Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Hybrid Dynamical Systems with Hybrid Inputs: Definition of Solutions and Applications to Interconnections

Pauline Bernard[1] | Ricardo G. Sanfelice[2]

[1]Control and Systems Center, MINES ParisTech, PSL Research University, 60 Boulevard Saint-Michel, 75006 Paris, France

[2]Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064, USA

Correspondence
*Pauline Bernard, Email: pauline.bernard@mines-paristech.fr

Summary

In this paper, we define solutions for hybrid systems with pre-specified hybrid inputs. Unlike previous work where solutions and inputs are assumed to be defined on the same domain a priori, we consider the case where intervals of flow and jump times of the input are not necessarily synchronized with those of the state trajectory. This happens in particular when the input is the output of another hybrid system, for instance in the context of observer design or reference tracking. The proposed approach relies on reparametrizing the jumps of the input in order to write it on a common domain. The solutions then consist of a pair made of the state trajectory and the reparametrized input. Our definition generalizes the notions of solutions of continuous-time and discrete-time systems with inputs. We provide an algorithm that automatically performs the construction of solutions for a given hybrid input. In the context of hybrid interconnections, we show how the solutions of the individual systems can be linked to the solutions of a closed-loop system. Examples illustrate the notions and the proposed algorithm.

KEYWORDS:
hybrid systems, interconnections, modeling, observers, hybrid inputs

1 | INTRODUCTION

1.1 | Background

A significant part of control theory consists of studying systems with inputs, whether it be for tracking control, output regulation, or estimation. In fact, dynamical properties relating inputs, outputs, and the state of single and multiple, interconnected systems are widely used for analysis and design of feedback control systems, which are naturally interconnected. Notions such as input-to-state stability (ISS) [1,2] have been rendered useful to study interconnection of continuous-time systems via small gain theorems. Extensions of small gain theorems to discrete-time, switched, and hybrid systems are available in [3], [4], and [5], respectively. Similarly, the so-called output-to-state stability (OSS) notion is convenient to bound the solutions by a function of the output of the system [6]; see also its extension to hybrid systems in [7]. Combining the ideas in the ISS and OSS notions, input-output-to-state stability (IOSS) provides bounds that depend on the inputs and outputs of the single and multiple systems; see [1,8,9]. The fact that these notions relate (functions of) the state to (functions of) the inputs and the state of a system make it very appealing for the study of interconnections. Indeed, under the appropriate assumptions, interconnections of systems that individually enjoy properties like ISS and IOSS give rise to closed-loop systems with similar properties, in particular, asymptotic stability.

[0]This research has been partially supported by the National Science Foundation under Grant no. ECS-1710621 and Grant no. CNS-1544396, by the Air Force Office of Scientific Research under Grant no. FA9550-16-1-0015, Grant no. FA9550-19-1-0053, and Grant no. FA9550-19-1-0169, and by CITRIS and the Banatao Institute at the University of California.

As the cited literature indicates, results for the study of interconnections of continuous-time and discrete-time systems are for the case when solutions to the systems are defined for all time, namely, for all continuous time $t \in [0, \infty)$ and for all discrete time $k \in \{0, 1, 2, \dots\}$, respectively. For these classes of systems, such notions of solutions also apply to their interconnections, due to the solution to each system being defined for all (continuous or discrete) time. On the other hand, when solutions are defined over a bounded horizon (or domain) then solutions to the interconnection can only be defined over the smallest such horizon, but, besides such technicality, interconnections of continuous-time or of discrete-time systems does not raise any critical problems in what pertains to definition of solutions. On the other hand, defining solutions to hybrid systems – with or without hybrid inputs – is much more challenging, due to the fact that, in general, solutions to a hybrid system do not have the same domain of definition. For instance, the notion of solution employed in [10] and in [11] uses both continuous time $t \in [0, \infty)$ and a discrete counter $j \in \{0, 1, 2, \dots\}$ to parameterize the evolution of the state (and input) trajectories defining a solution. In this setting, a solution that evolves continuously (or, equivalently, flows) for $t_1 > 0$ seconds at which time instant it jumps, and then flows until $t_2 > t_1$ seconds, and proceeding in this way, continues to flow up to $t_{j+1} > t_j$, and so forth, is defined on the set

$$([0, t_1] \times \{0\}) \cup ([t_1, t_2] \times \{1\}) \cup \dots \cup ([t_j, t_{j+1}] \times \{j\}) \cup \dots$$

which is a particular subset of $[0, \infty) \times \{0, 1, 2, \dots\}$. Due to such parameterization of solution, in principle, the domain of definition of the solutions to each hybrid system within an interconnection is not the same. Furthermore, when inputs play a role, the domain of definition of the input may not necessarily match that of the resulting state trajectory. Some of the intricacies in defining solutions to interconnections of hybrid systems are discussed in [12]. A particularly extreme case is when one of the systems in the interconnection has a solution that only evolves continuously (or, equivalently, only flows) and another system has a solution that only evolves discretely (or, equivalently, only jumps), in which case it is not obvious how to define a solution to the interconnection due to the difference on the domains. In previous works involving hybrid systems with inputs, the notion of solution assumes that the domain of the input and of the state trajectory are the same; see, e.g., [5,13,9]. In the case of state feedback, namely, when the input is a function of the state, the input inherits the domain of the state trajectory and the assumption made in the cited references is justified. It is also justified when designing a controller or an observer for a hybrid (or impulsive) system with jump times that are synchronized with the plant [14,15,16,17], and assumed to be known. In those cases, the definition of solutions is straightforward.

1.2 | Motivation

As motivated in Section 1.1, it is restrictive to assume that the domain of the individual solution to each system in an interconnection of hybrid systems is the same. The main challenge is that the domain of the (hybrid) input to each system in such an interconnection is not known a priori, due to typically being a function of the output of another hybrid system. This fact prevents one from assuming (as naturally done for continuous-time and for discrete-time systems) that the domain of the input and of the state trajectory coincide. In some cases, like when the input is a purely continuous-time signal or a purely discrete-time signal, one can actually redefine the input on the domain of the state trajectory, leading to matching domains. However, as said above, such a "pre-processing" of the input cannot be applied to general interconnections of hybrid systems, as it requires altering the domain of the output of another hybrid system. As pointed out in [12] such a modification is far from trivial, and serious difficulties emerge when the jumps of the system are not synchronized with those of the input, leading to very important questions yet to be answered:

- Assume a hybrid system is flowing and its input jumps before the state reaches its jump set: under which conditions should we allow the state to jump and continue evolving, and how should this jump be defined?
- Now, conversely, assume that the state of the system reaches its jump set and cannot continue flowing, while the input is such that it can continue to flow: do we stop the solution or do we allow the system to jump and the input to continue flowing afterwards?
- Combining those two questions, consider a series interconnection/cascade of hybrid systems: how to define a unified notion of solution if the jumps of both systems do not occur at the same time?

These problems appear, for instance, in the context of reference tracking when the reference is a hybrid trajectory. In [18], the reference is a trajectory of the system itself and the problem of reconciling the domains is done by designing an extended "closed-loop" system which naturally puts the reference and the system on the same domains. Similarly, when studying incremental stability for hybrid systems, trajectories with different domains need to be compared and they are typically brought on the same domain thanks to an extended system[19]. The issues mentioned above also arise in the context of observer design (and, more generally, output-feedback), where the input of the hybrid observer is the output of the hybrid plant we want to observe. In[20], the analysis is possible using tools for autonomous hybrid systems thanks to a timer which is used to model the jumps of the input and by building a closed-loop system whose jumps are solely triggered by the timer.

1.3 | Contributions

In this paper, we make the following main contributions:

- *Definition of solutions to hybrid systems with hybrid input*: in Section 2, we propose a novel definition for solutions to hybrid systems when the input is a hybrid arc with its own domain, which does not necessarily match the one of the produced state trajectory. The proposed approach relies on reparametrizing the jumps of the input in order to write it on a common domain with the state trajectory. The solutions then consist of a pair made of the state trajectory and the reparametrized input. Our definition generalizes the notions of solutions of continuous and discrete systems with inputs.
- *Algorithm for the construction of solutions*: we provide in Section 3 an algorithm that automatically performs the construction of solutions for a given hybrid input. We discuss its numerical implementation and the consequences of numerical errors on the definition of solutions.
- *Application to interconnection of hybrid systems and link with closed-loop dynamics*: in the particular case of series and feedback interconnections between two hybrid systems, we investigate in Section 4 the link between the solutions obtained from our definition, to those of an appropriately defined closed-loop system, crucial for Lyapunov-based designs.

All of the proposed notions are illustrated on examples. In particular, we show how our definition enables to define a hybrid observer for a hybrid plant, and provide a sufficient condition for observer design via a closed-loop system in Section 4.1.

2 | SOLUTIONS TO HYBRID DYNAMICAL SYSTEMS WITH INPUTS

For starters, the definition of a solution to a continuous-time system with inputs of the form $\dot{x} = f(x,u)$ requires the following data: an initial state x_0 and an input signal $t \mapsto u(t)$ (typically satisfying basic regularity properties). Then, a solution to the system is typically given by an absolutely continuous function $t \mapsto \phi(t)$ such that $\phi(0) = x_0$ and $\dot{\phi}(t) = f(\phi(t), u(t))$ is satisfied on the domain of definition of u and ϕ. Those domains typically coincide unless ϕ terminates before u, in which case the domain of u is simply truncated. A notion of solution for discrete-time systems with inputs can be defined similarly.

As pointed out in Section 1, the definition of a solution to a hybrid system with inputs is more intricate when we do not rely on the assumption that the domain of the input and of the state trajectory coincide. In this section, we define a notion of solution for hybrid systems with a hybrid arc as input. Due to the likely mismatch between the jump times of the given input u and of the actual state trajectory ϕ to be generated, the proposed notion jointly parametrizes u and ϕ in what we refer to as a j-reparametrization.

We first recall the following definitions and notation. For more details about those definitions, the reader is referred to[21].

Definition 1 (hybrid time domain)**.** A set $E \subseteq \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a hybrid time domain if for each $(T', J') \in E$, the truncation $E \cap \left([0,T'] \times \{0,1,\dots,J'\}\right)$ can be written as $\bigcup_{j=0}^{J-1} \left([t_j, t_{j+1}], j\right)$ for some finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \dots \leq t_J$ and $J \in \mathbb{N}$.

Definition 2 (hybrid arc)**.** A function $\phi : \operatorname{dom}\phi \to \mathbb{R}^n$ is a hybrid arc if $\operatorname{dom}\phi$ is a hybrid time domain and for each $j \in \mathbb{N}$, $t \mapsto \phi(t,j)$ is locally absolutely continuous on $\{t : (t,j) \in \operatorname{dom}\phi\}$.

Notation We denote by $\mathbb{R}$ (resp. $\mathbb{N}$) the set of real numbers (resp. integers), and $\mathbb{R}_{\geq 0} := [0,+\infty)$, $\mathbb{R}_{>0} =: (0,+\infty)$, and $\mathbb{N}_{>0} := \mathbb{N} \setminus \{0\}$. For a set S, $\operatorname{cl}(S)$ will denote its closure, $\operatorname{int}(S)$ its interior and $\operatorname{card} S$ its cardinality (possibly infinite). We denote $\subsetneq$ for a strict inclusion and $\subseteq$ for a nonstrict inclusion. If $S \subseteq \mathbb{R}^p$, we define the distance of $z \in \mathbb{R}^p$ to S by

$$|z|_S = \inf_{z' \in S} |z - z'| .$$

For a hybrid arc $(t,j) \mapsto \phi(t,j)$ defined on a hybrid time domain $\operatorname{dom}\phi$, we denote $\operatorname{dom}_t \phi$ (resp. $\operatorname{dom}_j \phi$) its projection on the time (resp. jump) axis, and for a positive integer j, $t_j(\phi)$ the time stamp associated to jump j (i.e., the only time satisfying $(t_j(\phi), j) \in \operatorname{dom}\phi$ and $(t_j(\phi), j-1) \in \operatorname{dom}\phi$), and $\mathcal{I}_j(\phi)$ the largest interval such that $\mathcal{I}_j(\phi) \times \{j\} \subseteq \operatorname{dom}\phi$. We define also $\mathcal{T}(\phi) = \{t_j(\phi) \; : \; j \in \operatorname{dom}_j \phi \cap \mathbb{N}_{>0}\}$ as the set of jump times, $T(\phi) = \sup \operatorname{dom}_t \phi \in \mathbb{R}_{\geq 0} \cup \{+\infty\}$ the maximal time of the domain, $J(\phi) = \sup \operatorname{dom}_j \phi \in \mathbb{N} \cup \{+\infty\}$ the total number of jumps, and, for a time t in $\mathbb{R}_{\geq 0}$, $\mathcal{J}_t(\phi) = \{j \in \mathbb{N}_{>0} \; : \; t_j(\phi) = t\}$ the set of jump counters associated to the jumps occurring at time t. It follows that $\operatorname{card} \mathcal{J}_t(\phi)$ is the number of jumps of ϕ occurring at time t.

2.1 | j-reparametrization of hybrid arcs

We define a j-reparametrization of a hybrid arc as follows.

Definition 3. Given a hybrid arc ϕ, a hybrid arc ϕ^r is a *j-reparametrization* of ϕ if there exists a function $\rho : \mathbb{N} \to \mathbb{N}$ such that

$$\rho(0) = 0 \qquad , \qquad \rho(j+1) - \rho(j) \in \{0, 1\} \quad \forall j \in \mathbb{N} \tag{1}$$

and

$$\phi^r(t,j) = \phi(t, \rho(j)) \qquad \forall (t,j) \in \operatorname{dom}\phi^r \; . \tag{2}$$

The hybrid arc ϕ^r is a *full j-reparametrization* of ϕ if

$$\operatorname{dom}\phi = \bigcup_{(t,j)\in \operatorname{dom}\phi^r} (t, \rho(j)) \; , \tag{3}$$

or, equivalently, $\operatorname{dom}_t \phi = \operatorname{dom}_t \phi^r$ and $J(\phi) = \rho(J(\phi^r))$. We will say that ρ is a *j-reparametrization map* from ϕ to ϕ^r.

In other words, ϕ^r takes at each time t the same values as ϕ, but maybe associated to a different jump index, because ϕ^r may have trivial jumps added to its domain. If the whole domain of ϕ is spanned by ϕ^r, the reparametrization is said to be full. Indeed, (3) says that $\operatorname{dom}\phi$ is the image of $\operatorname{dom}\phi^r$ by the map

$$(t,j) \mapsto (t, \rho(j)) \; .$$

Example 1. Consider the hybrid arc ϕ defined on $\operatorname{dom}\phi = \mathbb{R} \times \{0\}$ by

$$\phi(t,j) = t \quad \forall (t,j) \in \operatorname{dom}\phi \; ,$$

and ϕ^r defined on $\operatorname{dom}\phi^r = \{0\} \times \mathbb{N}$ by

$$\phi^r(t,j) = 0 \quad \forall (t,j) \in \operatorname{dom}\phi^r \; .$$

The hybrid arc ϕ^r is a j-reparametrization of ϕ with reparametrization map $\rho(j) = 0$ for all $j \in \mathbb{N}$. However, it is not a full reparametrization of ϕ because all of its domain is not spanned.

