An Algorithm to Generate Solutions to Hybrid Dynamical Systems with Inputs

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

Abstract— **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. 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 and discrete systems with inputs. We provide an algorithm that automatically performs the construction of solutions for a given hybrid input. Examples illustrate the notions and algorithm.**

I. Introduction

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 [1], [2], have been rendered useful to study interconnection of continuous-time systems via small gain theorems. Similarly, the so-called output-to-state stability notion is convenient to bound the solutions by a function of the output of the system [3]. Input-output-to-state stability combines these two properties to provide bounds that depend on the inputs and outputs of the single and multiple systems [4].

Defining solutions to continuous-time systems with continuous inputs or to discrete-time systems with discrete inputs does not raise any critical problems, besides perhaps making sure that, when the domain of the input is bounded, the state trajectory is defined at least over that interval of time. On the other hand, defining solutions to hybrid systems with hybrid inputs is much more challenging since, in principle, the domain of the input does not necessarily match that of the resulting state trajectory. In previous works involving hybrid systems with inputs (see, e.g., [5], [6]), the notion of solution assumes that the domain of the input and of the state trajectory are the same. 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 [7], [8], [9], [10], and assumed known. Therefore, the definition of solutions in those cases is straightforward.

The assumption that the domain of the input and of the state trajectory coincide relies on a pre-processing stage of the hybrid input signal to make the domains match. However, when such an assumption is applied to interconnections of hybrid systems, it requires altering the domain of the output of another hybrid system. As pointed out in [11] such a modification is far from trivial. In fact, serious difficulties appear 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 an 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 [12], 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. 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 [13], the analysis is done thanks to a timer which is used to model the jumps of the input and by building a closed-loop system containing the timer.

In this paper, we propose to define solutions to hybrid systems when the input is a hybrid arc with its own do-

This research has been partially supported by the National Science Foundation under CAREER Grant no. ECS-1450484, Grant no. ECS-1710621, and Grant no. CNS-1544396, by the Air Force Office of Scientific Research under Grant no. FA9550-16-1-0015, by the Air Force Research Laboratory under Grant no. FA9453-16-1-0053, and by CITRIS and the Banatao Institute at the University of California.

[1]Pauline Bernard (pauline.bernard@unibo.it) is with the Center for Research on Complex Automated Systems in the Department of Electronics, Computer Science and Systems (D.E.I) of University of Bologna.

[2]Ricardo Sanfelice (ricardo@ucsc.edu) is with the Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064, USA.

main, which does not necessarily match the one of the produced system trajectory. This is done in Section II via a reparametrization of the input jumps. Then, we provide in Section III an implementable algorithm to produce such solutions. We finally illustrate those notions in examples in Section IV.

II. 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, which typically coincide or a domain truncation is performed a priori. A notion of solution for discrete-time systems with inputs can be defined similarly.

As pointed out in Section I, 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 notations.

Definition 2.1 (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 $E \cap ([0, T'] \times \{0, 1, \ldots, J'\})$ can be written as $\bigcup_{j=0}^{J-1} ([t_j, t_{j+1}], j)$ for some finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \ldots \leq t_J$ and $J \in \mathbb{N}$.

Definition 2.2 (hybrid arc): A function $\phi : \mathrm{dom}\,\phi \to \mathbb{R}^n$ is a hybrid arc if $\mathrm{dom}\,\phi$ is a hybrid time domain and for each $j \in \mathbb{N}$, $t \mapsto \phi(t,j)$ is locally absolutely continuous on $\{t \ : \ (t,j) \in \mathrm{dom}\,\phi \ \}$.

