# EE291E Lecture Notes 3
# Autonomous Hybrid Automata

Claire J. Tomlin

January 28, 2021

We develop definitions to formalize the notion of hybrid systems as dynamical systems that describe the evolution in time of the values of a set of discrete and continuous variables. In this chapter, we restrict our attention to autonomous hybrid automata, that is time invariant hybrid systems without inputs. In subsequent chapters we will extend this class of hybrid models and introduce input variables to allow us to state and solve control problems.

# 1 Hybrid Time Sets and Trajectories

Since we are interested in hybrid phenomena that involve both continuous and discrete dynamics, we introduce the hybrid time trajectory, which will encode the set of times over which the evolution of the system is defined. Hybrid time sets will be used to define the time horizon of executions of hybrid systems.

**Definition 1 (Hybrid Time Set)** *A hybrid time set is a finite or infinite sequence of intervals $\tau = \{I_i\}_{i=0}^{N}$, such that*

- $I_i = [\tau_i, \tau_i']$ *for all* $i < N$;
- *if* $N < \infty$ *then either* $I_N = [\tau_N, \tau_N']$ *or* $I_N = [\tau_N, \tau_N')$; *and*
- $\tau_i \leq \tau_i' = \tau_{i+1}$ *for all* $i$.

An example of a hybrid time trajectory is given in Figure 1. We will use $\mathcal{T}$ to denote the set of all hybrid time sets.

Consider $\tau = \{I_i\}_{i=0}^{N} \in \mathcal{T}$. Notice that, for $i < N$, the right endpoint, $\tau_i'$, of the interval $I_i$ coincides with the left endpoint, $\tau_{i+1}$ of the interval $I_{i+1}$. The interpretation is that these are the times at which discrete transitions of the hybrid system take place. $\tau_i'$ corresponds to the time instant just before a discrete transition, whereas $\tau_{i+1}$ corresponds to the time instant just after the discrete transition. Discrete transitions are assumed to be instantaneous, therefore $\tau_i' = \tau_{i+1}$. The advantage of this convention is that it allows one to capture situations in which multiple discrete transitions take place one after the other at the same time instant, since it is possible for $\tau_{i-1}' = \tau_i = \tau_i' = \tau_{i+1}$.

For each hybrid time set, $\tau \in \mathcal{T}$, we define a precedence relation $\prec$ on $\tau$. For $t_1 \in [\tau_i, \tau_i'] \in \tau$ and $t_2 \in [\tau_j, \tau_j'] \in \tau$ we say that $t_1$ *precedes* $t_2$ (denoted by $t_1 \prec t_2$) if $t_1 < t_2$ or $i < j$. It is easy to show that this relation is a linear order on $\tau$. In Figure 1, we have $t_1 \prec t_2 \prec t_3 \prec t_4 \prec t_5 \prec t_6$.

On the set $\mathcal{T}$ of hybrid time sets we define a *prefix* relation. We say that $\tau = \{I_i\}_{i=0}^{N}$ is a prefix of $\hat{\tau} = \{\hat{I}_i\}_{i=0}^{M}$ (and write $\tau \sqsubseteq \hat{\tau}$) if either they are identical, or $\tau$ is finite, $N \leq M$, $I_i = \hat{I}_i$ for all $i = 0, \ldots, N-1$, and $I_N \subseteq \hat{I}_N$. We say that $\tau$ is a *strict prefix* of $\hat{\tau}$ (and write $\tau \sqsubset \hat{\tau}$) if $\tau \sqsubseteq \hat{\tau}$ and $\tau \neq \hat{\tau}$. It is easy to show that the prefix relation is a partial order on $\mathcal{T}$. In Figure 2, $\tau$ is a strict prefix of both $\hat{\tau}$ and $\tilde{\tau}$, but $\hat{\tau}$ is not a prefix of $\tilde{\tau}$ and $\tilde{\tau}$ is not a prefix of $\hat{\tau}$.