Now take ϕ defined on $\operatorname{dom}\phi = ([0,1] \times \{0\}) \cup ([1,2] \times \{1\})$ by

$$\phi(t,j) = t - j \quad \forall (t,j) \in \operatorname{dom}\phi \; .$$

In other words, ϕ flows for $t \in [0,1]$ from 0 until reaching 1, then jumps back to 0, and flows again for $t \in [1,2]$. Consider ϕ^r defined on $\operatorname{dom}\phi^r = ([0,1/2] \times \{0\}) \cup ([1/2,1] \times \{1\}) \cup ([1,2] \times \{2\})$ by

$$\phi^r(t,j) = \begin{cases} t & \forall (t,j) \in [0,1/2] \times \{0\} \cup ([1/2,1] \times \{1\}), \\ t-1 & \forall (t,j) \in [1,2] \times \{2\} \end{cases}$$

Then, it is easy to check that ϕ^r is a full j-reparametrization of ϕ with ρ such that $\rho(0) = 0$, $\rho(1) = 0$, $\rho(2) = 1$.

Actually, given ϕ, an infinite number of reparametrizations can be obtained by limiting the domain or adding trivial fictitious jumps, by changing ρ. △

2.2 | Solutions to hybrid systems with hybrid inputs

Consider the hybrid system

$$\mathcal{H} \begin{cases} \dot{x} \in F(x,u) & (x,u) \in C \\ x^+ \in G(x,u) & (x,u) \in D \end{cases} \qquad , \qquad y = h(x,u) \tag{4}$$

with state x taking values in $\mathbb{R}^{d_x}$, input u taking values in $\mathbb{R}^{d_u}$, flow map $F : \mathbb{R}^{d_x}\times\mathbb{R}^{d_u} \rightrightarrows \mathbb{R}^{d_x}$, jump map $G : \mathbb{R}^{d_x}\times\mathbb{R}^{d_u} \rightrightarrows \mathbb{R}^{d_x}$, flow set $C \subseteq \mathbb{R}^{d_x} \times \mathbb{R}^{d_u}$ and jump set $D \subseteq \mathbb{R}^{d_x} \times \mathbb{R}^{d_u}$. We adopt the following definition.

Definition 4. Consider a hybrid arc u. A pair $\phi = (x, u^r)$ is a solution to $\mathcal{H}$ with input u and output y if

1) $\operatorname{dom} x = \operatorname{dom} u^r (= \operatorname{dom}\phi)$

2) u^r is a j-reparametrization of u with reparametrization map ρ_u, and with also $\operatorname{card}\mathcal{J}_{T(u)}(\phi) = \operatorname{card}\mathcal{J}_{T(u)}(u)$ if this reparametrization is full.

3) for all $j \in \mathbb{N}$ such that $\mathcal{I}_j(\phi)$ has nonempty interior,

$$(x(t,j), u^r(t,j)) \in C \quad \forall t \in \operatorname{int}\mathcal{I}_j(\phi)$$
$$\dot{x}(t,j) \in F(x(t,j), u^r(t,j)) \quad \text{for a.a. } t \in \mathcal{I}_j(\phi)$$

4) for all $t \in \mathcal{T}(\phi)$, denoting $j_0 = \min\mathcal{J}_t(\phi)$ and $n_u = \operatorname{card}\ \mathcal{J}_t(u)$, we have

   a) for all $j \in \mathcal{J}_t(\phi)$ such that $j < j_0 + n_u$, we have $\rho_u(j) = \rho_u(j-1)+1$, and:
   if $j = j_0$ and $t > 0$,
   - $(x(t, j_0 - 1), u^r(t, j_0 - 1)) \in C \cup D$
   - $x(t, j_0) \in G^0_e(x(t, j_0 - 1), u^r(t, j_0 - 1))$

   else
   - $(x(t, j - 1), u^r(t, j - 1)) \in \operatorname{cl}(C) \cup D$
   - $x(t, j) \in G_e(x(t, j - 1), u^r(t, j - 1))$

   with
$$G^0_e(x,u) = \begin{cases} x & \text{if } (x,u) \in C \setminus D \\ G(x,u) & \text{if } (x,u) \in D \setminus C \\ \{x, G(x,u)\} & \text{if } (x,u) \in D \cap C \end{cases} \quad , \quad G_e(x,u) = \begin{cases} x & \text{if } (x,u) \in \operatorname{cl}(C) \setminus D \\ G(x,u) & \text{if } (x,u) \in D \setminus \operatorname{cl}(C) \\ \{x, G(x,u)\} & \text{if } (x,u) \in D \cap \operatorname{cl}(C) \end{cases}$$

   b) for all $j \in \mathcal{J}_t(\phi)$ such that $j \geq j_0 + n_u$, we have $\rho_u(j) = \rho_u(j-1)$ and
   - $(x(t, j - 1), u^r(t, j - 1)) \in D$
   - $x(t, j) \in G(x(t, j - 1), u^r(t, j - 1))$

5) for all $(t, j) \in \operatorname{dom}\phi$,

$$y(t,j) = h(x(t,j), u^r(t,j)) \ .$$

The solution ϕ is said to be *maximal* if there does not exist any other solution $\tilde{\phi}$ such that

$$\operatorname{dom}\phi \subset \operatorname{dom}\tilde{\phi} \quad , \qquad \tilde{\phi}(t,j) = \phi(t,j) \quad \forall (t,j) \in \operatorname{dom}\phi \ .$$

The set of maximal solutions to $\mathcal{H}$ initialized in $\mathcal{X}_0$ with input u is denoted $\mathcal{S}_{\mathcal{H}}(\mathcal{X}_0; u)$. △

Conditions 1) and 2) say that u^r is a j-reparametrization of u that is defined on the same domain as x, and that when the whole domain of u is spanned (namely, u^r is a full reparametrization u), the solution stops evolving whenever u does. Indeed, in that case, by Definition 3, $\operatorname{dom}_t \phi = \operatorname{dom}_t u$ (in particular $T(\phi) = T(u)$), and if $T(u) \in \operatorname{dom}_t \phi$, the extra condition $\operatorname{card}\mathcal{J}_{T(u)}(\phi) = \operatorname{card}\mathcal{J}_{T(u)}(u)$ says that ϕ jumps as many times as u at its final time, similarly to solutions of discrete systems with input.

At a time t where the input does not jump ($n_u = 0$), x can jump according to its own jump map G if ϕ is in D by Condition 4b). In that case, u^r contains a trivial jump, namely for all $j \in \mathcal{J}_t(\phi)$,

$$u^r(t,j) = u^r(t,j-1) \quad , \quad \rho_u(j) = \rho_u(j-1) \ .$$

On the other hand, at a time t where the input jumps, Condition 4a) says that:

- at the first jump if $t > 0$, ϕ must be in $C \cup D$ and x is reset either trivially (via the identity) or to a point in $G(x, u)$ according to G^0_e.
- for the remaining jumps of u, or if $t = 0$, those conditions are relaxed with G_e, replacing C by $\operatorname{cl}(C)$.

After all the jumps of u have been processed, ϕ can carry on jumping if it is in D, with x reset to a point of $G(x,u)$ and recording trivial jumps in u^r according to Condition 4b).

The difference between G_e^0 and G_e in Condition 4a) is that x is forced to jump according to G if ϕ is in $D \setminus C$ instead of $D \setminus \mathrm{cl}(C)$. This stricter condition at the first jump of u after an interval of flow is to avoid the situation where ϕ would leave C after flow and then be allowed to flow again from the same point after the jump of u; namely it prevents flows through a hole of C. This condition is already enforced when the input does not jump ($n_u = 0$) by conditions 3) and 4b). In other words, if ϕ leaves C after an interval of flow, it either jumps according to G if it is in D or dies. Hence the condition that ϕ should be in $C \cup D$ instead of $\mathrm{cl}(C) \cup D$ at the first jump of u. On the other hand, for the remaining jumps of u or at $t = 0$, there is no reason to force x to jump with G on $\mathrm{cl}(C) \setminus C$ since x could possibly flow into C. That is why G_e^0 is relaxed into G_e. This distinction disappears if C is closed. Note that more generally, the solution stops if ϕ leaves $\mathrm{cl}(C) \cup D$.

Remark 1. Condition 4) imposes that at a given time, u performs all its jumps consecutively and right away. This choice is important because it determines which value of u is used in the jump map of x. In particular, it enables to recover the definition of solutions of discrete systems with input if $F \equiv \emptyset$ and $C = \emptyset$. Not forcing the jumps of u to be processed right away would lead to a richer set of solutions where x and u jump either simultaneously or not, and with any ordering. In that case, Conditions 4) would be replaced by :

4') for all $t \in \mathcal{T}(\phi)$ and for all $j \in \mathcal{J}_t(\phi)$,

$$\text{either} \quad \begin{cases} (x(t,j-1), u^r(t,j-1)) \in \mathrm{cl}(C) \cup D \\ x(t,j) \in G_e(x(t,j-1), u^r(t,j-1)) \\ \rho_u(j) = \rho_u(j-1) + 1 \end{cases} \quad \text{or} \quad \begin{cases} (x(t,j-1), u^r(t,j-1)) \in D \\ x(t,j) \in G(x(t,j-1), u^r(t,j-1)) \\ \rho_u(j) = \rho_u(j-1) \end{cases},$$

with $\mathrm{cl}(C)$ replaced by C for $j = j_0$ if $t > 0$. With this alternate definition, it would no longer make sense to require $\mathrm{card}\, \mathcal{J}_{T(u)}(\phi) = \mathrm{card}\, \mathcal{J}_{T(u)}(u)$ at the boundary of the time domain in Condition 2), which would be simplified into

2') u^r is a j-reparametrization of u with reparametrization map ρ_u.

This richer set of solutions is particularly relevant when several jumps having a common time stamp represent in fact jumps occurring very close in time. In this case, we do not know if the jump of u truly happens before or after a possible jump of x, and it makes sense to take any value of u at that time in the jump map of x. △

Remark 2. Another way of building solutions to a hybrid system with a hybrid input u would be to look for solutions that jump whenever u jumps. In other words, a jump of u would force a jump of the state according to its own jump map. However, this would significantly limit the number of solutions since the state would need to be in its jump set every time the input jumps. Besides, the value of the input does not always contain the information about its forthcoming jump, as illustrated in Section 5, thus preventing the implementation of such an approach. In particular, in the context of observer design, the hybrid input is the output from the observed hybrid plant: the jumps of the observer and of the plant cannot always be synchronized. △

Remark 3. In the case where $\mathrm{dom}\, x = \mathrm{dom}\, u$ is assumed from the start as in[5], u^r is equal to u and Conditions 1) and 2) in Definition 4 are automatically satisfied. Also, in such a case, in Condition 4), the number of jumps of u is equal to the number of jumps of x so that Condition 4b) holds vacuously. The only difference with the definition of solutions in[5] is in the way we define the jumps in Condition 4a). In[5], (x,u) would jump only in D and x would always be reset to values in $G(x,u)$. This case is covered by the definition of G_e^0 (resp. G_e), but we also allow trivial jumps of x when u jumps and (x,u) is in C (resp. $\mathrm{cl}(C)$) (see examples in Section 2.3). △

2.3 | Examples

The purpose of this section is to illustrate the notions introduced in Definitions 3 and 4. For that, let us consider a series interconnection of two hybrid systems $\mathcal{H}_a$ and $\mathcal{H}_b$, where the output of $\mathcal{H}_a$ is the input to $\mathcal{H}_b$, namely

$$\mathcal{H}_a \begin{cases} \dot{x}_a \in F_a(x_a) & x_a \in C_a \\ x_a^+ \in G_a(x_a) & x_a \in D_a \end{cases}, \quad y_a = h_a(x_a), \quad \mathcal{H}_b \begin{cases} \dot{x}_b \in F_b(x_b, y_a) & (x_b, y_a) \in C_b \\ x_b^+ \in G_b(x_b, y_a) & (x_b, y_a) \in D_b \end{cases} \tag{5}$$

Example 2 (Observer design)**.** An important kind of interconnection of this type is the cascade of a plant with its observer. In that case, $\mathcal{H}_a$ is a hybrid plant whose state we want to estimate, and $\mathcal{H}_b$ plays the role of the observer whose input is the output y_a

of the plant $\mathcal{H}_a$. Typically, the goal of the observer $\mathcal{H}_b$ is to provide as output y_b an estimate $\hat{x}_a$ of x_a. This is rendered possible by Definition 4 which defines solutions even when the jumps of y_a (i.e. of the plant) are not synchronized with those of the observer. A sensible definition could thus be the following.

Definition 5. $\mathcal{H}_b$ is an *observer* for $\mathcal{H}_a$ on $\mathcal{X}_{a,0} \subseteq \mathbb{R}^{d_{xa}}$ relative to a set $\mathcal{A}$, if there exists a subset $\mathcal{X}_{b,0}$ of $\mathbb{R}^{d_{xb}}$ such that for any $x_a \in \mathcal{S}_{\mathcal{H}_a}(\mathcal{X}_{a,0})$ with output y_a and for any $(x_b, y_a^r) \in \mathcal{S}_{\mathcal{H}_b}(\mathcal{X}_{b,0}; y_a)$:

(a) y_a^r is a full j-reparametrization of y_a, with associated reparametrization map ρ_a ;

(b) considering the corresponding full j-reparametrization of x_a defined by

$$x_a^r(t,j) = x_a(t, \rho_a(j)) \qquad \forall (t,j) \in \operatorname{dom} \phi_b \ ,$$

we have

$$\lim \left| \Big(y_b(t,j), x_a^r(t,j) \Big) \right|_{\mathcal{A}} = 0 \ . \tag{6}$$

Condition (a) ensures that the solution to the observer $\mathcal{H}_b$ exists as long as the underlying solution x_a to $\mathcal{H}_a$ does. This is important in observer design and comes as an extra constraint besides those of Definition 4. As for Condition (b), it traduces the intuitive idea of "y_b converges to x_a" (in the sense of $\mathcal{A}$), even if those hybrid arcs do not have the same domain. This is done by *reparametrizing* x_a into x_a^r defined on the domain of x_b thanks to Definition 3 and 4. Note that the argument of the limit in (6) is intentionally omitted because it depends on whether we ask for convergence only for complete solutions when $t+j \to +\infty$, or for any solution when (t,j) approaches the boundary of the domain. Regarding $\mathcal{A}$, ideally, we would like $\mathcal{A}$ diagonal, i.e., given by

$$\mathcal{A} = \left\{ (x_a, y_b) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \ : \ x_a = y_b \right\}$$

but it is in general difficult to obtain unless $G_a = \mathrm{Id}$ or the observer becomes perfectly synchronized with the plant after some time. Indeed, if x_a and y_b don't jump exactly at the same time and $G_a \neq \mathrm{Id}$, the mismatch $y_b - x_a$ cannot be made small however small the delay at the jumps is: this is the so-called *peaking* phenomenon. In that case, denoting

$$\underline{G}_a(x_a) = \begin{cases} G_a(x_a) & \text{if } x_a \in D_a \\ \emptyset & \text{otherwise} \end{cases}$$

we can only hope to stabilize the set

$$\mathcal{A} = \left\{ (x_a, y_b) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \ : \ x_a = y_b \ \text{ or } \ x_a \in \underline{G}_a(y_b) \ \text{ or } \ y_b \in \underline{G}_a(x_a) \right\} ,$$

as in [20], or even

$$\mathcal{A} = \left\{ (x_a, y_b) \in (C_a \cup D_a \cup G(D_a))^2 \ : \ \exists k \in \mathbb{N} \ : \ x_a \in \underline{G}_a^k(y_b) \ \text{ or } \ y_b \in \underline{G}_a^k(x_a) \right\}$$

when consecutive jumps are possible, as in [18]. △

Example 3 (Output reference tracking)**.** Another important application is the cascade of a hybrid exosystem $\mathcal{H}_a$ generating a reference y_a that a controlled plant $\mathcal{H}_b$ must follow. In other words, we want y_b to track y_a. This is rendered possible by Definition 4 which defines solutions even when the jumps of y_a (i.e. of the exosystem) are not synchronized with those of the plant. In the same spirit as the observer, we can define:

Definition 6. $\mathcal{H}_b$ *asymptotically tracks* $\mathcal{H}_a$ on $\mathcal{X}_0 \subseteq \mathbb{R}^{d_x}$ relative to a set $\mathcal{A}$, if there exists a subset $\mathcal{X}_{b,0}$ of $\mathbb{R}^{d_{xb}}$ such that for any $x_a \in \mathcal{S}_{\mathcal{H}_a}(\mathcal{X}_{a,0})$ with output y_a and for any $(x_b, y_a^r) \in \mathcal{S}_{\mathcal{H}_b}(\mathcal{X}_{b,0}; y_a)$:

(a) y_a^r is a full j-reparametrization of y_a, with associated reparametrization map ρ_a ;

(b) we have

$$\lim \left| \Big(y_b(t,j), y_a^r(t,j) \Big) \right|_{\mathcal{A}} = 0 \ . \tag{7}$$

The main difference with an observer is that here y_b only has to reproduce y_a, and not the entire state x_a. However, the peaking phenomenon remains when the jumps of y_b are not exactly synchronized with those of y_a. △

Suppose now we want to use the output y_a of $\mathcal{H}_a$ to make $\mathcal{H}_b$ jump according to G_b whenever $\mathcal{H}_a$ jumps. We will consider two settings:

- "Jump triggering": the information of an upcoming jump of $\mathcal{H}_a$ is contained in y_a *before* it happens, namely there exists a subset Y_a of $\mathbb{R}^{d_{y_a}}$ such that $\mathcal{H}_a$ jumps if and only if $y_a \in Y_a$. In that case, we would like to design C_b and D_b so that $\mathcal{H}_b$ jumps according to G_b synchronously with $\mathcal{H}_a$ whenever $y_a \in Y_a$;
- "Jump detection": the information of a jump of $\mathcal{H}_a$ can be detected in y_a *after* it has happened, namely we would like to design C_b and D_b to make $\mathcal{H}_b$ jump right after $\mathcal{H}_a$.