Notation For a set $\mathcal{S}$, $\mathrm{cl}(\mathcal{S})$ will denote its closure, and $\mathrm{card}\,\mathcal{S}$ its cardinality (possibly infinite). For a hybrid arc $(t,j) \mapsto \phi(t,j)$ defined on a hybrid time domain $\mathrm{dom}\,\phi$, we denote $\mathrm{dom}_t\,\phi$ (resp.$\mathrm{dom}_j\,\phi$) 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 $(t_j(\phi), j) \in \mathrm{dom}\,\phi$ and $(t_j(\phi), j-1) \in \mathrm{dom}\,\phi$), and $\mathcal{I}_j(\phi)$ the largest interval such that $\mathcal{I}_j(\phi) \times \{j\} \subseteq \mathrm{dom}\,\phi$. We define also $\mathcal{T}(\phi) = \{t_j(\phi) : j \in \mathrm{dom}_j\,\phi \cap \mathbb{N}_{>0}\}$ as the set of jump times, $T(\phi) = \sup \mathrm{dom}_t\,\phi \in \mathbb{R}_{\geq 0} \cup \{+\infty\}$ the maximal time of the domain, $J(\phi) = \sup \mathrm{dom}_j\,\phi \in \mathbb{N} \cup \{+\infty\}$ 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 $\mathrm{card}\,\mathcal{J}_t(\phi)$ is the number of jumps of ϕ occurring at time t.

A. *j-reparametrization of hybrid arcs*

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

Definition 2.3: 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

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

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

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

or, equivalently, $\mathrm{dom}_t\,\phi = \mathrm{dom}_t\,\phi^r$ 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.

Example 2.4: Consider the hybrid arc ϕ defined on $\mathrm{dom}\,\phi = \mathbb{R} \times \{0\}$ by

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

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

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

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 $\mathrm{dom}\,\phi = ([0,1] \times \{0\}) \cup ([1,2] \times \{1\})$ by

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

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 $\mathrm{dom}\,\phi^r = ([0,1/2] \times \{0\}) \cup ([1/2,1] \times \{1\}) \cup ([1,2] \times \{2\})$ by

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

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. △

B. Solutions to hybrid systems with hybrid inputs

Consider the hybrid system

$$\mathcal{H}\left\{\begin{array}{rcll} \dot{x} & \in & F(x,u) & (x,u)\in C \\ x^+ & \in & G(x,u) & (x,u)\in D \\ \\ y & = & h(x,u) & \end{array}\right. \tag{4}$$

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 2.5: 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, with $\operatorname{card}\mathcal{J}_{T(u)}(\phi)=\operatorname{card}\mathcal{J}_{T(u)}(u)$ if this reparametrization is full.
3) for all $j\in\mathbb{N}$ such that $\mathcal{I}_j(\phi)$ has nonempty interior,
$$\begin{array}{c}(x(t,j),u^r(t,j))\in C \quad \forall t\in \operatorname{int}\mathcal{I}_j(\phi)\\ \dot{x}(t,j)\in F(x(t,j),u^r(t,j)) \quad \text{for a.a. } t\in\mathcal{I}_j(\phi)\end{array}$$
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 $t>0$,
      - $(x(t,j_0-1),u^r(t,j_0-1))\in C\cup D$
      - $x(t,j_0)\in G_e^0(x(t,j_0-1),u^r(t,j_0-1))$

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

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

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

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 2.3, $\operatorname{dom}_t\phi=\operatorname{dom}_t u$ (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_e^0.
- for the remaining jumps of u, or if $t=0$, those conditions are relaxed, 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 $D\setminus\operatorname{cl}(C)$. 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 $\operatorname{cl}(C)\cup D$. This distinction disappears if C is closed. Note that more generally, the solution stops if ϕ leaves $\operatorname{cl}(C)\cup D$.

Remark 2.6: 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. It enables to recover the definition of solutions of discrete systems with input if $F\equiv\emptyset$ and $C=\emptyset$. Removing this constraint 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 $j\in\mathcal{J}_t(\phi)$, either
$$\left\{\begin{array}{l}(x(t,j-1),u^r(t,j-1))\in\operatorname{cl}(C)\cup D\\ x(t,j)\in G_e(x(t,j-1),u^r(t,j-1))\\ \\ \rho_u(j)=\rho_u(j-1)+1\end{array}\right.$$
or
$$\left\{\begin{array}{l}(x(t,j-1),u^r(t,j-1))\in D\\ x(t,j)\in G(x(t,j-1),u^r(t,j-1))\\ \\ \rho_u(j)=\rho_u(j-1)\end{array}\right. ,$$

with $\operatorname{cl}(C)$ replaced by C for $j=j_0$ if $t>0$. With this alternate definition, it would no longer make sense to require card $\mathcal{J}_{T(u)}(\phi) = \text{card}\,\mathcal{J}_{T(u)}(u)$ 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 with a common time stamp model 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 2.7: 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 IV-B, 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 2.8: In the case where $\text{dom}\, x = \text{dom}\, u$ is assumed from the start as in [5], u^r is equal to u and Conditions 1) and 2) in Definition 2.5 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. $\text{cl}(C)$) (see examples in Section IV).

