow crosses the boundary between adjacent covering sets, the required discrete transitions are captured by the set $J$. Finally in the last step the continuous generator of $H_Y$ is obtained from the continuous generator of $H_X$ by the methods described in [7, 8].

The above algorithm does compute a simulation of $H_X$ as asserted in the next theorem whose proof we were forced to omit due to lack of space:

**Theorem 3.5** *Let* $H_X$ *be an hybrid control system over* $X$ *and* $R \subseteq A_X \times A_Y$ *an admissible relation. Then hybrid control system* $H_Y$ *obtained through Algorithm 3.4 is a* $R$*-abstraction of* $H_X$.

## 4 Conclusions and Future Work

We have considered an unified framework for a general class of control systems that captures discrete, continuous and hybrid control systems. In this framework we presented notions of abstraction and parallel composition with synchronization. It was shown that these notions are compatible, that is, the composition of abstractions is an abstraction of the composition. These notions were then instantiated for hybrid control systems where a concrete algorithm was presented to automatically extract abstractions.

Future research will consider several other important properties of hybrid systems. For example, it is crucial to determine when abstractions of non-zeno hybrid control systems are non-zeno. These issues are currently under investigation as well as design methodologies that take advantage of the hierarchical and compositional structure of hybrid control systems.

### References


[1] M.A. Arbib and E. Manes. Machines in a category : An expository introduction. *SIAM Review*, 16(2):163–192, April 1974.

[2] P.E. Caines and Y.J. Wei. Hierarchical hybrid control systems: A lattice theoretic formulation. *IEEE Transactions on Automatic Control : Special Issue on Hybrid Systems*, 43(4):501–508, April 1998.

[3] Gerardo Lafferriere, George J. Pappas, and Shankar Sastry. Subanalytic stratifications and bisimulations. In T. Henzinger and S. Sastry, editors, *Hybrid Systems : Computation and Control*, volume 1386 of *Lecture Notes in Computer Science*, pages 205–220. Springer Verlag, Berlin, 1998.

[4] N. Lynch, R. Segala, F. Vaandrager, and H.B. Weinberg. Hybrid I/O automata. In *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 496–510. Springer-Verlag, 1996.

[5] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety.* Springer Verlag, New York, 1995.

[6] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[7] George J. Pappas, Gerardo Lafferriere, and Shankar Sastry. Hierarchically consistent control systems. *IEEE Transactions on Automatic Control*, 45(6):1144–1160, June 2000.

[8] George J. Pappas and Slobodan Simic. Consistent hierarchies of nonlinear abstractions. In *Proceedings of the 39th IEEE Conference in Decision and Control.* Sydney, Australia, December 2000.

[9] J. Raisch and S.D. O'Young. Discrete approximations and supervisory control of continuous systems. *IEEE Transactions on Automatic Control : Special Issue on Hybrid Systems*, 43(4):569–573, April 1998.

[10] Paulo Tabuada and George J. Pappas. Hybrid abstractions that preserve timed languages. In *Hybrid Systems : Computation and Control*, volume 2034 of *Lecture Notes in Computer Science.* Springer Verlag, 2001.

[11] Glynn Winskel and Mogens Nielsen. *Handbook of Logic and Foundations of Theoretical Computer Science*, chapter Models for Concurrency. Oxford University Press, 1994.