Example 4 (Jump triggering)**.** We start by assuming there exists a subset Y_a of $\mathbb{R}^{d_{y_a}}$ such that $\mathcal{H}_a$ jumps if and only if $y_a = h_a(x_a) \in Y_a$, namely $x_a \in D_a \Leftrightarrow h_a(x_a) \in Y_a$ and no flow is possible from $\mathrm{cl}(C_a) \cap D_a$. An example of this situation presented in[22] is the resettable timer defined by

$$\mathcal{H}_a \begin{cases} \dot{\tau} = -1 & \tau \in C_a := [0, \sup \mathcal{I}] \cap \mathbb{R} \\ \tau^+ \in \mathcal{I} & \tau \in D_a := \{0\} \end{cases} \quad , \qquad y_a = \tau \tag{8}$$

where $\mathcal{I}$ is a closed subset of $\mathbb{R}$, containing the possible lengths of flow interval between successive jumps. Because no flow is possible from $C_a \cap D_a = \{0\}$, we know $\mathcal{H}_a$ is going to jump if and only if $y_a = 0$. Therefore, $Y_a = \{0\}$.

To synchronize $\mathcal{H}_b$ with $\mathcal{H}_a$ a natural choice is

$$C_b = \mathbb{R}^{d_{x_b}} \times (\mathbb{R}^{d_{y_a}} \setminus Y_a) \quad , \quad D_b = \mathbb{R}^{d_{x_b}} \times Y_a \,. \tag{9}$$

Let us build solutions to $\mathcal{H}_b$ according to Definition 4. Take $x_a(0,0) \in \mathrm{cl}(C_a) \cup D_a$ and consider a maximal solution x_a to $\mathcal{H}_a$. Take $x_b(0,0) \in \mathbb{R}^{d_{x_b}}$. If the domain of x_a is reduced to $\{(0,0)\}$, then $T(y_a) = 0$ and $\mathrm{card}\, \mathcal{J}_{T(y_a)} = 0$ so that x_b also stops at $x_b(0,0)$ according to Condition 2) of Definition 4. Now assume $\mathrm{dom}\, x_a \neq \{(0,0)\}$.

First consider the case where $y_a(0,0) \notin D_a$. Then, x_a necessarily flows for $t \in \mathcal{I}_1$, with $\mathcal{I}_1$ a nonempty interval of $\mathbb{R}_{\geq 0}$. By definition of Y_a, $y_a(t,0) \notin Y_a$ for $t \in [0, \sup \mathcal{I}_1)$, so $\mathcal{H}_b$ flows too for $t \in \mathcal{I}_1'$ with $\mathcal{I}_1' \subseteq \mathcal{I}_1$ and $y_a^r := y_a$ on $\mathcal{I}_1' \times \{0\}$. The only way we can have $\mathcal{I}_1' \subsetneq \mathcal{I}_1$ is if x_b explodes in finite time: in that case the solution stops. Otherwise, $\mathcal{I}_1' = \mathcal{I}_1$. Now either the whole domain of x_a has been browsed, in which case x_b stops, or x_a jumps at time $t_1 = \max \mathcal{I}_1$ and necessarily $y_a(t_1, 0) \in Y_a$. If $y_a(t_1, 1) \notin Y_a$, x_a jumps only once, i.e. $\mathcal{J}_{t_1}(y_a) = \{1\}$ and $n_{y_a} = 1$; otherwise, consecutive jumps happen with $\mathcal{J}_{t_1}(y_a) = \{1, 2, \dots\}$ until $y_a \notin Y_a$. Since $(x_b, y_a)(t_1, 0) \in D_b \setminus C_b$ and $t_1 > 0$, x_b is reset to a point in $G_b((x_b, y_a)(t_1, 0))$ according to G_e^0 in the first part of Condition 4a) in Definition 4 with $j = 1 = j_0$ and $t_1 > 0$. We thus take $y^r := y$ on $([0, t_1] \times \{0\}) \cup (\{t_1\} \times \{1\})$. After this first jump,

- either $y_a(t_1, 1) \notin Y_a$, so that $n_{y_a} = 1$, and x_a flows for $t \in \mathcal{I}_2$. Since $(x_b, y_a)(t_1, 1) \in C_b \setminus D_b$, x_b cannot jump further according to Condition 4b) of Definition 4 with $j = 2 \geq j_0 + n_y$: x_b flows and we start again with the same reasoning.
- or $y_a(t_1, 1) \in Y_a$ so that x_a jumps again and $n_{y_a} \geq 2$. If $y_a(t_1, 1) \in \mathrm{int}(Y_a)$, then $(x_b, y_a)(t_1, 1) \in D_b \setminus \mathrm{cl}(C_b)$ and x_b jumps to $x_b(t_1, 2) \in G_b((x_b, y_a)(t_1, 1))$ according to the second part of Condition 4a) in Definition 4 with $j = 2 < j_0 + n_y$. However, if $y_a(t_1, 1) \in \partial Y_a$, then $(x_b, y_a)(t_1, 1) \in D_b \cap \mathrm{cl}(C_b)$, and x_b jumps to $x_b(t_1, 2) \in \{x_b(t_1, 1)\} \cup G_b((x_b, y_a)(t_1, 1))$. We also take $y^r := y$ on $([0, t_1] \times \{0\}) \cup (\{t_1\} \times \{1, 2\})$ and we then start again with the same reasoning.

If now $y_a(0,0) \in Y_a$, x_a starts with a jump. If $y_a(0,0) \in \mathrm{int}(Y_a)$, then $(x_b, y_a)(0,0) \in D_b \setminus \mathrm{cl}(C_b)$ and x_b jumps to $x_b(0,1) \in G_b((x_b, y_a)(0,0))$ according to the second part of Condition 4a) in Definition 4 with $j = 1 = j_0$. However, if $y_a(0,0) \in Y_a \setminus \mathrm{int}(Y_a)$, then $(x_b, y_a)(0,0) \in D_b \cap \mathrm{cl}(C_b)$, and x_b jumps to $x_b(0,1) \in \{x_b(0,0)\} \cup G_b((x_b, y_a)(0,0))$. Then, we carry on with the same reasoning in the bullets above.

So we conclude that $\mathcal{H}_b$ jumps only when $\mathcal{H}_a$ jumps and inherits the domain of its input y_a, so that $y_a^r = y_a$ (unless x_b escapes in finite time while flowing with F_b). Besides, if y_a cannot be in $Y_a \setminus \mathrm{int}(Y_a)$ after a jump of $\mathcal{H}_a$, i.e. if

$$h_a(G_a(D_a)) \cap (Y_a \setminus \mathrm{int}(Y_a)) = \emptyset \,, \tag{10}$$

$\mathcal{H}_b$ jumps according to G_b every time $\mathcal{H}_a$ jumps, except maybe at $t = 0$ where one trivial jump may be allowed if $y_a(0,0) \in Y_a \setminus \mathrm{int}(Y_a)$. To ensure this, the first part of Condition 4a) was crucial to force x_b to be reset to a point in $G_b(x_b, y_a)$ when $(x_b, y_a) \in D_b \setminus C_b$. If we had used G_e instead of G_e^0, trivial jumps would have been allowed since $(x_b, y_a) \in D_b \cap \mathrm{cl}(C_b)$ at the jumps. Instead, if y_a is in $Y_a \setminus \mathrm{int}(Y_a)$ after a jump of $\mathcal{H}_a$, trivial jumps of $\mathcal{H}_b$ are allowed by G_e, thus losing the property of jump triggering. △

Example 5 (Jump detection)**.** Consider now the relaxed case where we allow $\mathcal{H}_b$ to jump according to G_b right after $\mathcal{H}_a$ has jumped. In other words, the jumps of $\mathcal{H}_a$ can be detected in y_a after they have happened, for instance because y_a is in a specific set after the jump or because the jump creates a discontinuity in y_a. This is the case of the timer

$$\mathcal{H}_a \begin{cases} \dot{\tau} = 1 & \tau \in C_a := [0, \sup \mathcal{I}] \cap \mathbb{R} \\ \tau^+ = 0 & \tau \in D_a := \mathcal{I} \end{cases} \quad , \qquad y_a = \tau \tag{11}$$

which creates the same time domains as (8), but this time the information of its jumps is encoded in the output only after they have happened, namely when y_a has been reset to 0.

In order to force $\mathcal{H}_b$ to jump with G_b right after every jump of $\mathcal{H}_a$, we need to choose C_b and D_b such that:

- (x_b, y_a^r) is not in $\mathrm{cl}(C_b)$ after the jumps of $\mathcal{H}_a$, otherwise flow is allowed before $\mathcal{H}_b$ has jumped using G_b;
- after a jump of $\mathcal{H}_b$ using G_b, (x_b, y_a^r) should no longer be in D_b unless $\mathcal{H}_a$ jumps again, otherwise further jumps of $\mathcal{H}_b$ are allowed.

Assume the jumps of $\mathcal{H}_a$ create a discontinuity in y_a which is lower-bounded by some positive scalar δ, and that there exists a continuous map F_{y_a} such that along the flow dynamics of $\mathcal{H}_a$, y_a is solution to

$$\dot{y}_a = F_{y_a}(y_a) \, .$$

Then, the jump detection can be modeled by adding a memory state $\hat{y}_a$ to $\mathcal{H}_b$ which copies y_a and triggers the jumps in $\mathcal{H}_b$ whenever $\hat{y}_a - y_a$ is larger than δ, namely

$$\tilde{\mathcal{H}}_b \begin{cases} \dot{x}_b \in F_b(x_b, y_a) & (x_b, \hat{y}_a, y_a) \in \tilde{C}_b \\ \dot{\hat{y}}_a = F_{y_a}(y_a) & \\ & \\ x_b^+ \in G_b(x_b, y_a) & (x_b, \hat{y}_a, y_a) \in \tilde{D}_b \\ \hat{y}_a^+ = y_a & \end{cases} \tag{12}$$

with

$$\tilde{C}_b = \left\{ (x_b, \hat{y}_a, y_a) \in \mathbb{R}^{d_{x_b}} \times \mathbb{R}^{d_{y_a}} \times \mathbb{R}^{d_{y_a}} \ : \ \hat{y}_a = y_a \right\} \tag{13a}$$

$$\tilde{D}_b = \{ (x_b, \hat{y}_a, y_a) \in \mathbb{R}^{d_{x_b}} \times \mathbb{R}^{d_{y_a}} \times \mathbb{R}^{d_{y_a}} \ : \ |\hat{y}_a - y_a| \geq \delta \} \tag{13b}$$

Indeed, if $y_a(0,0) = \hat{y}_a(0,0)$, then $y_a = \hat{y}_a$ during flow since $\hat{y}_a(t,0) = y_a(0,0) + \int_0^t F_{y_a}(y_a(s,0))ds = y_a(t,0)$ by definition of solutions to differential equations with continuous right-hand side. Therefore, $\tilde{\mathcal{H}}_b$ flows as long as $\mathcal{H}_a$ does (unless it explodes in finite time) and $y_a^r(t,0) = y_a(t,0)$ during that time. If $\mathcal{H}_a$ jumps at $t = t_1$, $|y_a(t_1,1) - y_a(t_1,0)| \geq \delta$, and since $(x_b, \hat{y}_a, y_a)(t_1,0) \in \tilde{C}_b \setminus \tilde{D}_b$, according to Condition 4a), $(x_b, \hat{y}_a)(t_1,1) = (x_b, \hat{y}_a)(t_1,0)$. Besides, we still have $y_a^r(t_1,1) = y_a(t_1,1)$. Therefore, after this jump, $|\hat{y}_a(t_1,1) - y_a^r(t_1,1)| \geq \delta$, i.e. $(x_b, \hat{y}_a, y_a)(t_1,1) \in \tilde{D}_b \setminus \mathrm{cl}(\tilde{C}_b)$ so:

- either $\mathcal{H}_a$ has finished jumping, and from Condition 4b), $\mathcal{H}_b$ jumps with $(x_b, \hat{y}_a)(t_1,2) \in (G_b((x_b, y_a)(t_1,1)), y_a(t_1,1))$ and $y_a^r(t_1,2) = y_a(t_1,1)$. Therefore, we recover $\hat{y}_a(t_1,2) = y_a^r(t_1,2)$, i.e. $(x_b, \hat{y}_a, y_a)(t_1,2) \in \tilde{C}_b \setminus \tilde{D}_b$ and $\tilde{\mathcal{H}}_b$ flows again with $\mathcal{H}_a$.
- or $\mathcal{H}_a$ jumps again with $|y_a(t_1,2) - y_a(t_1,1)| \geq \delta$, and from Condition 4a), we also have $(x_b, \hat{y}_a)(t_1,2) \in (G_b((x_b, y_a)(t_1,1)), y_a(t_1,1))$. So this time, $y_a^r(t_1,2) = y_a(t_1,2)$, and we still have $|\hat{y}_a(t_1,2) - y_a^r(t_1,2)| \geq \delta$, i.e. $(x_b, \hat{y}_a, y_a)(t_1,2) \in \tilde{D}_b \setminus \mathrm{cl}(\tilde{C}_b)$ and another jump follows.

In other words, x_b jumps according to G_b as many times as $\mathcal{H}_a$ does, with one jump delay. If now $|\hat{y}_a(0,0) - y_a(0,0)| \geq \delta$, $\tilde{\mathcal{H}}_b$ necessarily jumps at $t = 0$. So if $\mathcal{H}_a$ does not jump at $t = 0$, we recover the flow condition after the jump and apply the previous case; if $\mathcal{H}_a$ jumps at $t = 0$, then, as above, x_b jumps according to G_b as long as $\mathcal{H}_a$ does, until $\mathcal{H}_a$ stops jumping and $\mathcal{H}_b$ performs one additional jump to recover the flow condition. In other words, when $|\hat{y}_a(0,0) - y_a(0,0)| \geq \delta$, x_b jumps according to G_b one more time than $\mathcal{H}_a$. We finally deduce that with (12), the state x_b of $\tilde{\mathcal{H}}_b$ jumps according to G_b right after every jump of $\mathcal{H}_a$, with maybe one more jump at $t = 0$ if $|\hat{y}_a(0,0) - y_a(0,0)| \geq \delta$, and maybe one fewer if the solution x_a stops while jumping.