III. An Algorithm to Generate Solutions to Hybrid System with Hybrid Inputs

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 3.1: 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 2.5 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 $\text{dom}\, x$ by

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

and x is simply characterized by

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

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

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

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

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

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

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 now present Algorithm III.1 which constructs maximal solutions to $\mathcal{H}$ with a hybrid input u according to Definition 2.5.

Proposition 3.2: 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, y and $\mathcal{D}$ are possible outputs of Algorithm III.1 with input u.

The algorithm operates 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 3.1. This gives Conditions 3) and 4b). $\underline{x}$ is appended to our solution x.
3) If (line 20) $\underline{x}$ ends before reaching the end of the interval I_u, or ends outside of $\text{cl}(C) \cup D$ (resp. $C \cup D$ after flow, namely if $T_m > 0$ 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).

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 3.1, and through the set-valued jump maps G, G_e^0 and G_e.

IV. Examples

We consider here a series interconnection of two hybrid systems $\mathcal{H}_1$ and $\mathcal{H}_2$, where the output of $\mathcal{H}_1$ is the input to $\mathcal{H}_2$. Suppose we want to use the output of $\mathcal{H}_1$ to make $\mathcal{H}_2$ jump according to its jump map whenever $\mathcal{H}_1$ does. We will consider two settings:

- "Jump triggering": the information of the jumps of $\mathcal{H}_1$ is contained in the output of $\mathcal{H}_1$ *before* they happen,

Algorithm III.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: $\quad\quad \mathcal{D} \leftarrow \mathcal{D} \cup \Big(\{(t_j, j)\} + \operatorname{dom} \underline{x}\Big)$
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: $\quad\quad \rho_u(j + \underline{j}) \leftarrow j_u \quad \forall \underline{j} \in \{0, 1, \ldots, \underline{j}_m\} \cap \mathbb{N}$
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: $J \leftarrow \sup_j \mathcal{D}$ $\quad\triangleright$ 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

namely we would like to make $\mathcal{H}_2$ jump synchronously with $\mathcal{H}_1$;

- "Jump detection": the information of the jumps of $\mathcal{H}_1$ can be detected in the output of $\mathcal{H}_1$ *after* they have happened, namely we would like to make $\mathcal{H}_2$ jump right after $\mathcal{H}_1$.

A. *Jump triggering*

We model the first situation with a resettable timer defined by

$$\mathcal{H}_1 \left\{ \begin{array}{rcll} \dot{\tau} & = & -1 & \tau \in C_1 := [0, \sup \mathcal{I}] \cap \mathbb{R} \\ \tau^+ & \in & \mathcal{I} & \tau \in D_1 := \{0\} \\ & & & \\ y & = & \tau & \end{array} \right. \tag{5}$$

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 $\tau = 0$, we know $\mathcal{H}_1$ is going to jump when its output y reaches 0. Consider a hybrid system $\mathcal{H}_2$ with input given by the hybrid output y of (5), namely

$$\mathcal{H}_2 \left\{ \begin{array}{rcll} \dot{x} & \in & F(x, y) & (x, y) \in C_2 \\ x^+ & \in & G(x, y) & (x, y) \in D_2 \end{array} \right. \tag{6}$$

with

$$C_2 = \mathbb{R}^n \times ((0, \sup \mathcal{I}] \cap \mathbb{R})) \quad , \quad D_2 = \mathbb{R}^n \times \{0\} \; . \tag{7}$$