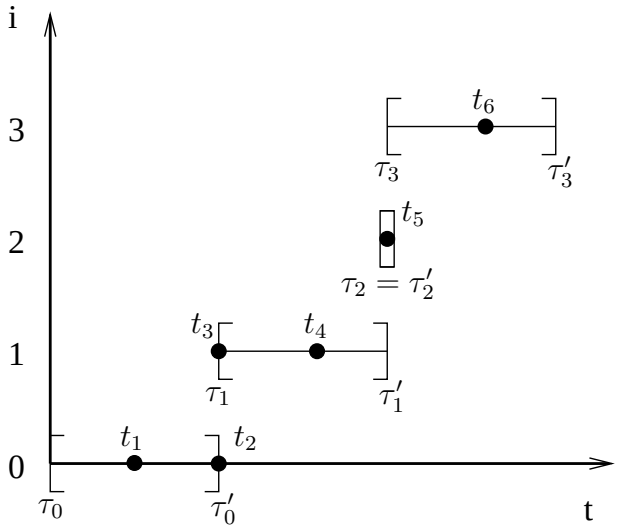


Figure 1: A hybrid time set $\tau = \{[\tau_i, \tau_i']\}_{i=0}^{3}$.

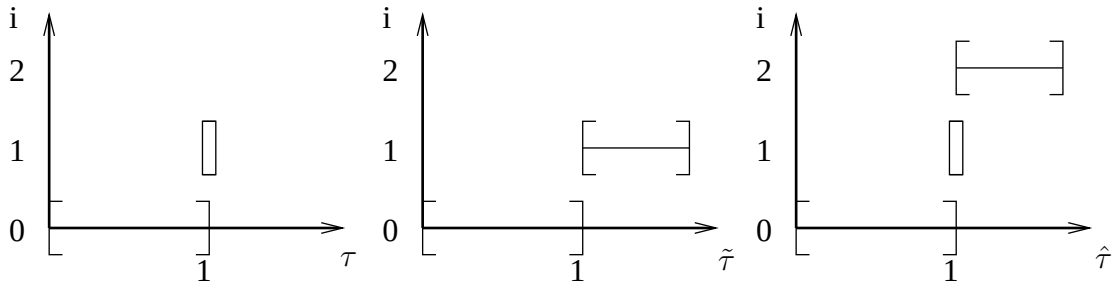


Figure 2: $\tau \sqsubset \hat{\tau}$ and $\tau \sqsubset \tilde{\tau}$.

Since hybrid time sets incorporate both discrete and continuous aspects one can consider two notions of "length" for them. One notion is discrete and reflects the number of intervals contained in the hybrid time set. The *discrete extent* can be defined as a function $\langle \cdot \rangle : \mathcal{T} \to \mathbb{N} \cup \{\infty\}$ that for $\tau \in \mathcal{T}$ returns

$$\langle \tau \rangle = \begin{cases} N & \text{if } \tau \text{ is a finite sequence} \\ \infty & \text{if } \tau \text{ is an infinite sequence} \end{cases}$$

The second notion of length is continuous, and reflects the length of time over which the hybrid time set is defined. More formally, the *continuous extent* can be defined as a function $\| \cdot \| : \mathcal{T} \to \mathbb{R}_+$ that for $\tau = \{I_i\}_{i=0}^{N} \in \mathcal{T}$ returns

$$\|\tau\| = \sum_{i=0}^{N} (\tau_i' - \tau_i)$$

Clearly, if $\tau \sqsubseteq \hat{\tau}$, $\langle \tau \rangle \leq \langle \hat{\tau} \rangle$ and $\|\tau\| \leq \|\hat{\tau}\|$.