This method requires that y_a has independent dynamics and that the discontinuity in y_a at jumps is lower-bounded away from zero (uniformly in time). This is not always satisfied with the data of $\mathcal{H}_a$. However, note that we can always modify the data of $\mathcal{H}_a$ in order to have it verified by at least a part of y_a, which is enough. The idea is to add a discrete state q to $\mathcal{H}_a$ that is toggled at each jump namely

$$\tilde{\mathcal{H}}_a \begin{cases} \dot{x}_a \in F_a(x_a) & (x_a, q) \in C_a \times \{0,1\} \\ \dot{q} = 0 & \\ & \\ x_a^+ \in G_a(x_a) & (x_a, q) \in D_a \times \{0,1\} \\ q^+ = 1 - q & \end{cases} , \qquad \tilde{y}_a = (h_a(x_a), q) =: (y_a, y_q) \tag{14}$$

It is the same system, but a jump can now be detected by a toggle of the discrete state q. The flow dynamics of y_q are independent and the jumps create in y_q a discontinuity of norm equal to 1. Therefore, repeating the same arguments, the jump detection can simply be modeled by

$$\tilde{\mathcal{H}}_b \begin{cases} \dot{x}_b \in F_b(x_b, y_a) & (x_b, \hat{q}, y_a, y_q) \in \tilde{C}_b \\ \dot{\hat{q}} = 0 & \\ & \\ x_b^+ \in G_b(x_b, y_a) & (x, \hat{q}, y_a, y_q) \in \tilde{D}_b \\ \hat{q}^+ = y_q & \end{cases} \tag{15}$$

with

$$\tilde{C}_b = \left\{ (x, \hat{q}, y_a, y_q) \in \mathbb{R}^{d_{x_b}} \times \{0,1\} \times \mathbb{R}^{d_{y_a}} \times \{0,1\} \ : \ \hat{q} = y_q \right\} \tag{16a}$$

$$\tilde{D}_b = \{ (x, \hat{q}, y_a, y_q) \in \mathbb{R}^{d_{x_b}} \times \{0,1\} \times \mathbb{R}^{d_{y_a}} \times \{0,1\} \ : \ \hat{q} = 1 - y_q \} . \tag{16b}$$

Note that we could also easily model a more realistic delayed jump detection by adding a timer in $\mathcal{H}_b$ as in[23,17]. △

3 | ALGORITHM TO GENERATE SOLUTIONS TO HYBRID SYSTEMS WITH HYBRID INPUTS

3.1 | Algorithm

The construction of a solution to a hybrid system with hybrid input can be made explicit through an algorithm. Before we introduce this algorithm, it is useful to define/build solutions when the input is a continuous time function $u_{CT} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_u}$.

Definition 7. Consider an interval I_u of $\mathbb{R}_{\geq 0}$ such that $0 \in I_u$, and a function $u_{CT} : I_u \to \mathbb{R}^{d_u}$. The hybrid arc (x, u^r) is solution to $\mathcal{H}$ with continuous-time input u_{CT} and output y, if (x, u^r) is solution to $\mathcal{H}$ as in Definition 4 with hybrid input u and output y, where u is the hybrid arc defined on $I_u \times \{0\}$ by

$$u(t, 0) = u_{CT}(t) \quad \forall t \in I_u .$$

In other words, u^r is trivially given on dom x by

$$u^r(t, j) = u_{CT}(t) \quad \forall (t, j) \in \operatorname{dom} x ,$$

and x is simply characterized by

- $\operatorname{dom}_t x \subseteq I_u$ and if $\operatorname{dom}_t x = I_u$, card $\mathcal{J}_{T(u)}(x) = 0$.
- for all $j \in \mathbb{N}$ such that $\mathcal{I}_j(x)$ has non-empty interior,

$$(x(t, j), u_{CT}(t)) \in C \quad \forall t \in \operatorname{int} \mathcal{I}_j(x)$$
$$\dot{x}(t, j) \in F(x(t, j), u_{CT}(t)) \quad \text{for a.a. } t \in \mathcal{I}_j(x)$$

- for all $(t, j) \in \operatorname{dom} x$ such that $(t, j-1) \in \operatorname{dom} x$,

$$(x(t, j-1), u_{CT}(t)) \in D$$
$$x(t, j) \in G(x(t, j-1), u_{CT}(t))$$

- $\operatorname{dom} x = \operatorname{dom} y$ and for all (t, j) in dom x,

$$y(t, j) = h(x(t, j), u_{CT}(t)) .$$

The solution x is said to be maximal if (x, u^r) is maximal. By abuse of notation, the set of maximal solutions to $\mathcal{H}$ initialized in $\mathcal{X}_0$ with continuous-time input u_{CT} is also denoted $\mathcal{S}_{\mathcal{H}}(\mathcal{X}_0; u_{CT})$. △

Based on this definition, and on the observation that the solutions are easily built when the input is a continuous-time function, we can introduce Algorithm 1 (see next page), which constructs maximal solutions (x, u^r) to $\mathcal{H}$ with a hybrid input u and output y according to Definition 4 as follows:

1. The algorithm starts by defining I_u, the time interval to elapse before reaching the next jump of u. The interval is a singleton if u has an immediate jump.
2. Over the time interval I_u, u evolves continuously and, if possible (line 9), the algorithm builds (line 12) a maximal hybrid solution $\underline{x}$ to system (4) starting from x_0 according to Definition 7. This gives Conditions 3) and 4b). $\underline{x}$ is appended to the solution x.
3. If (line 20) $\underline{x}$ ends before reaching the end of the interval I_u, or ends outside of $\mathrm{cl}(C) \cup D$ (resp. $C \cup D$ after flow, namely if $T_m := T(\underline{x}) > 0$ for the first case of Condition 4a)), the algorithm stops.
4. Otherwise, j_u is incremented, I_u is updated to the next interval of flow of u, and x jumps according to G_e^0 if $T_m > 0$ (i.e. after flow), and G_e otherwise, to satisfy Condition 4a).

By construction, we deduce the following result.

Proposition 1. Consider a hybrid arc u. The hybrid arc $\phi = (x, u^r)$ is a maximal solution to $\mathcal{H}$ with input u and output y if and only if x, u^r, and y are possible outputs of Algorithm 1 with input u.

Note that there are two sources of non uniqueness of solutions in the algorithm: first, in the construction of solutions with continuous input with Definition 7, and through the set-valued jump maps G_e^0 and G_e.

3.2 | Numerical implementation of Algorithm 1

To illustrate the algorithm and observe the impact of numerical errors on the definition of solutions, we simulate the series interconnection (5) of two autonomous hybrid systems modeling periodically reset timers, denoted $\mathcal{H}_a$ and $\mathcal{H}_b$ with period $\bar{t}_a$ and $\bar{t}_b$ respectively. More precisely, we take $y_a = x_a$ and define the data (F_a, C_a, G_a, D_a) of $\mathcal{H}_a$ and (F_b, C_b, G_b, D_b) of $\mathcal{H}_b$ as

$$F_a(x_a) = F_b(x_b, y_a) = 1\,, \quad G_a(x_a) = G_b(x_b, y_a) = 0\,, \quad C_a = [0, \bar{t}_a]\,, \quad D_a = \{\bar{t}_a\}\,, \quad C_b = [0, \bar{t}_b] \times \mathbb{R}\,, \quad D_b = \{\bar{t}_b\} \times \mathbb{R} \quad (17)$$

From its initial condition in $[0, \bar{t}_a]$, $\mathcal{H}_a$ flows until it reaches $\bar{t}_a$, then jumps with x_a reset to 0, starts again flowing etc. As for $\mathcal{H}_b$, if it were not for the input y_a, it would behave in the same way, with period $\bar{t}_b$. But although the dynamics of $\mathcal{H}_b$ are independent from the value of y_a, considering y_a as input means we need to apply Definition 4 to build solutions. In other words, $\mathcal{H}_b$ is reset to zero when x_b reaches $\bar{t}_b$, but it also jumps (maybe trivially) when $y_a = x_a$ jumps. To simulate such a behavior, we implement[1] Algorithm 1 using the function `HyEQsolver` from the Matlab Hybrid Toolbox[24].

3.2.1 | Numerical implementation

Given an initial condition $x_{a,0}$ of $\mathcal{H}_a$, `HyEQsolver` gives a solution x_a to $\mathcal{H}_a$ on an horizon of time T_a chosen here equal to 10. Then, to build a solution to $\mathcal{H}_b$, we browse the domain of $y_a = x_a$ as described by Algorithm 1.

More precisely, on each interval of flow I_u of x_a, `HyEQsolver` is called to produce a solution to $\mathcal{H}_b$ on the horizon of time determined by I_u. This solution is appended to x_b and a reparametrization x_a^r of x_a is jointly built on I_u by adding trivial jumps to x_a whenever x_b jumps: x_a^r and x_b are defined on the same domain. If the end of the time interval I_u has not been reached by x_b, the algorithm stops. Otherwise, at the end of I_u, a jump is created in (x_b, x_a^r) with x_b reset either trivially or to $0 = G_b(x_b, y_a)$, according to G_e or G_e^0 defined in Definition 4 (using (C_b, D_b, G_b) in place of of (C, D, G) therein).

Actually, since numerically x_a is never exactly equal to $\bar{t}_a$ and x_b is never exactly equal to $\bar{t}_b$, we enlarge D_a and D_b as

$$D_a = [\bar{t}_a, +\infty) \quad , \quad D_b = [\bar{t}_b, +\infty) \times \mathbb{R}$$

which give the same solutions as long as they are initialized in $[0, \bar{t}_a]$ and $[0, \bar{t}_b]$. In the simulations below, we use $\bar{t}_a = 1$ and $\bar{t}_b = 0.5$.

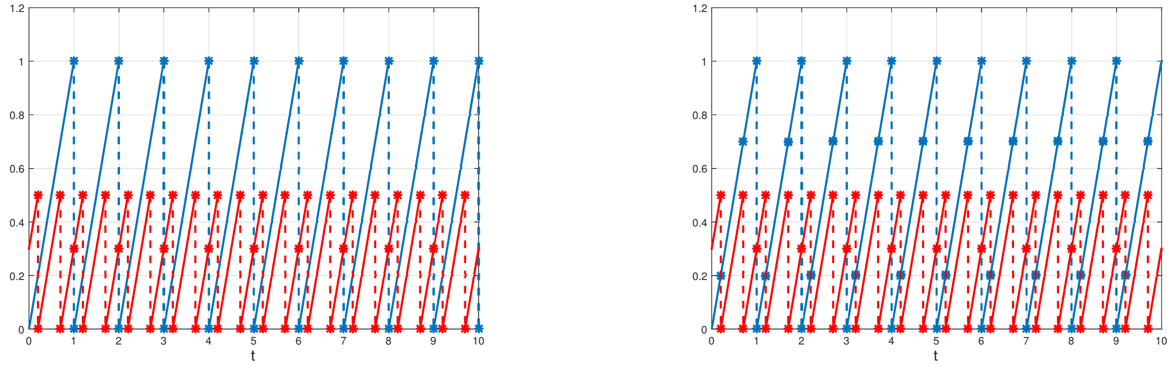
[1] Code available at `https://github.com/HybridSystemsLab/AlgorithmHSwithInputs`

Algorithm 1 Maximal solution to $\mathcal{H}$ initialized in $\mathcal{X}_0$ with hybrid input u

1: $\mathcal{D}, x, y, u^r, \rho_u \leftarrow \emptyset$
2: $j \leftarrow 0$
3: $t_j \leftarrow 0$
4: $j_u \leftarrow 0$
5: $x_0 \in \mathcal{X}_0$
6: $I_u \leftarrow \{t \in \mathbb{R}_{\geq 0}\ \ (t, j_u) \in \operatorname{dom} u\}$
7: **while** $I_u \neq \emptyset$ **do**
8: $\quad u_{CT}(t - t_j) \leftarrow u(t, j_u) \quad \forall t \in I_u$
9: $\quad$ **if** $\mathcal{S}_{\mathcal{H}}(x_0; u_{CT}) = \emptyset$ **then**
10: $\quad\quad$ go to line 35
11: $\quad$ **else**
12: $\quad\quad$ Pick $\underline{x} \in \mathcal{S}_{\mathcal{H}}(x_0; u_{CT})$ with output $\underline{y}$
13: $\quad\quad T_m \leftarrow T(\underline{x})$
14: $\quad\quad j_m \leftarrow J(\underline{x})$
15: $\quad\quad \mathcal{D} \leftarrow \mathcal{D} \cup \left(\{(t_j, j)\} + \operatorname{dom} \underline{x}\right)$
16: $\quad\quad x(t_j + \underline{t}, j + \underline{j}) \leftarrow \underline{x}(\underline{t}, \underline{j}) \quad \forall (\underline{t}, \underline{j}) \in \operatorname{dom} \underline{x}$
17: $\quad\quad y(t_j + \underline{t}, j + \underline{j}) \leftarrow \underline{y}(\underline{t}, \underline{j}) \quad \forall (\underline{t}, \underline{j}) \in \operatorname{dom} \underline{x}$
18: $\quad\quad u^r(t_j + \underline{t}, j + \underline{j}) \leftarrow u_{CT}(\underline{t}) \quad \forall (\underline{t}, \underline{j}) \in \operatorname{dom} \underline{x}$
19: $\quad\quad \rho_u(j + \underline{j}) \leftarrow j_u \quad \forall \underline{j} \in \{0, 1, \ldots, j_m\} \cap \mathbb{N}$
20: $\quad\quad$ **if** $T_m \notin \operatorname{dom}_t \underline{x}$ **or** $j_m = +\infty$ **or** $T_m < T(u_{CT})$ **or** $(\underline{x}(T_m, j_m), u_{CT}(T_m)) \notin \operatorname{cl}(C) \cup D$ **or** $(T_m > 0$ **and** $(\underline{x}(T_m, j_m), u_{CT}(T_m)) \notin C \cup D)$ **then**
21: $\quad\quad\quad$ go to line 35
22: $\quad\quad$ **else**
23: $\quad\quad\quad t_j \leftarrow t_j + T_m$
24: $\quad\quad\quad j \leftarrow j + j_m + 1$
25: $\quad\quad\quad j_u \leftarrow j_u + 1$
26: $\quad\quad\quad I_u \leftarrow \{t \in \mathbb{R}_{\geq 0}\ :\ (t, j_u) \in \operatorname{dom} u\}$
27: $\quad\quad\quad$ **if** $T_m > 0$ **then**
28: $\quad\quad\quad\quad x_0 \in G_e^0(\underline{x}(T_m, j_m), u_{CT}(T_m))$
29: $\quad\quad\quad$ **else**
30: $\quad\quad\quad\quad x_0 \in G_e(\underline{x}(T_m, j_m), u_{CT}(T_m))$
31: $\quad\quad\quad$ **end if**
32: $\quad\quad$ **end if**
33: $\quad$ **end if**
34: **end while**
35: $J \leftarrow \sup_j \mathcal{D}$ $\quad\quad$ ▷ Convention : $\sup \emptyset = -\infty$
36: **if** $J \in [0, +\infty)$ **then**
37: $\quad \rho_u(j) \leftarrow \rho_u(J) \quad \forall j \in \mathbb{N}\ :\ j \geq J$
38: **end if**
39: **return** x, y, u^r, ρ_u

3.2.2 | Numerical solutions for non synchronized timers

We start by considering initial conditions $x_{a,0} = 0$ and $x_{b,0} = 0.3$ for which the two timers are never reset at the same time. Solutions are plotted on Figure 1. We see that $\mathcal{H}_b$ is always in $C_b \setminus D_b$ when $\mathcal{H}_a$ jumps so that every jump of $\mathcal{H}_a$ triggers a trivial jump of $\mathcal{H}_b$. This can be seen on Figure (1a). Then, on Figure (1b), we show the reparametrization x_a^r of x_a on the same domain as x_b. We see that trivial jumps have been added in x_a^r at every jump time of x_b where x_a does not jump.