Let us build solutions to $\mathcal{H}_2$ according to Definition 2.5. Suppose first $\tau(0,0) \in C_1 \setminus \{0\}$. Then $\mathcal{H}_1$ flows for $t \in [0, t_1]$, with $t_1 > 0$ and $\tau(t_1, 0) = 0$. Since $y = \tau$, $(x, y)(t, 0) \in C_2 \setminus D_2$ for $t \in [0, t_1)$, so $\mathcal{H}_2$ flows too, and $y^r := y$ on $[0, t_1] \times \{0\}$. At $t = t_1$, $\mathcal{H}_1$ jumps, with y reset to a value in $\mathcal{I}$: if this value is non-zero, $\mathcal{H}_1$ jumps only once, namely $\mathcal{J}_{t_1}(y) = \{1\}$ and $n_y = 1$; otherwise, consecutive jumps happen with $\mathcal{J}_{t_1}(y) = \{1, 2, \ldots\}$ until y becomes non-zero. Since $(x, y)(t_1, 0) \in D_2 \setminus C_2$, $\mathcal{H}_2$ jumps too: x is reset to a point in $G(x(t_1, 0), 0)$ according to G_e^0 in the first part of Condition 4a) 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 $y(t_1, 1) \neq 0$, so that $n_y = 1$, and $\mathcal{H}_1$ flows for $t \in [t_1, t_2]$, with $t_2 > t_1$ and $y(t_2, 1) = 0$. Since $(x, y)(t_1, 1) \in C_2 \setminus D_2$, $\mathcal{H}_2$ cannot jump according to Condition 4b) with $j = 2 \geq j_0 + n_y$, so that $\mathcal{H}_2$ flows and we start again with the same reasoning.
- or $y(t_1, 1) = 0$ (if $0 \in \mathcal{I}$), so that $n_y \geq 2$, $\mathcal{H}_1$ jumps again, with $y(t_1, 2) \in \mathcal{I}$. Since $(x, y)(t_1, 1) \in D_2 \cap \operatorname{cl}(C_2)$, $\mathcal{H}_2$ jumps to $x(t_1, 2) \in \{x(t_1, 1)\} \cup G(x(t_1, 1), 0)$ according to the second part of Condition 4a) with $j = 2 < j_0 + n_y$. We thus 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 $\tau(0, 0) = 0$, $\mathcal{H}_1$ starts with a jump. Since $t = 0$ and $(x, y)(0, 0) \in D_2 \cap \operatorname{cl}(C_2)$, the second part of Condition 4a) with $j = 1 = j_0$ says $\mathcal{H}_2$ jumps to $x(0, 1) \in \{x(0, 0)\} \cup G(x(0, 0), 0)$. Then, we carry on with the same reasoning in the bullets above.

So we conclude that $\mathcal{H}_2$ jumps only when $\mathcal{H}_1$ jumps and inherits the domain of y, so that $y^r = y$. Besides, if $0 \notin \mathcal{I}$, $\mathcal{H}_2$ jumps according to G every time $\mathcal{H}_1$ jumps, except maybe at $t = 0$ where one trivial jump is allowed. To ensure this, the first part of Condition 4a) was crucial to force x to be reset to a point in $G(x, 0)$ when $(x, y) \in D_2 \setminus C_2$. If we had used G_e instead of G_e^0, trivial jumps would have been allowed at $(x,y) \in D_2 \cap \mathrm{cl}(C_2)$. On the other hand, if $0 \in \mathcal{I}$ and $\mathcal{H}_1$ jumps several times consecutively, trivial jumps are allowed by G_e after the first jump, thus losing the property of jump triggering.

B. Jump detection

Now consider rather a timer defined by

$$\mathcal{H}_1 \left\{ \begin{array}{rcll} \dot{\tau} & = & 1 & \tau \in C_1 := [0, \sup \mathcal{I}] \cap \mathbb{R} \\ \tau^+ & = & 0 & \tau \in D_1 := \mathcal{I} \\ & & & \\ y & = & \tau & \end{array} \right. \tag{8}$$

It can create the same time domains as (5), but this time the information of its jumps is encoded in the output only after they have happened, namely when y has been reset to 0.