**Definition 2 (Classification of Hybrid Time Sets)** *A hybrid time set $\tau \in \mathcal{T}$ is called*

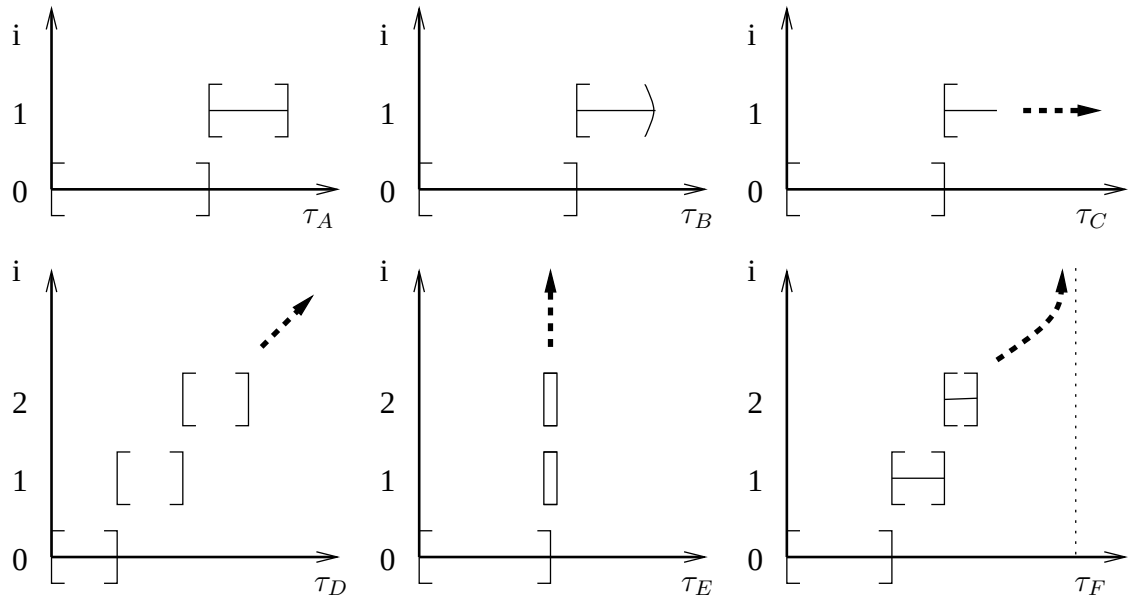


Figure 3: $\tau_A$ finite, $\tau_B$ finite-open, $\tau_C$ and $\tau_D$ infinite, $\tau_E$ and $\tau_F$ Zeno.

- finite, *if* $\langle\tau\rangle$ *is finite and the last interval in* $\tau$ *is closed;*
- finite-open, *if* $\langle\tau\rangle$ *is finite, and the last interval in* $\tau$ *is bounded and right open;*
- infinite, *if* $\langle\tau\rangle = \infty$, *or if* $\|\tau\| = \infty$*;*
- Zeno, *if it is infinite but* $\|\tau\| < \infty$.

Figure 3 shows examples of finite, finite open, infinite and Zeno hybrid time sets

**Definition 3 (Hybrid Trajectory)** *A hybrid trajectory* $(\tau, q, x)$ *consists of a hybrid time set* $\tau = \{I_i\}_0^N \in \mathcal{T}$ *and two sequences of functions* $q = \{q_i(\cdot)\}_0^N$ *and* $x = \{x_i(\cdot)\}_0^N$ *with* $q_i(\cdot) : I_i \to Q$ *and* $x(\cdot) : I_i \to \mathbb{R}^n$.

# 2 Autonomous Hybrid Automata

## 2.1 Fundamental Definitions

**Definition 4 (Autonomous Hybrid Automaton)** *An autonomous hybrid automaton* $H$ *is a collection* $H = (Q, X, \text{Init}, f, \text{Dom}, R)$, *where*

- $Q = \{q_1, q_2, \ldots\}$ *is a set of* discrete states*;*
- $X = \mathbb{R}^n$ *is a set of* continuous states*;*
- $\text{Init} \subseteq Q \times X$ *is a set of* initial states*;*