(a) x_b (red) and x_a (blue)

(b) x_b (red) and x_a^r (blue)

FIGURE 1 Trajectories of x_a solution to $\mathcal{H}_a$ and (x_b, x_a^r) solution to $\mathcal{H}_b$ with input x_a with $(x_{a,0}, x_{b,0}) = (0, 0.3)$.

3.2.3 | Numerical solutions for synchronized timers

Now consider the case where $x_{b,0} = 0$. Let us first see what should happen in theory. Due to the definition of the dynamics, and because $\bar{t}_a = 2\bar{t}_b$, at every jump of $\mathcal{H}_a$, we have $x_b = \bar{t}_b$, namely $\mathcal{H}_b$ is in $C_b \cap D_b$. Therefore, according to the definition of G_e^0, we have the choice between a trivial reset of x_b or a reset to 0. In the former case, $\mathcal{H}_b$ then performs another jump to be reset to 0. In other words, each jump of $\mathcal{H}_a$ triggers one or two jumps in $\mathcal{H}_b$.

If we had chosen instead

$$C_b = [0, \bar{t}_b) ,$$

$\mathcal{H}_b$ would be in $D_b \setminus C_b$ at the jumps of $\mathcal{H}_a$, and by definition of G_e^0, x_b would be forced to be reset to 0, so that only one jump would happen. In other words, $\mathcal{H}_a$ and $\mathcal{H}_b$ would be perfectly synchronized.

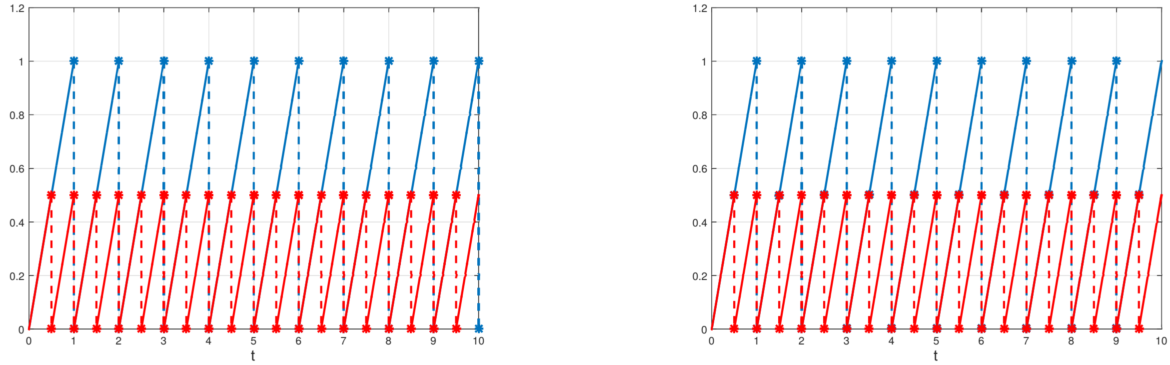
In simulations now, the solutions are plotted on Figure 2. Although they appear perfectly synchronized, it turns out that the jumps of $\mathcal{H}_a$ actually trigger one or two jumps in $\mathcal{H}_b$. In fact, due to numerical errors, x_b usually gets past $\bar{t}_b$ slightly before or slightly after x_a gets past $\bar{t}_a$, resulting in a jump of $\mathcal{H}_b$ slightly before or after the one of $\mathcal{H}_a$. And regarding the openness of C_b, the exact same results are obtained taking C_b open or closed because the jumps are rarely triggered at $x_b = \bar{t}_b$ exactly, but rather for $x_b > \bar{t}_b$ so that x_b is not in C_b whatever its definition. Since this cannot be seen on Figure 2, we plot on Figure 3 the jumps of x_a and x_b: x_a jumps 10 times from 1 to 0, whereas x_b jumps synchronously with x_a for the first 5 jumps and then has sometimes trivial jumps around 0.5 when it is slightly delayed with respect to $\mathcal{H}_a$.

We conclude that numerically speaking,

- the outer-semicontinuity of the map G_e, namely the choice between a jump along Id or G_b in Definition 4, accounts for the solutions where $\mathcal{H}_b$ is slightly delayed with respect to its input resulting in consecutive jumps instead of simultaneous ones,
- when C_b is open, the distinction between G_e^0 and G_e in Condition 4)a) is not visible in simulations since the numerical errors make it impossible to exploit the solution in ∂C_b, namely we obtain the solutions corresponding to the closure of C_b.

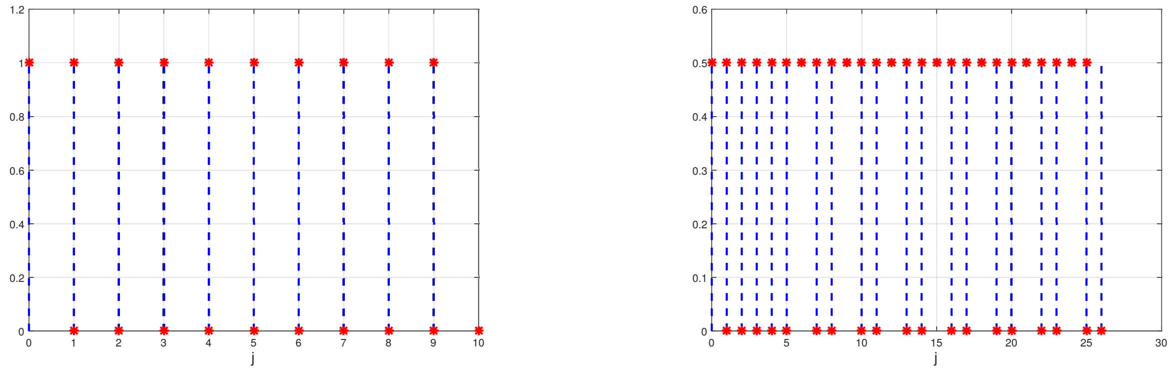
This is coherent with the results obtained in[21] for standard hybrid systems, which say that robustness comes with outer-semicontinuity of the maps and closure of the sets.

Actually, more generally, we could also obtain simulations where $\mathcal{H}_b$ jumps slightly ahead of $\mathcal{H}_a$ due to numerical errors. Those solutions do not appear with Definition 4 since Condition 4) requires the jumps of the input (here x_a) to be processed first and consecutively. In fact, those extra solutions would be covered by robustness of the definition if we chose Conditions 2') and 4') of Remark 1 instead of Condition 2) and 4). Indeed, in that case, the jump of x_a would be allowed to be processed after the reset of x_b. We will see in Example 8 in Section 4.1 how those extra solutions also appear when writing the cascade of $\mathcal{H}_a$ and $\mathcal{H}_b$ as a single extended hybrid system.



(a) x_b (red) and x_a (blue) **(b)** x_b (red) and x_a^r (blue)

FIGURE 2 Trajectories of x_a solution to $\mathcal{H}_a$ and (x_b, x_a^r) solution to $\mathcal{H}_b$ with input x_a with $(x_{a,0}, x_{b,0}) = (0, 0)$.



(a) x_a **(b)** x_b

FIGURE 3 Jumps of x_a solution to $\mathcal{H}_a$ and (x_b, x_a^r) solution to $\mathcal{H}_b$ with input x_a and with initial condition $(x_{a,0}, x_{b,0}) = (0, 0)$:the graphs represent the value of the hybrid arcs before and after each jump. We see that the first jumps of x_a trigger only one jump in x_b, while the following ones trigger two jumps in x_b, namely x_b is first trivially reset and then jumps to 0.

4 | APPLICATION TO INTERCONNECTIONS OF HYBRID SYSTEMS AND LINK TO CLOSED-LOOP SYSTEMS

The study of interconnected hybrid systems is crucial in multiple contexts, from reference tracking to observer design along with output-feedback. To facilitate this analysis and, in particular, in order to use Lyapunov tools, it is handy to generate solutions based on a single global hybrid system that captures the behavior of all the interconnected systems. Therefore, we investigate the link between solutions in the sense of Definition 4 and such a closed-loop system.

4.1 | Series Interconnections

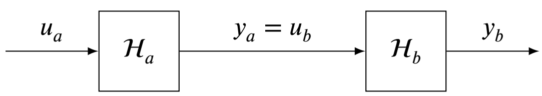


FIGURE 4 Series interconnection of two hybrid systems

In control theory, the input of a system is often the output of another system. For instance, in observer design the input of the observer is the output/measurement of the system we want to observe. The examples considered in the previous section also fall into that category. For two cascaded hybrid systems $\mathcal{H}_a = (C_a, F_a, D_a, G_a, h_a)$ and $\mathcal{H}_b = (C_b, F_b, D_b, G_b, h_b)$ with inputs u_a and u_b and outputs y_a and y_b such that $y_a = u_b$ as in Figure 4, it is natural to consider the cascaded closed-loop system $\mathcal{H}_{cl}$ (also denoted $\mathcal{H}_a \to \mathcal{H}_b$) with input u_a and output y_b defined by

$$\mathcal{H}_{cl} \begin{cases} \begin{pmatrix} \dot{x}_a \\ \dot{x}_b \end{pmatrix} \in F_{cl}(x_a, x_b, u_a) & (x_a, x_b, u_a) \in C_{cl} \\ \begin{pmatrix} x_a^+ \\ x_b^+ \end{pmatrix} \in G_{cl}(x_a, x_b, u_a) & (x_a, x_b, u_a) \in D_{cl} \end{cases} \quad , \qquad y_b = h_b(x_b, h_a(x_a, u_a)) \tag{18}$$

with

$$C_{cl} = \left\{ (x_a, x_b, u_a) \in \mathbb{R}^{d_{xa}} \times \mathbb{R}^{d_{xb}} \times \mathbb{R}^{d_{ua}} \ : (x_a, u_a) \in C_a \ , \ (x_b, h_a(x_a, u_a)) \in C_b \right\} \tag{19}$$

$$\begin{aligned} D_{cl} = & \left\{ (x_a, x_b, u_a) \in \mathbb{R}^{d_{xa}} \times \mathbb{R}^{d_{xb}} \times \mathbb{R}^{d_{ua}} \ : (x_a, u_a) \in D_a \ , \ (x_b, h_a(x_a, u_a)) \in \mathrm{cl}(C_b) \cup D_b \right\} \\ & \cup \left\{ (x_a, x_b, u_a) \in \mathbb{R}^{d_{xa}} \times \mathbb{R}^{d_{xb}} \times \mathbb{R}^{d_{ua}} \ : (x_a, u_a) \in \mathrm{cl}(C_a) \cup D_a \ , \ (x_b, h_a(x_a, u_a)) \in D_b \right\} \end{aligned} \tag{20}$$

and

$$F_{cl}(x_a, x_b, u_a) = \begin{pmatrix} F_a(x_a, u_a) \\ F_b(x_b, h_a(x_a, u_a)) \end{pmatrix} \tag{21}$$

$$G_{cl}(x_a, x_b, u_a) = \begin{pmatrix} \underline{G}_a(x_a, u_a) \\ \underline{\mathrm{Id}}_b(x_b) \end{pmatrix} \cup \begin{pmatrix} \underline{\mathrm{Id}}_a(x_a) \\ \underline{G}_b(x_b, h_a(x_a, u_a)) \end{pmatrix} \cup \begin{pmatrix} \underline{G}_a(x_a, u_a) \\ \underline{G}_b(x_b, h_a(x_a, u_a)) \end{pmatrix} \tag{22}$$

where we have denoted for i in $\{a, b\}$

$$\underline{G}_i(x_i, u_i) = \begin{cases} G_i(x_i, u_i) & \text{if } (x_i, u_i) \in D_i \\ \emptyset & \text{otherwise} \end{cases} \quad , \qquad \underline{\mathrm{Id}}_i(x_i) = \begin{cases} x_i & \text{if } x_i \in \mathrm{cl}(C_i) \\ \emptyset & \text{otherwise} \end{cases} \ . \tag{23}$$

Similar closed-loop or *extended* systems have been introduced in the literature whenever it was needed to compare hybrid arcs with different domains, for instance in the context of reference tracking [18] or incremental stability [19]. The main difference with those references is that we allow here both x_a and x_b to jump simultaneously with G_a and G_b, whereas in [18,19] this kind of jump is decomposed into two successive jumps, one where x_a jumps with G_a and x_b is trivially reset, and vice versa for the second. In other words, the third jump map in (22) is absent. The main reasons for allowing simultaneous jumps here are:

- We want to recover the framework of discrete-time systems with $C_i = \emptyset$;
- Due to the presence of u_a, one simultaneous jump of x_a and x_b cannot always be decomposed in two successive jumps of x_b and then x_a, because u_a may also jump in-between.

Thanks to the "simultaneous jump" part of $\mathcal{H}_{cl}$, it is sufficient to allow trivial jumps of x_i only on $\mathrm{cl}(C_i)$, as can be seen on the definition of $\underline{\mathrm{Id}}_i$. In other words, unlike in [19], x_i is forced to jump with G_i on $D_i \setminus \mathrm{cl}(C_i)$. Note that it is however not possible to replace $\mathrm{cl}(C_i)$ by C_i in the definition of $\underline{\mathrm{Id}}_i$. Indeed, x_a could flow from ∂C_a at a time where x_b needs to jump, in which case a trivial jump of x_a should be allowed.

We would like to link the solutions of hybrid systems with hybrid inputs defined in the previous sections, to the solutions of the closed-loop (18). We are going to show in Lemma 1 that (roughly speaking) if x_a is a solution to $\mathcal{H}_a$ with input u_a and output y_a, and x_b is a solution to $\mathcal{H}_b$ with input $u_b = y_a$, then, "$((x_a, x_b), u_a)$" (modulo some j-reparametrizations) is a solution to $\mathcal{H}_{cl}$. However, we will see in Lemma 1 that the set of solutions to $\mathcal{H}_{cl}$ is larger, in the sense that the converse statement relating the solutions of $\mathcal{H}_{cl}$ to solutions of $\mathcal{H}_a$ and $\mathcal{H}_b$ holds under the following additional conditions.

Definition 8 (Converse Conditions)**.** Take a solution $\phi_{cl} = ((x_{a,cl}, x_{b,cl}), u_{a,cl})$ to system $\mathcal{H}_{cl}$ with input u_a. Denote ρ_{u_a} the input j-reparametrization map from u_a to $u_{a,cl}$. For $i = a, b$, at a time t in $\mathcal{T}(\phi_{cl})$ and a jump $j \in \mathcal{J}_t(\phi_{cl})$, we will say that $x_{i,cl}$ verifies its jump condition if

- $(x_{i,cl}(t, j-1), u_{i,cl}(t, j-1)) \in D_i$
- $x_{i,cl}(t, j) \in G_i(x_{i,cl}(t, j-1), u_{i,cl}(t, j-1))$

where we denote $u_{b,cl} = y_{a,cl} = h_a(x_{a,cl}, u_{a,cl})$. Then, ϕ_{cl} is said to verify the *Converse Conditions* (CCs) if for any t in $\mathcal{T}(\phi_{cl})$, denoting $j_0 = \min \mathcal{J}_t(\phi_{cl})$ and $n_{u_a} = \text{card}(\mathcal{J}_t(u_a))$,

CC.1) there exists an integer $n_{x_a} \geq n_{u_a}$ such that for all $j \in \mathcal{J}_t(\phi_{cl})$, denoting $y_{a,cl} = h_a(x_{a,cl}, u_{a,cl})$,

- if $j < j_0 + n_{u_a}$
  - $\rho_{u_a}(j) = \rho_{u_a}(j-1) + 1$
- if $j_0 + n_{u_a} \leq j < j_0 + n_{x_a}$
  - $\rho_{u_a}(j) = \rho_{u_a}(j-1)$
  - $x_{a,cl}$ verifies its jump condition
- if $j \geq j_0 + n_{x_a}$
  - $\rho_{u_a}(j) = \rho_{u_a}(j-1)$
  - $x_{a,cl}$ does not verify its jump condition
  - $x_{b,cl}$ verifies its jump condition.