A first idea to make $\mathcal{H}_2$ jump according to G following the jumps of $\mathcal{H}_1$, could be to define $\mathcal{H}_2$ as in (6) with flow/jump sets defined in (7), but with input governed by (8) instead of (5). Suppose $\tau(0,0) \in C_1 \setminus \{0\}$, $\mathcal{H}_1$ flows for $t \in [0, t_1]$ with $\tau(t_1, 0) \in \mathcal{I}$ and jumps at time t_1. For $t \in [0, t_1]$, $(x,y)(t,0) \in C_2 \setminus D_2$, so $\mathcal{H}_2$ flows too, and $y^r := y$. At $t = t_1$, $\mathcal{H}_1$ jumps, with y reset to 0. Since $(x,y)(t_1,0) \in C_2 \setminus D_2$, $\mathcal{H}_2$ jumps too, with $x(t_1,1) = x(t_1,0)$ according to G_e^0 in the first part of Condition 4a) and we take $y^r(t_1,1) = y(t_1,1) = 0$. At this point, $\mathcal{H}_1$ flows again with y leaving 0, and $(x,y)(t_1,1) \in D_2 \cap \mathrm{cl}(C_2)$. Therefore, $\mathcal{H}_2$ can either flow into C_2 according to Condition 3), or jump with x reset in $G(x(t_1,1),0)$ according to Condition 4b). In that later case, a trivial jump is added in y^r, namely $y^r(t_1,2) = y(t_1,1) = 0$, and $\mathcal{H}_2$ can still either flow or jump. We conclude that with this definition $\mathcal{H}_2$ can jump any number of times according to $G(x,0)$ after each jump of $\mathcal{H}_1$. This is not what we want.

In fact, the problems we have met are twofold: 1) (x, y^r) should not be in $\mathrm{cl}(C_2)$ after the jumps of $\mathcal{H}_1$, otherwise flow is allowed before $\mathcal{H}_2$ has jumped using G; 2) after a jump of $\mathcal{H}_2$ using G, (x, y^r) should no longer be in D_2 unless $\mathcal{H}_1$ jumps again, otherwise further jumps of $\mathcal{H}_2$ are allowed. In the case where $\min \mathcal{I} > 0$, a possible solution is to add a state $\hat{\tau}$ to $\mathcal{H}_2$ in the following way:

$$\mathcal{H}_2 \left\{ \begin{array}{rcll} \dot{x} & \in & F(x,y) & (x,\hat{\tau},y) \in C_2 \\ \dot{\hat{\tau}} & = & 1 & \\ & & & \\ x^+ & \in & G(x,y) & (x,\hat{\tau},y) \in D_2 \\ \hat{\tau}^+ & = & y & \end{array} \right. \tag{9}$$

with

$$C_2 = \Big\{ (x,\hat{\tau},y) \in \mathbb{R}^n \times \mathbb{R}_{\geq 0} \times ([0, \sup \mathcal{I}] \cap \mathbb{R}) \ : \ |\hat{\tau} - y| \leq \frac{\varepsilon}{2} \Big\} \tag{10a}$$

$$D_2 = \{ (x,\hat{\tau},y) \in \mathbb{R}^n \times \mathbb{R}_{\geq 0} \times ([0, \sup \mathcal{I}] \cap \mathbb{R}) \ : \ |\hat{\tau} - y| \geq \varepsilon \} \tag{10b}$$