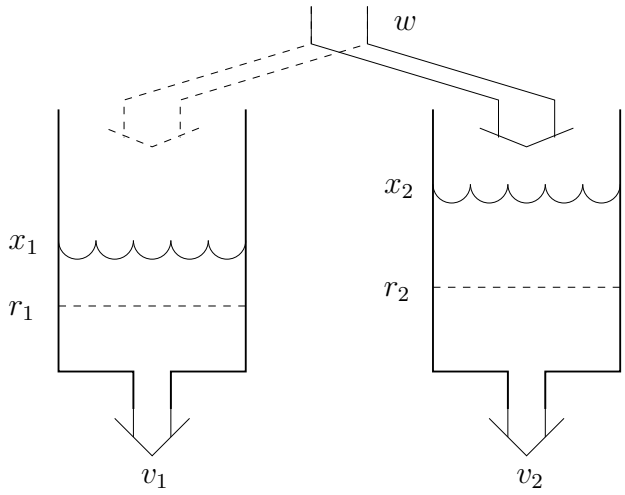


Figure 4: The water tank system.

- $f(\cdot,\cdot) : Q \times X \to \mathbb{R}^n$ *is a* vector field*;*
- $\mathrm{Dom}(\cdot) : Q \to 2^X$ *is a* domain*;*
- $R(\cdot,\cdot) : Q \times X \to 2^X$ *is a* reset relation*.*

Recall that $2^X$ denotes the power set (set of all subsets) of $X$. We refer to $(q, x) \in Q \times X$ as the state of $H$.

Roughly speaking, autonomous hybrid automata define possible ways that their state can evolve over time. These possible evolutions take the form of hybrid trajectories, $(\tau, q, x)$. Starting from an initial value $(q_0, x_0) \in \mathrm{Init}$, the continuous state, $x$, flows according to the differential equation $\dot{x} = f(q_0, x), x(0) = x_0$, while the discrete state, $q$, remains constant at $q_0$. Continuous evolution can go on as long as $x(t) \in \mathrm{Dom}(q_0)$. If at some point $R(q_0, x) \neq \emptyset$ a discrete transition can take place. During a discrete transition both the continuous and the discrete state may be reset to some value in $R(q_0, x)$. After the discrete transition, continuous evolution resumes and the whole process is repeated.

Before we give the formal definition of this process we give an example to illustrate how autonomous hybrid automata can be used to model physical systems.

**Example (Water Tank)** The two tank system, shown in Figure 4, consists of two tanks containing water. Both tanks are leaking at a constant rate. Water is added at a constant rate to the system through a hose, which at any point in time is dedicated to either one tank or the other. It is assumed that the hose can switch between the tanks instantaneously.

For $i \in \{1, 2\}$, let $x_i$ denote the volume of water in Tank $i$ and $v_i > 0$ denote the constant flow of water out of Tank $i$. Let $w$ denote the constant flow of water into the system. The objective is to keep the water volumes above $r_1$ and $r_2$, respectively, assuming that the water volumes are above $r_1$ and $r_2$ initially. This is to be achieved by a controller that switches the inflow to Tank 1 whenever $x_1 \leq r_1$ and to Tank 2 whenever $x_2 \leq r_2$.

It is straightforward to define an autonomous hybrid automaton to describe this process:

- $Q = \{q_1, q_2\}$;
- $X = \mathbb{R}^2$;
- $Init = Q \times \{x \in \mathbb{R}^2 \mid x_1 \geq r_1 \wedge x_2 \geq r_2\}$;
- $f(q_1, x) = (w - v_1, -v_2)$ and $f(q_2, x) = (-v_1, w - v_2)$;
- $\mathrm{Dom}(q_1) = \{x \in \mathbb{R}^2 \mid x_2 \geq r_2\}$ and $\mathrm{Dom}(q_2) = \{x \in \mathbb{R}^2 \mid x_1 \geq r_1\}$;
- $R(q_1, x) = (q_2, x)$ if $x_2 \leq r_2$, $R(q_2, x) = (q_1, x)$ if $x_1 \leq r_1$ and $R(q, x) = \emptyset$ otherwise.