CC.2) if $t > 0$ and $n_{u_a} \geq 1$,

- $(x_{a,cl}(t, j_0 - 1), u_{a,cl}(t, j_0 - 1)) \in C_a \cup D_a$
- $x_{a,cl}(t, j_0) \in G_a(x_{a,cl}(t, j_0 - 1), u_{a,cl}(t, j_0 - 1))$ if $(x_{a,cl}(t, j_0 - 1), u_{a,cl}(t, j_0 - 1)) \in D_a \setminus C_a$

CC.3) if $t > 0$ and $n_{x_a} \geq 1$,

- $(x_{b,cl}(t, j_0 - 1), y_{a,cl}(t, j_0 - 1)) \in C_b \cup D_b$
- $x_{b,cl}(t, j_0) \in G_b(x_{b,cl}(t, j_0 - 1), y_{a,cl}(t, j_0 - 1))$ if $(x_{b,cl}(t, j_0 - 1), y_{a,cl}(t, j_0 - 1)) \in D_b \setminus C_b$

CC.4) if $t \in \text{int dom}_t(\phi_{cl})$ and $n_{x_a} = 0$, $(x_{a,cl}(t, j), u_{a,cl}(t, j)) \in C_a$ for all $j \in \mathcal{J}_t(\phi_{cl})$.

CC.5) if $t = T(\phi_{cl})$, then $n_{x_a} = \text{card}\, \mathcal{J}_t(\phi_{cl})$.

△

Remark 4. The fact that u_a performs all its jumps consecutively before $j < j_0 + n_{u_a}$ is already contained in the fact that ϕ_{cl} is a solution to $\mathcal{H}_{cl}$ according to Condition 4) in Definition 4. The additional constraints contained in the CCs of Definition 8 are:

- After removing the jumps of u_a, i.e., for $j \geq j_0 + n_{u_a}$, x_a does all its jumps consecutively and right away. This is because it is going to play the role of input for $\mathcal{H}_b$ and must therefore satisfy the constraint of consecutiveness of input jumps imposed by Condition 4) in Definition 4. This disappears if Condition 4) is replaced by Condition 4') defined in Remark 1.
- For the first jump of u_a, (x_a, u_a) must be in $C_a \cup D_a$ and x_a must jump according to G_a if (x_a, u_a) is in $D_a \setminus C_a$; similarly, at the first jump of x_a, (x_b, u_b) must be in $C_b \cup D_b$ and x_b must jump according to G_b, if (x_b, u_b) is in $D_b \setminus C_b$. Those constraints disappear if C_i are closed (because then the corresponding states are necessarily in C_i after flow) or if we remove the constraint involving G_e^0 at $j = j_0$ in Condition 4a) of Definition 4.
- At times t in the interior of the domain, (x_a, u_a) must be in C_a if neither x_a nor u_a jumps at all at time t (this enables to ensure that when we remove the jumps due to x_b in $x_{a,cl}$, we obtain a hybrid arc x_a that is in C_a in the interior of the flow interval.). This constraint disappears if C_a is closed.
- Since x_a is going to play the role of input for $\mathcal{H}_b$, x_b must stop whenever x_a does according to Condition 2) in Definition 4. This disappears if we take Condition 2') defined in Remark 1 instead.

In other words, the CCs would be automatically verified when C_a and C_b are closed if Conditions 2) and 4) of Definition 4 were replaced by Conditions 2') and 4') of Remark 1. Also, in the particular case where ϕ_{cl} jumps if and only if u_a jumps, then $n_{u_a} = \text{card}\, \mathcal{J}_t(\phi_{cl})$ at all jumps times, and CC.1,4,5) automatically hold, so that only CC.2,3) remain. This will be exploited for feedback interconnections in Lemma 2. △

Lemma 1 (Cascaded hybrid systems)**.** Consider two hybrid systems $\mathcal{H}_a = (C_a, F_a, D_a, G_a, h_a)$ and $\mathcal{H}_b = (C_b, F_b, D_b, G_b, h_b)$ with inputs u_a and u_b and outputs y_a and y_b respectively, and the corresponding closed-loop system $\mathcal{H}_{cl}$ defined in (18). Take any solution $\phi_a = (x_a, u_a^r)$ to $\mathcal{H}_a$ with input u_a and output y_a, and any solution $\phi_b = (x_b, u_b^r)$ to $\mathcal{H}_b$ with input $u_b = y_a$ and output y_b. Denote ρ_b the j-reparametrization map from u_b to u_b^r. Then, considering the corresponding j-reparametrizations of x_a and u_a^r defined by

$$x_{a,cl}(t,j) = x_a(t, \rho_b(j)) \qquad \forall (t,j) \in \operatorname{dom} x_b\ ,$$
$$u_{a,cl}(t,j) = u_a^r(t, \rho_b(j)) \qquad \forall (t,j) \in \operatorname{dom} x_b\ ,$$

$\phi_{cl} = ((x_{a,cl}, x_b), u_{a,cl})$ is solution to $\mathcal{H}_{cl}$ with input u_a and output y_b, and satisfies CC.1,2,3,4). It also satisfies CC.5) if $T(\phi_b) = T(\phi_a)$.

Conversely, if $\phi_{cl} = ((x_{a,cl}, x_{b,cl}), u_{a,cl})$ is a solution to the hybrid system $\mathcal{H}_{cl}$ with input u_a satisfying the CCs, there exists a solution (x_a, u_a^r) to $\mathcal{H}_a$ with input u_a and output y_a such that

- (x_b, u_b^r) with $x_b = x_{b,cl}$ and $u_b^r = y_{a,cl} = h_a(x_{a,cl}, u_{a,cl})$, is solution to $\mathcal{H}_b$ with input $u_b = y_a$
- $x_{a,cl}$ and $u_{a,cl}$ are full j-reparametrizations of x_a and u_a^r respectively.

Proof. See Appendix. □

An important consequence of Lemma 1 is the following.

Corollary 1 (Observer design)**.** Consider two cascaded hybrid systems $\mathcal{H}_a = (C_a, F_a, D_a, G_a, h_a)$ and $\mathcal{H}_b = (C_b, F_b, D_b, G_b, h_b)$ as in (5) and the corresponding closed-loop system $\mathcal{H}_{cl}$ defined in (18). $\mathcal{H}_b$ is an observer for $\mathcal{H}_a$ in the sense of Definition 5 if and only if for any maximal solution $\phi_{cl} = (x_{a,cl}, x_b)$ to $\mathcal{H}_{cl}$ (without u_a) satisfying the CCs (see Definition 8),

(a) either ϕ_{cl} is complete, or $x_{a,cl}$ explodes in finite time, or no flow nor jump is possible for $x_{a,cl}$ from its final value.

(b) $\lim |(h_b(x_b(t,j)), x_{a,cl}(t,j))|_{\mathcal{A}} = 0$.

Proof. Direct consequence from Lemma 1 once having noticed that the first condition means that $\operatorname{dom}\phi_{cl}$ is limited by $x_{a,cl}$, not by x_b, thus giving item (a) of Definition 5; and that the second condition corresponds to (6) in item (b) of Definition 5. □

This latter result is important because the analysis of $\mathcal{H}_{cl}$ is handier and allows the use of Lyapunov tools.

Example 6 (Jump triggering)**.** Let's go back to Example 4 and compare the solutions of the series interconnection $\mathcal{H}_a \to \mathcal{H}_b$, with $\mathcal{H}_a$ defined in (8) and $\mathcal{H}_b$ defined in (5)-(9), to those produced by the corresponding closed-loop (18). The flow condition of $\mathcal{H}_{cl}$ is given by

$$\begin{pmatrix} \dot{x}_a \\ \dot{x}_b \end{pmatrix} \in \begin{pmatrix} F_a(x_a) \\ F_b(x_b, h_a(x_a)) \end{pmatrix} \qquad \text{if } x_a \in C_a \text{ and } h_a(x_a) \notin Y_a$$

and the possibilities at jumps are

$$\begin{pmatrix} x_a^+ \\ x_b^+ \end{pmatrix} \in \begin{cases} \begin{pmatrix} G_a(x_a) \\ G_b(x_b, h_a(x_a)) \end{pmatrix} & \text{if } x_a \in D_a \quad (\Longleftrightarrow h_a(x_a) \in Y_a) \\ \begin{pmatrix} G_a(x_a) \\ x_b \end{pmatrix} & \text{if } h_a(x_a) \in Y_a \setminus \operatorname{int}(Y_a) \\ \begin{pmatrix} x_a \\ G_b(x_b, h_a(x_a)) \end{pmatrix} & \text{if } x_a \in \partial C_a \text{ and } h_a(x_a) \in Y_a \end{cases}$$

Indeed, $x_a \in D_a \Leftrightarrow h_a(x_a) \in Y_a$ and $(x_b, y_a) \in \operatorname{cl}(C_b) \Leftrightarrow h_a(x_a) \in \operatorname{cl}(\mathbb{R}^{d_{y_a}} \setminus Y_a)$, which gives the second jump condition. Besides, the fact that no flow is possible from $\operatorname{cl}(C_a) \cap D_a$ implies that $D_a \cap \operatorname{int}(C_a) = \emptyset$, which gives the third condition. It is easy to see that as planned by the first part of Lemma 1, the solutions found in Example 4 are indeed solutions to the closed loop system. However, notice that the closed-loop system also admits extra solutions: for instance if $x_a \in \partial C_a$ and $h_a(x_a) \in Y_a$, x_b can jump according to G_b any number of times without changing x_a, or x_a could jump with G_a and x_b trivially reset if $h_a(x_a) \in Y_a \setminus \operatorname{int}(Y_a)$ even at the first jumps of x_a. Let us show that those solutions are excluded by the CCs, thus confirming the converse part of Lemma 1.

- if at some point $x_a \in \partial C_a$ and $h_a(x_a) \in Y_a$, then $x_a \in D_a \cap \mathrm{cl}(C_a)$, then no flow is possible by assumption. Therefore, the solution jumps. Assume it jumps via the third jump map, namely x_a is trivially reset and x_b jumps via G_b. As long as this jump map is used, x_a is still in $D_a \cap C_a$ and no flow is possible. So either x_a is reset infinitely many times trivially or the solution ends up using one of the other two jump maps where x_a is reset to $G_a(x_a)$. The first possibility is excluded by CC.5) since at the final time $n_{x_a} < +\infty$. The second possibility is excluded by CC.1) since x_a does not perform all its jumps with G_a consecutively. Therefore, solutions using the third jump map are excluded, meaning that x_a necessarily jumps according to G_a at every jump. Therefore, for any solution (x_a, x_b) of $\mathcal{H}_{cl}$ satisfying the CCs, x_a is solution to $\mathcal{H}_a$ and x_b inherits the domain of x_a as we saw above.

- Now let us study the jumps of x_b. Take a jump time of (x_a, x_b) and consider the first jump at this time. If $t = 0$ and $y_a(0,0) \in Y_a \setminus \mathrm{int}(Y_a)$, x_b can be trivially reset. Otherwise, if $t > 0$, the solution has just flowed so that it is in $\mathrm{cl}(C_{cl}) \cap D_{cl}$, meaning that $h_a(x_a) \in \partial Y_a$, and therefore $(x_b, y_a) \in D_b \setminus C_b$. According to CC.3), x_b necessarily jumps according to G_b. At the following jumps, x_b could be trivially reset if $h_a(x_a) \in Y_a \setminus \mathrm{int}(Y_a)$: we recover condition (10) to ensure that x_b always jumps according to G_b.

This illustrates the fact that $\mathcal{H}_{cl}$ introduces new solutions, but keeping only the solutions of $\mathcal{H}_{cl}$ that satisfy the CCs, enables to recover the solutions found in Example 4. In fact, in the particular context of jumps triggering where we want the jumps of $\mathcal{H}_b$ to be synchronized with those of $\mathcal{H}_a$, we should rather consider the simple closed-loop system:

$$\begin{pmatrix} \dot{x}_a \\ \dot{x}_b \end{pmatrix} \in \begin{pmatrix} F_a(x_a) \\ F_b(x_b, h_a(x_a)) \end{pmatrix} \qquad x_a \in C_a$$
$$\begin{pmatrix} x_a^+ \\ x_b^+ \end{pmatrix} \in \begin{pmatrix} G_a(x_a) \\ G_b(x_b, h_a(x_a)) \end{pmatrix} \qquad x_a \in D_a$$

△

Example 7 (Jump detection)**.** Let us now go back to Example 5 and compare the solutions of the series interconnection $\tilde{\mathcal{H}}_a \to \tilde{\mathcal{H}}_b$, with $\tilde{\mathcal{H}}_a$ defined in (14) and $\tilde{\mathcal{H}}_b$ defined in (15)-(16), to those produced by the corresponding closed-loop (18). The flow condition of $\mathcal{H}_{cl}$ is given by

$$\begin{pmatrix} \dot{x}_a \\ \dot{q} \\ \dot{x}_b \\ \dot{\hat{q}} \end{pmatrix} \in \begin{pmatrix} F_a(x_a) \\ 0 \\ F_b(x_b, h_a(x_a)) \\ 0 \end{pmatrix} \qquad \text{if } x_a \in C_a \text{ and } q = \hat{q}$$

and the possibilities at jumps are

$$\begin{pmatrix} x_a^+ \\ q^+ \\ x_b^+ \\ \hat{q}^+ \end{pmatrix} \in \begin{cases} \begin{pmatrix} G_a(x_a) \\ 1-q \\ G_b(x_b, h_a(x_a)) \\ q \end{pmatrix} & \text{if } x_a \in D_a \text{ and } \hat{q} = 1-q \\ \begin{pmatrix} G_a(x_a) \\ 1-q \\ x_b \\ \hat{q} \end{pmatrix} & \text{if } x_a \in D_a \text{ and } \hat{q} = q \\ \begin{pmatrix} x_a \\ q \\ G_b(x_b, h_a(x_a)) \\ q \end{pmatrix} & \text{if } x_a \in \mathrm{cl}(C_a) \cup D_a \text{ and } \hat{q} = 1-q \end{cases}$$

It is easy to check that the solutions found in Example 5 are solutions to the closed-loop. Regarding the CCs,

- CC.1) requires that at each jump time of the solution, x_a performs all its jumps according to G_a right away and consecutively. Therefore, only the first two jump maps can be used, except maybe at the last jump (observing that the third jump map can be used only once)

- CC.2) is void because $\tilde{\mathcal{H}}_a$ does not have an input

- CC.3) is automatically satisfied because C_b is closed (see Remark 4)
- at any jump time $t > 0$, the first jump necessarily follows the second jump map since $\hat{q} = q$ after flow. Therefore, x_a jumps according to G_a and CC.4) is void.
- CC.5) only requires that if at some point the component x_a can no longer flow with F_a nor jump with G_a, the solution stops.