and $0 < \varepsilon < \frac{\min \mathcal{I}}{2}$. Assume $\mathcal{H}_2$ is initialized in C_2. Then, $(x,\hat{\tau},y)$ remains in $C_2 \setminus D_2$ defined in (10a)-(10b) as long as $\mathcal{H}_1$ flows, so $\mathcal{H}_2$ flows too. At a time $t_1 \geq 0$, $\mathcal{H}_1$ jumps, $(x,\hat{\tau})(t_1,1) = (x,\hat{\tau})(t_1,0)$ and $y^r(t_1,1) = y(t_1,1) = 0$ according to Condition 4a). Since $y^r(t_1,0) \in D_1 = \mathcal{I}$, $|\hat{\tau}(t_1,1) - y^r(t_1,1)| = |\hat{\tau}(t_1,1)| \geq 2\varepsilon - \frac{\varepsilon}{2} \geq \varepsilon$, so that $(x,\hat{\tau},y^r)(t_1,1) \in D_2 \setminus \mathrm{cl}(C_2)$. Therefore, $\mathcal{H}_2$ cannot flow and from Condition 4b), $(x,\hat{\tau})(t_1,2) \in (G(x(t_1,1),0),0)$ and $y^r(t_1,2) = y(t_1,1) = 0$. Therefore, $(x,\hat{\tau},y^r)(t_1,2) \in C_2 \setminus D_2$, both $\mathcal{H}_1$ and $\mathcal{H}_2$ flow, and we can apply again the same reasoning. Assume now $|\hat{\tau}(0,0) - y(0,0)| \geq \varepsilon$. Then, $(x,\hat{\tau},y)(0,0) \in D_2 \setminus \mathrm{cl}(C_2)$, so that $\mathcal{H}_2$ necessarily jumps. Whether $\mathcal{H}_1$ jumps at $t = 0$ or not, $(x,\hat{\tau})(0,1) \in (G((x,y)(0,0)), y(0,0))$ and $y^r(0,1) = y(0,0)$ from Condition 4). Therefore, we recover $|\hat{\tau}(0,1) - y^r(0,1)| \leq \frac{\varepsilon}{2}$ and we can apply the previous reasoning from an initial condition in C_2. We conclude that $\mathcal{H}_2$ detects the jumps of $\mathcal{H}_1$ and jumps according to G right after it, if $0 \notin \mathcal{I}$.

Note that in the case where $0 \in \mathcal{I}$, such a detection mechanism can also be built but the information of the jump of $\mathcal{H}_1$ needs to be encoded in a discrete state q toggled at each jump of $\mathcal{H}_1$, instead of the continuous τ, to prevent (x,y) from being in $\mathrm{cl}(C_2)$ after the jump of $\mathcal{H}_1$.

V. Conclusion

We have shown how solutions to hybrid system with inputs can be defined when the input is an hybrid arc whose domain does not match that of the solution. This work is instrumental in defining cascade interconnections of hybrid systems, and in particular, observers for hybrid systems.

References

[1] Z.-P. Jiang, A. R. Teel, and L. Praly. Small-gain theorem for ISS systems and applications. *Math. Control Signals Syst.*, 7:95–120, 1994.

[2] A. R. Teel. A nonlinear small gain theorem for the analysis of control systems with saturation. *IEEE Transactions on Automatic Control*, 41(9):1256–1270, 1996.

[3] E. D. Sontag and Y. Wang. Output-to-state stability and detectability of nonlinear systems. *Systems & Control Letters*, 29:279–290, 1997.

[4] S. N. Dashkovskiy, D. V. Efimov, and E. D. Sontag. Input to state stability and allied system properties. *Automation and Remote Control*, 72(8):1579–1614, 2011.

[5] C. Cai and A. R. Teel. Characterizations of input-to-state stability for hybrid systems. *Syst. & Cont. Letters*, 58:47–53, 2009.

[6] D. Nesic and A.R. Teel. Input-output stability properties of networked control systems. 49:1650–1667, 2004.

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

[8] E. A. Medina and D. A. Lawrence. State estimation for linear impulsive systems. *Annual American Control Conference*, pages 1183–1188, 2009.

[9] R. G. Sanfelice, J. J. B. Biemond, N. van de Wouw, and W. P. M. H. Heemels. An embedding approach for the design of state-feedback tracking controllers for references with jumps. *International Journal of Robust and Nonlinear Control*, 24(11):1585–1608, 2013.

[10] P. Bernard and R.G. Sanfelice. Observers for hybrid dynamical systems with linear maps and known jump times. *IEEE Conference on Decision and Control*, 2018.

[11] R. G. Sanfelice. Interconnections of hybrid systems: Some challenges and recent results. *Journal of Nonlinear Systems and Applications*, 2(1-2):111–121, 2011.

[12] B. Biemond, N. van de Wouw, M.H. Heemels, and H. Nijmeijer. Tracking control for hybrid systems with state-triggered jumps. *IEEE Transactions on Automatic Control*, 58(4):876–890, 2013.

[13] F. Forni, A. R. Teel, and L. Zaccarian. Follow the bouncing ball : global results on tracking and state estimation with impacts. *IEEE Transactions on Automatic Control*, 58(6):1470–1485, 2013.