Though very simple, the water tank hybrid automaton has a number of interesting properties, and will be used throughout this section (and beyond) to illustrate various concepts. ∎

An execution of an autonomous hybrid automaton is a hybrid trajectory, $(\tau, q, x)$, over its state variables. The elements listed in Definition 4 impose restrictions on the elements of the set of possible trajectories that the hybrid automaton finds "acceptable".

**Definition 5 (Execution)** *An execution of a hybrid automaton $H$ is a hybrid trajectory, $(\tau, q, x)$, which satisfies the following conditions:*

- initial condition*:* $(q_0, x_0) \in \mathrm{Init}$
- discrete evolution*:* $(q_{i+1}(\tau_{i+1}), x_{i+1}(\tau_{i+1})) \in R(q_i(\tau_i'), x_i(\tau_i'))$*;*
- continuous evolution*: for all $i$,*
  1. $q_i(\cdot) : I_i \to Q$ *is constant over $t \in I_i$, that is, $q_i(t) = q_i(\tau_i)$ for all $t \in I_i$;*
  2. $x_i(\cdot) : I_i \to X$ *is the solution to the differential equation $\dot{x}_i = f(q_i(t), x_i(t))$ over $I_i$ starting at $x_i(\tau_i)$; and,*
  3. *for all $t \in [\tau_i, \tau_i')$, $x_i(t) \in \mathrm{Dom}(q_i(t))$.*

Definition 5 specifies which subset of the hybrid trajectories over $(Q, X)$ are executions of $H$ by imposing a number of restrictions. The first restriction dictates that the executions should start at an acceptable initial state in Init. The second restriction determines when discrete transitions may take place and what the state after discrete transitions may be. The requirement is that the state after a discrete transition is related to the state before a discrete transition through the reset relation $R$. In this context, it is convenient to think of $R$ as *enabling* discrete transitions: the execution *may* take a transition from a state $s$ as long as $R(s) \neq \emptyset$. The third restriction determines what happens along continuous evolution, and when continuous evolution must give way to a discrete transition. The first part implies that along continuous evolution the discrete state remains constant. The second part requires that along continuous evolution the continuous state flows according to the vector field $f$. The third part requires that along continuous evolution the state must remain in the domain,