It is easy to see that any solution to $\mathcal{H}_{cl}$ satisfying those CCs corresponds to a solution found in Example 5. Actually, the extra solutions to $\mathcal{H}_{cl}$ are those which use alternatively the third and second jump maps instead of the first: this corresponds in fact to writing the first jump map as the composition of the third and second, namely first x_b is updated via G_b and then x_a via G_a instead of simultaneously. Therefore, those extra solutions have extra jumps but still model a jump detection. In fact, we could also model the jump detection simply with the jump map

$$\begin{pmatrix} x_a^+ \\ q^+ \\ x_b^+ \\ \hat{q}^+ \end{pmatrix} \in \begin{cases} \begin{pmatrix} G_a(x_a) \\ 1-q \\ x_b \\ \hat{q} \end{pmatrix} & \text{if } x_a \in D_a \text{ and } \hat{q} = q \\ \begin{pmatrix} x_a \\ q \\ G_b(x_b, h_a(x_a)) \\ q \end{pmatrix} & \text{if } x_a \in \mathrm{cl}(C_a) \cup D_a \text{ and } \hat{q} = 1-q \end{cases}$$

△

Example 8 (Cascade of timers)**.** We finally revisit the numerical example of Section 3.2 made of the series interconnection of two timers. In this case, the equivalent closed-loop system (18) has flow dynamics given by

$$\begin{pmatrix} \dot{x}_a \\ \dot{x}_b \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \text{if } x_a \in C_a \text{ and } x_b \in C_b$$

and the possibilities at jumps are

$$\begin{pmatrix} x_a^+ \\ x_b^+ \end{pmatrix} \in \begin{cases} \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{if } x_a = \bar{t}_a \text{ and } x_b = \bar{t}_b \\ \begin{pmatrix} 0 \\ x_b \end{pmatrix} & \text{if } x_a = \bar{t}_a \text{ and } x_b \in [0, \bar{t}_b] \\ \begin{pmatrix} x_a \\ 0 \end{pmatrix} & \text{if } x_a \in [0, \bar{t}_a] \text{ and } x_b = \bar{t}_b \end{cases}$$

We observe that when x_a and x_b reach $\bar{t}_a$ and $\bar{t}_b$ respectively at the same time, they can either both be reset to 0 in a single jump or one after the other in two jumps. The solution where x_a is first reset to 0 while x_b jumps trivially, was predicted by Definition 4 in the case where C_b is closed, and was observed numerically on Figure 3b. On the other hand, the solution where x_b is first reset to 0 (before x_a) did not appear. This is because Condition 4) of Definition 4 requires to process all the jumps of the input (here x_a) right away. In fact, CC.1) is not satisfied for those solutions. It turns out however that those solutions can appear on simulations, when, due to numerical errors, x_b jumps slightly ahead of x_a. In this sense, the closed-loop extended system (18) models a larger class of solutions (as predicted by Lemma 1) and can therefore offer more robustness to a control/observer design. △

4.2 | Feedback Interconnections

In the previous section, we have studied the series interconnection of $\mathcal{H}_a = (C_a, F_a, D_a, G_a, h_a)$ and $\mathcal{H}_b = (C_b, F_b, D_b, G_b, h_b)$ with $u_b = y_a$. We now consider the case of feedback where also $u_a = y_b$ as in Figure 5, for instance if $\mathcal{H}_b$ is an observer-controller for $\mathcal{H}_a$. We have seen that by connecting $\mathcal{H}_b$ with $\mathcal{H}_a$, $\mathcal{H}_b$ jumps whenever $\mathcal{H}_a$ does. Now that $\mathcal{H}_a$ is also connected with $\mathcal{H}_b$, we have that $\mathcal{H}_a$ jumps whenever $\mathcal{H}_b$ does, so that the solutions are defined on a common time domain containing the jumps

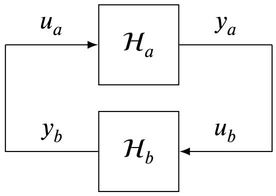


FIGURE 5 Feedback interconnection of hybrid systems

of both $\mathcal{H}_a$ and $\mathcal{H}_b$. In fact, in that case, the construction of solutions is not sequential but simultaneous so it is natural to build them at the same time through the closed-loop $\mathcal{H}_a \rightleftarrows \mathcal{H}_b$ defined by

$$\mathcal{H}_{cl} \begin{cases} \begin{pmatrix} \dot{x}_a \\ \dot{x}_b \end{pmatrix} \in F_{cl}(x_a, x_b) & (x_a, x_b) \in C_{cl} \\ \begin{pmatrix} x_a^+ \\ x_b^+ \end{pmatrix} \in G_{cl}(x_a, x_b) & (x_a, x_b) \in D_{cl} \end{cases} \tag{24}$$

with

$$C_{cl} = \left\{ (x_a, x_b) \in \mathbb{R}^{d_{xa}} \times \mathbb{R}^{d_{xb}} \ : (x_a, h_b(x_b)) \in C_a \ , \ (x_b, h_a(x_a)) \in C_b \right\} \tag{25}$$

$$\begin{aligned} D_{cl} = & \left\{ (x_a, x_b) \in \mathbb{R}^{d_{xa}} \times \mathbb{R}^{d_{xb}} \ : (x_a, h_b(x_b)) \in D_a \ , \ (x_b, h_a(x_a)) \in \mathrm{cl}(C_b) \cup D_b \right\} \\ & \cup \left\{ (x_a, x_b) \in \mathbb{R}^{d_{xa}} \times \mathbb{R}^{d_{xb}} \ : (x_a, h_b(x_b)) \in \mathrm{cl}(C_a) \cup D_a \ , \ (x_b, h_a(x_a)) \in D_b \right\} \end{aligned} \tag{26}$$

and

$$F_{cl}(x_a, x_b) = \begin{pmatrix} F_a(x_a, h_b(x_b)) \\ F_b(x_b, h_a(x_a)) \end{pmatrix} \tag{27}$$

$$G_{cl}(x_a, x_b) = \begin{pmatrix} \underline{G}_a(x_a, h_b(x_b)) \\ \underline{\mathrm{Id}}_b(x_b) \end{pmatrix} \cup \begin{pmatrix} \underline{\mathrm{Id}}_a(x_a) \\ \underline{G}_b(x_b, h_a(x_a)) \end{pmatrix} \cup \begin{pmatrix} \underline{G}_a(x_a, h_b(x_b)) \\ \underline{G}_b(x_b, h_a(x_a)) \end{pmatrix} \tag{28}$$

with $\underline{\mathrm{Id}}_i$ and $\underline{G}_i$ defined in (23). Here again, allowing for a simultaneous jump of x_a and x_b in G_{cl} is crucial because unlike in [18,19], G_a and G_b depend on both x_a and x_b, so that one simultaneous jump cannot be decomposed into sequential jumps of x_a first and then x_b, or vice-versa.

Lemma 2. Consider two hybrid systems $\mathcal{H}_a = (C_a, F_a, D_a, G_a, h_a)$ and $\mathcal{H}_b = (C_b, F_b, D_b, G_b, h_b)$ with $h_a : \mathbb{R}^{d_{xa}} \to \mathbb{R}^{d_{ya}}$ and $h_b : \mathbb{R}^{d_{xb}} \to \mathbb{R}^{d_{yb}}$. Take a solution $\phi_{cl} = (x_a, x_b)$ to (24). If for all $t \in \mathcal{T}(\phi_{cl}) \cap \mathbb{R}_{>0}$, denoting $j_0 = \min \mathcal{J}_t(\phi_{cl})$,

- $(x_a(t, j_0 - 1), y_b(t, j_0 - 1)) \in C_a \cup D_a$
- $x_a(t, j_0) \in G_a(x_a(t, j_0 - 1), y_b(t, j_0 - 1))$ if $(x_a(t, j_0 - 1), y_b(t, j_0 - 1)) \in D_a \setminus C_a$
- $(x_b(t, j_0 - 1), y_a(t, j_0 - 1)) \in C_b \cup D_b$
- $x_b(t, j_0) \in G_b(x_b(t, j_0 - 1), y_a(t, j_0 - 1))$ if $(x_b(t, j_0 - 1), y_a(t, j_0 - 1)) \in D_b \setminus C_b$

then, $\phi_a = (x_a, h_b(x_b))$ is solution to $\mathcal{H}_a$ with input $h_b(x_b)$ and $\phi_b = (x_b, h_a(x_a))$ is solution to $\mathcal{H}_b$ with input $h_a(x_a)$.

This extra condition is added to ensure that G_e^0 is used instead of G_e at the first jumps of the input in Condition 4 of Definition 4. It corresponds to CC.2,3) in Definition 8 and is always satisfied if C_a and C_b are closed. As planned in Remark 4, the other CCs have disappeared because they are automatically satisfied thanks to the fact that ϕ_a, ϕ_b and ϕ_{cl} share the same domain.

Corollary 2. Consider two hybrid systems $\mathcal{H}_a = (C_a, F_a, D_a, G_a, h_a)$ and $\mathcal{H}_b = (C_b, F_b, D_b, G_b, h_b)$ with $h_a : \mathbb{R}^{d_{xa}} \to \mathbb{R}^{d_{ya}}$ and $h_b : \mathbb{R}^{d_{xb}} \to \mathbb{R}^{d_{yb}}$. Assume C_a and C_b are closed. Then, for any solution $\phi_{cl} = (x_a, x_b)$ to (24), $\phi_a = (x_a, h_b(x_b))$ is solution to $\mathcal{H}_a$ with input $h_b(x_b)$ and $\phi_b = (x_b, h_a(x_a))$ is solution to $\mathcal{H}_b$ with input $h_a(x_a)$.

5 | CONCLUSION

We have shown how solutions to hybrid systems with inputs can be defined when the input is an hybrid arc whose domain does not match that of the solution. A novel definition was proposed and discussed that relies on a reparametrization of the input jumps, along with an explicit algorithm for the construction of solutions. Those notions were applied to the important cases of series or feedback interconnections of two hybrid systems, for which the link to a closed-loop system was investigated.

This work is instrumental in defining and studying observers for hybrid systems. Ongoing work involve defining notions of detectability that should be intrinsically necessary for the existence of an observer. Similarly to the context of incremental stability [19], detectability requires to compare hybrid trajectories that do not share the same domain. Therefore, in the same spirit as this paper, such trajectories first need to be reparametrized onto a common domain. Applications to tracking and output-feedback can of course also be studied following the concepts of this paper.

Future work also involve the extension of the code for the numerical implementation of Algorithm 1 to general hybrid systems with hybrid inputs. The case where the input does not impact the dynamics of the system, as in the example of Section 3.2, was a first step[2], and a complete toolbox for the simulation of interconnected hybrid systems should now be developed.

ACKNOWLEDGMENT

The authors would like to thank Marcello Guarro for his help in the numerical implementation of the algorithm.

References

1. Jiang ZP, Teel AR, Praly L. Small-gain theorem for ISS systems and applications. *Math. Control Signals Syst.* 1994; 7: 95–120.

2. Teel AR. A nonlinear small gain theorem for the analysis of control systems with saturation. *IEEE Transactions on Automatic Control* 1996; 41(9): 1256-1270.

3. Jiang ZP, Wang Y. Input-to-state stability for discrete-time nonlinear systems. *Automatica* 2001; 37: 857-869.

4. Mancilla-Aguilar JL, Garcia RA. On converse Lyapunov theorems for ISS and iISS switched nonlinear systems. *Systems & Control Letters* 2001; 42(1): 47–53.

5. Cai C, Teel AR. Characterizations of input-to-state stability for hybrid systems. *Syst. & Cont. Letters* 2009; 58: 47–53.

6. Sontag ED, Wang Y. Output-to-state stability and detectability of nonlinear systems. *Systems & Control Letters* 1997; 29: 279–290.

7. Cai C, Teel AR. Output-to-state stability for hybrid systems. *Systems & Control Letters* 2011; 60: 62–68.

8. Dashkovskiy SN, Efimov DV, Sontag ED. Input to state stability and allied system properties. *Automation and Remote Control* 2011; 72(8): 1579–1614.

9. Sanfelice RG. Input-Output-to-State Stability Tools for Hybrid Systems and Their Interconnections. *IEEE Transactions on Automatic Control* 2014; 59(5): 1360-1366.

10. Lygeros J, Johansson K, Simić S, Zhang J, Sastry SS. Dynamical Properties of Hybrid Automata. *IEEE Transactions on Automatic Control* 2003; 48(1): 2-17.

11. Goebel R, Sanfelice RG, Teel AR. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. New Jersey: Princeton University Press . 2012

[2] Code available at `https://github.com/HybridSystemsLab/AlgorithmHSwithInputs`

12. Sanfelice RG. Interconnections of Hybrid Systems: Some Challenges and Recent Results. *Journal of Nonlinear Systems and Applications* 2011; 2(1-2): 111–121. doi: http://jnsaonline.watsci.org/abstract_pdf/2011v2/v2n1-pdf/13.pdf

13. Nesic D, Teel A. Input-output stability properties of networked control systems. *IEEE Transactions on Automatic Control* 2004; 49: 1650-1667.

14. Tanwani A, Shim H, Liberzon D. Observability for switched linear systems : characterization and observer design. *IEEE Transactions on Automatic Control* 2013; 58(4): 891-904.

15. Medina EA, Lawrence DA. State Estimation for Linear Impulsive Systems. *Annual American Control Conference* 2009: 1183-1188.

16. Sanfelice RG, Biemond JJB, Wouw v. dN, Heemels WPMH. An Embedding Approach for the Design of State-Feedback Tracking Controllers for References with Jumps. *International Journal of Robust and Nonlinear Control* 2013; 24(11): 1585–1608. doi: http://dx.doi.org/10.1002/rnc.2944

17. Bernard P, Sanfelice RG. Observers for hybrid dynamical systems with linear maps and known jump times. *IEEE Conference on Decision and Control* 2018: 3140-3145.

18. Biemond B, Wouw v. dN, Heemels M, Nijmeijer H. Tracking Control for Hybrid Systems With State-Triggered Jumps. *IEEE Transactions on Automatic Control* 2013; 58(4): 876-890.

19. Biemond JB, Postoyan R, Heemels W, Wouw dNV. Incremental stability of hybrid dynamical systems. *IEEE Transactions on Automatic Control* 2018; 111(108662).

20. Forni F, Teel AR, Zaccarian L. Follow the bouncing ball : global results on tracking and state estimation with impacts. *IEEE Transactions on Automatic Control* 2013; 58(6): 1470-1485.

21. Goebel R, Sanfelice RG, Teel A. *Hybrid Dynamical Systems : Modeling, Stability and Robustness*. Princeton University Press . 2012.

22. Bernard P, Sanfelice RG. An Algorithm to Generate Solutions to Hybrid Dynamical Systems with Inputs. *2019 American Control Conference* 2019: 2996-3001.

23. Altin B, Sanfelice RG. On robustness of pre-asymptotic stability to delayed jumps in hybrid systems. *Annual American Control Conference* 2018.

24. Sanfelice RG, Copp DA, Nanez P. A Toolbox for Simulation of Hybrid Systems in Matlab/Simulink: Hybrid Equations (HyEQ) Toolbox. In: ; 2013: 101–106

□

APPENDIX

We prove here Lemma 1.

Proof. To show that $((x_{a,cl}, x_b), u_{a,cl})$ is solution to $\mathcal{H}_{cl}$ with input u_a, we are going to check every condition of Definition 4.

1. $\operatorname{dom} \phi_{cl} = \operatorname{dom}(x_{a,cl}, x_b) = \operatorname{dom} x_b = \operatorname{dom} u_{a,cl}$
2. $u_{a,cl}$ is a j-reparametrization of u_a^r (with reparametrization map ρ_b) which is a j-reparametrization of u_a according to the Condition 2) of Definition 4 with reparametrization map ρ_a. So $u_{a,cl}$ is a j-reparametrization of u_a with reparametrization map $\rho_u = \rho_a \circ \rho_b$. Besides, if $u_{a,cl}$ is a full-reparametrization of u_a, u_a^r necessarily is too. Denoting $T := T(u_a) = T(\phi_a) = T(\phi_{cl})$, according to Condition 2) applied to ϕ_a, $\operatorname{card} \mathcal{J}_T(u_a) = \operatorname{card} \mathcal{J}_T(\phi_a)$. Now, applying Condition 2) to ϕ_b, we get $\operatorname{card} \mathcal{J}_T(\phi_b) = \operatorname{card} \mathcal{J}_T(y_a) = \mathcal{J}_T(\phi_a)$. Since $\operatorname{card} \mathcal{J}_T(\phi_b) = \operatorname{card} \mathcal{J}_T(\phi_{cl})$ by definition, we deduce $\operatorname{card} \mathcal{J}_T(u_a) = \operatorname{card} \mathcal{J}_T(\phi_{cl})$.

3. From the Condition 1) of Definition 4, u_b^r is a j-reparametrization of $u_b = y_a$ and
$$u_b^r(t,j) = u_b(t,\rho_b(j)) = h_a(x_{a,cl}(t,j),u_{a,cl}(t,j)) \qquad \forall (t,j) \in \mathrm{dom}\,\phi_{cl}$$
so that
$$(x_b(t,j), h_a(x_{a,cl}(t,j),u_{a,cl}(t,j))) = (x_b(t,j),u_b^r(t,j)) = \phi_b(t,j) \qquad \forall (t,j) \in \mathrm{dom}\,\phi_{cl}\ .$$
We have also
$$(x_{a,cl}(t,j),u_{a,cl}(t,j)) = (x_a(t,\rho_b(j)),u_a^r(t,\rho_b(j))) = \phi_a(t,\rho_b(j)) \qquad \forall (t,j) \in \mathrm{dom}\,\phi_{cl}\ .$$
So for all j, int $\mathcal{I}_j(\phi_{cl}) \subseteq$ int $\mathcal{I}_{\rho_b(j)}(\phi_a)$, int $\mathcal{I}_j(\phi_{cl}) =$ int $\mathcal{I}_j(\phi_b)$, and, applying Condition 3) of Definition 4 to ϕ_a and ϕ_b, we get that Condition 3) is verified for ϕ_{cl}.

4. Let t in $\mathcal{T}(\phi_{cl}) = \mathcal{T}(\phi_b)$ and $j_0 = \min \mathcal{J}_t(\phi_{cl}) = \min \mathcal{J}_t(\phi_b)$. According to Condition 4) of Definition 4 applied to ϕ_b, there exists n_{u_b} such that for all $j \in \mathcal{J}_t(\phi_b)$, $\rho_b(j) = \rho_b(j-1)+1$ if $j < j_0 + n_{u_b}$, and $\rho_b(j) = \rho_b(j-1)$ if $j \geq j_0 + n_{u_b}$. By definition of the j-reparametrization,
$$\mathcal{J}_t(\phi_a) = \{\rho_b(j) \ : \ j \in \mathcal{J}_t(\phi_b)\}$$
According to Condition 4) of Definition 4 applied to ϕ_a, there exists n_{u_a} such that for all $j \in \mathcal{J}_t(\phi_a)$, $\rho_a(j) = \rho_a(j-1)+1$ if $j < \rho_b(j_0) + n_{u_a}$, and $\rho_a(j) = \rho_a(j-1)$ if $j \geq \rho_b(j_0) + n_{u_a}$. Therefore, the reparametrization map $\rho_u = \rho_a \circ \rho_b$ from u_a to $u_{a,cl}$ verifies : for all $j \in \mathcal{J}_t(\phi_{cl})$, $\rho_u(j) = \rho_u(j-1)+1$ if $j < j_0 + n_{u_a}$, and $\rho_u(j) = \rho_u(j-1)$ if $j \geq j_0 + n_{u_a}$. The rest of Condition 4) follows in a tedious yet straightforward way from Condition 4) of Definition 4 applied to ϕ_a and ϕ_b.

5. The Condition 5) is clear from the definition of y_b.

The prioritized input jumps conditions follows from the following remarks:

- CC.1) the fact that u_a performs all its jumps consecutively before $j < j_0 + n_{u_a}$ is contained in the fact that ϕ_{cl} is a solution to $\mathcal{H}_{cl}$ according to item 4) in Definition 4. After removing the jumps of u_a, i.e., for $j \geq j_0 + n_{u_a}$, x_a does all its jumps consecutively (up to $j_0 + n_{u_b} = j_0 + n_{x_a}$) according to item 4) in Definition 4, because it is an input for $\mathcal{H}_b$.
- CC.2) at $j = j_0$, if $t > 0$, and u_a jumps ($n_{u_a} \geq 1$), (x_a,u_a) is necessarily in $C_a \cup D_a$, and x_a jumps according to G_a if (x_a,u_a) is in $D_a \setminus C_a$ from the definition of G_e^0 in item 4) of Definition 4 applied to ϕ_a.
- CC.3) similarly, if $t > 0$, and x_a jumps ($n_{x_a} \geq 1$), the input to $\mathcal{H}_b$ jumps, thus giving a similar condition on x_b at the first jump.
- CC.4) if t is in the interior of $\mathrm{dom}_t\, \phi_{cl}$ and if x_a does not jump ($n_{x_a} = 0$), t is necessarily in the interior of a flow interval of x_a, and therefore, by item 3) of the definition 4, $(x_a,u_a) \in C_a$.
- CC.5) if $T(\phi_{cl}) = T(\phi_a)$ and $T := T(\phi_{cl}) \in \mathcal{T}(\phi_{cl})$, either the full domain of ϕ_a is browsed in ϕ_b (and thus in ϕ_{cl}) and from Condition 2) applied to ϕ_b, card $\mathcal{J}_T(\phi_b) =$ card $\mathcal{J}_T(y_a)$ and with CC.1), $n_{x_a} =$ card $\mathcal{J}_T(\phi_a) =$ card $\mathcal{J}_T(y_a)$, so that card $\mathcal{J}_T(\phi_{cl}) =$ card $\mathcal{J}_T(\phi_b) = n_{x_a}$; or the full domain of ϕ_a is not browsed in ϕ_b, meaning that ϕ_b stops jumping before ϕ_a at time T, and therefore also card $\mathcal{J}_T(\phi_{cl}) = n_{x_a}$. In other words, the third item of CC.1) is empty.

Conversely, take a solution $\phi_{cl} = ((x_{a,cl},x_{b,cl}),u_{a,cl})$ to system $\mathcal{H}_{cl}$ with input u_a verifying CC.1,2,3,4). Denote ρ_u the j-reparametrization map between u_a and $u_{a,cl}$. We build hybrid arcs x_a and u_a^r in the following way :

- start with $\mathcal{D}_a = \mathcal{I}_0(\phi_{cl}) \times \{0\}$, $x_a \equiv x_{a,cl|\mathcal{D}_a}$, $u_a^r \equiv u_{a,cl|\mathcal{D}_a}$, $j_a = 0$, $j_u = 0$, $\rho_a(0) = 0$, $\rho_b(0) = 0$.
- for j from 1 to $J(\phi_{cl})$ do (we denote $t_j = t_j(\phi^r)$ to simplify the notations) :
  - if $\rho_u(j) = \rho_u(j-1)+1$, then $j_u \leftarrow j_u + 1$.
  - if either $\rho_u(j) = \rho_u(j-1)+1$, or $x_{a,cl}$ verifies its jump condition, then $j_a \leftarrow j_a + 1$.
  - $\mathcal{D}_a \leftarrow \mathcal{D}_a \cup (\mathcal{I}_j(\phi_{cl}) \times \{j_a\})$
  - $x_a(t,j_a) \leftarrow x_{a,cl}(t,j)$ for all t in $\mathcal{I}_j(\phi_{cl})$
  - $u_a^r(t,j_a) \leftarrow u_{a,cl}(t,j)$ for all t in $\mathcal{I}_j(\phi_{cl})$
  - $\rho_a(j_a) \leftarrow j_u$

- $\rho_b(j) \leftarrow j_a$

Then, we take $y_a = h_a(x_a, u_a^r)$. Let us prove that $\phi_a = (x_a, u_a^r)$ is solution to $\mathcal{H}_a$ with input u_a and output y_a.

1. $\operatorname{dom} x_a = \operatorname{dom} u_a^r = \mathcal{D}_a$ which is a hybrid time domain by construction (since ϕ_{cl} is an hybrid arc)
2. u_a^r is a j-reparametrization of u_a with reparametrization map ρ_a. Indeed, if at a given iteration j_a does not change, j_u does not change either, so that taking $\rho_a(j_a) \leftarrow j_u$ does not change anything ; a change of j_u corresponding to an actual jump of u_a according to the definition of ρ_u, ρ_a stays constant as long as u_a does not jump and is increased by one when u_a jumps. Besides, since u_a^r is built from $u_{a,cl}$, if u_a^r is a full j-reparametrization of u_a, $u_{a,cl}$ is too. By Condition 2) applied to $\phi_{a,cl}$, we deduce that $\operatorname{card} \mathcal{J}_T(\phi_{cl}) = \operatorname{card} \mathcal{J}_T(u_a)$, and since the jumps in ϕ_a are extracted from those of ϕ_{cl}, $\operatorname{card} \mathcal{J}_T(\phi_a) \leq \operatorname{card} \mathcal{J}_T(\phi_{cl})$, so that necessarily to have a full reparametrization, $\operatorname{card} \mathcal{J}_T(\phi_a) = \operatorname{card} \mathcal{J}_T(u_a)$.
3. for all j_a in $\operatorname{dom}_j \phi_a$, there exist positive integers $j_1, j_2, \dots, j_k$ such that
$$\mathcal{I}_{j_a}(\phi_a) = \mathcal{I}_{j_1}(\phi_{cl}) \cup \ldots \cup \mathcal{I}_{j_k}(\phi_{cl})$$
and $j_2, \dots, j_{k-1}$ correspond to jumps of ϕ_{cl} where $(x_{a,cl}, u_{a,cl})$ is constant, and in C_a if the corresponding jumps times are in the interior of the interval according to CC.4). Therefore, x_a and u_a^r are absolutely continuous on $\mathcal{I}_{j_a}(\phi_a)$, for almost all t in $\mathcal{I}_{j_a}(\phi_a)$, $\dot{x}_a \in F_a(x_a(t, j_a), u_a^r(t, j_a))$, and for all t in $\operatorname{int} \mathcal{I}_{j_a}(\phi_a)$, $(x_a(t, j_a), u_a^r(t, j_a)) \in C_a$.
4. Take $t \in \mathcal{T}(\phi_a)$, denote $j_0 = \min \mathcal{J}_t(\phi_a)$ and $n_u = \operatorname{card}\ \mathcal{J}_t(u_a)$, we have for all $j \in \mathcal{J}_t(\phi_a)$:
   (a) for $j < j_0 + n_u$, $\rho_u(j) = \rho_u(j-1)+1$, and from the definition of G_{cl}, $(x_a(t, j-1), u_a^r(t, j-1)) \in \operatorname{cl}(C_a) \cup D_a$ and $x_a(t, j) \in G_e(x_a(t, j-1), u_a^r(t, j-1))$. More precisely, from CC.2), if $t > 0$, $(x_a(t, j_0-1), u_a^r(t, j_0-1)) \in C_a \cup D_a$ and $x_a(t, j_0-1)$ jumps according to G_a if $(x_a(t, j_0-1), u_a^r(t, j_0-1)) \in D_a \setminus C_a$. Necessarily, $x_a(t, j_0) \in G_e^0(x_a(t, j_0-1), u_a^r(t, j_0-1))$.
   (b) for $j \geq j_0 + n_u$, $\rho_u(j) = \rho_u(j-1)$ and necessarily $(x_{a,cl}(t_j, j-1), u_{a,cl}(t_j, j-1)) \in D_a$ and $x_{a,cl}(t, j) \in G_a(x_{a,cl}(t_j, j-1), u_{a,cl}(t_j, j-1))$ from the construction of ϕ_a.
5. $y_a = h_a(x_a, u_a^r)$ by definition.

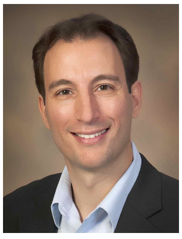
Now let us prove that (x_b, u_b^r) with $x_b = x_{b,cl}$ and $u_b^r = y_{a,cl} = h_a(x_{a,cl}, u_{a,cl})$ is solution to $\mathcal{H}_b$ with input $u_b = y_a$.

1. $\operatorname{dom} x_b = \operatorname{dom} u_b^r$ by definition.
2. $x_{a,cl}$ and $u_{a,cl}$ are j-reparametrizations of x_a and u_a^r with reparametrization map ρ_b by construction. Besides, since x_a and u_a^r are built from $x_{a,cl}$ and $u_{a,cl}$ only, the corresponding j-reparametrizations are full. Therefore, $\operatorname{dom}_t \phi_{cl} = \operatorname{dom}_t x_a = \operatorname{dom}_t y_a$ and in particular, $T(\phi_{cl}) = T(y_a)$. From CC.5), we get $\operatorname{card} \mathcal{J}_{T(\phi_{cl})}(\phi_{cl}) = \operatorname{card} \mathcal{J}_{T(y_a)}(y_a)$ by observing that by construction $\operatorname{card} \mathcal{J}_{T(y_a)}(y_a) = n_{x_a}$.
3. The flow condition holds by definition of C_{cl} and F_{cl}.
4. As for the jump condition, item 4) is given by the definition of D_{cl} and G_{cl}, by CC.1) which imposes that the jumps of $u_b = y_a$ happen successively for $j < j_0 + n_{x_a}$, and by CC.3) at $j = j_0$ when $t > 0$.
5. $y_b = h_b(x_b, u_b^r)$ by definition.

□

AUTHOR BIOGRAPHY

Pauline Bernard graduated in Applied Mathematics from MINES ParisTech in 2014. She joined the Systems and Control Center of MINES ParisTech and obtained her Ph.D. in Mathematics and Control from PSL Research University in 2017. For her work on Observer design for nonlinear systems, she obtained the European PhD award on Control for Complex and Heterogenous Systems 2018. As a post-dotoral scholar, she then visited the Hybrid Systems Lab at the University California Santa Cruz, USA, and the Center for Research on Complex Automated Systems at the University of Bologna, Italy. In 2019, she became an assistant professor at the Systems and Control Center of MINES ParisTech, PSL Research University, France. Her research interests include the observation and output regulation of nonlinear and hybrid systems.



Ricardo G. Sanfelice received the B.S. degree in Electronics Engineering from the Universidad de Mar del Plata, Buenos Aires, Argentina, in 2001, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of California, Santa Barbara, CA, USA, in 2004 and 2007, respectively. In 2007 and 2008, he held postdoctoral positions at the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology and at the Centre Automatique et Systèmes at the École de Mines de Paris. In 2009, he joined the faculty of the Department of Aerospace and Mechanical Engineering at the University of Arizona, Tucson, AZ, USA, where he was an Assistant Professor. In 2014, he joined the University of California, Santa Cruz, CA, USA, where he is currently Professor in the Department of Electrical and Computer Engineering. Prof. Sanfelice is the recipient of the 2013 SIAM Control and Systems Theory Prize, the National Science Foundation CAREER award, the Air Force Young Investigator Research Award, and the 2010 IEEE Control Systems Magazine Outstanding Paper Award. His research interests are in modeling, stability, robust control, observer design, and simulation of nonlinear and hybrid systems with applications to power systems, aerospace, and biology.