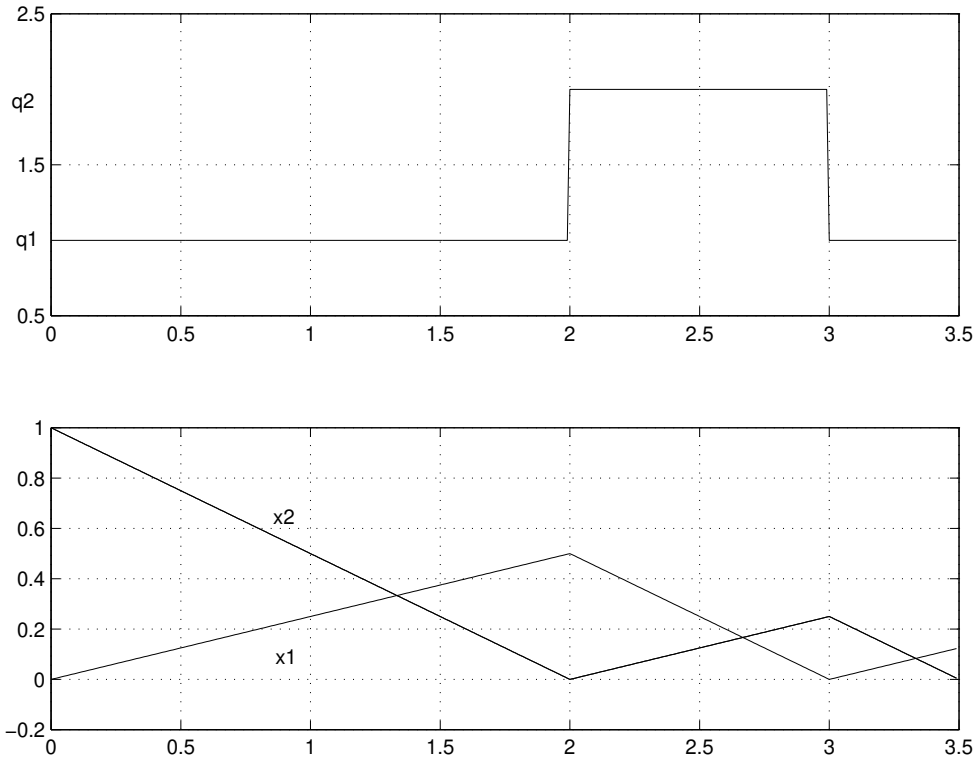


Figure 5: Example of an execution of the water tank hybrid automaton.

Dom. In this context, it is convenient to think of Dom as *forcing* discrete transitions: the execution *must* take a transition if the state is about to leave the domain. The definition is somewhat subtle here. Notice that the execution is required to remain in the domain throughput continuous evolution, *except* at the time instant just before a discrete transition. This assumption provides some additional generality, since one can define hybrid automata whose executions leave the domain for an instant. It will prove useful when studying the fundamental properties of hybrid automata in subsequent sections; for example, it will allow us to deal with hybrid automata whose domain is an open set.

**Example (Water Tank Automaton)** Figure 5 shows a finite execution of the water tank automaton. The hybrid time set $\tau$ of the execution in the figure consists of three intervals, $\tau = \{[0, 2], [2, 3], [3, 3.5]\}$. The evolution of the discrete state is shown in the upper plot, and the evolution of the continuous state is shown in the lower plot. The values chosen for the constants are $r_1 = r_2 = 0$, $v_1 = v_2 = 1/2$ and $w = 3/4$. The initial state is $(q_1, 0, 1)$. ∎

To simplify the notation, we will use $s_0 = s(\tau_0)$ to denote the initial state of an execution $(\tau, s)$. Since we restrict attention to autonomous hybrid automata, we can assume that $\tau_0 = 0$, without loss of generality.

Unlike continuous dynamical systems, the interpretation is that a hybrid automaton *accepts* (as opposed to generates) an execution. This difference in perspective allows one to consider, for example, hybrid automata that accept multiple executions for some initial states, a

property that can prove very useful when modeling uncertain systems, as the following example indicates.

**Example (Thermostat)** The thermostat system models the temperature variation in a room. The temperature is regulated by a thermostat, which switches on a heater when the temperature gets too low.

Let $x \in \mathbb{R}$ denote the temperature in the room. The heater is in one of two states, either on, or off. It is assumed that when the heater is on the temperature in the room rises towards 100 degrees according to the differential equation $\dot{x} = -x + 100$. When the heater is off on the other hand, the temperature decreases towards 0 degrees according to the differential equation $\dot{x} = -x$. The goal of the thermostat is to keep the temperature close to 75 degrees. To accomplice this it switches the heater on if the temperature reaches 70 degrees and off if the temperature reaches 80 degrees. There is, however some uncertainty in the switching process; the temperature may reach 68 degrees before the heating takes effect, and it may reach 82 degrees before the room begins to cool.

It is straight forward to define an autonomous hybrid automaton, $T$, to describe this process:

- $Q = \{\text{on}, \text{off}\}$;
- $X = \mathbb{R}$;
- $\text{Init} = X$;
- $f(\text{on}, x) = -x + 100$ and $f(\text{off}, x) = -x$;
- $\text{Dom}(\text{on}) = \{x \in \mathbf{X}_T \mid x \leq 82\}\text{Dom}(\text{off}) = \{x \in \mathbf{X}_T \mid x \geq 68\}$;
- $R(\text{on}, x) = (\text{off}, x)$ if $x \geq 80$, $R(\text{off}, x) = (\text{on}, x)$ if $x \leq 70$ and $R(q, x) = \emptyset$ otherwise.

It is easy to see that $T$ accepts a whole family of executions for each initial condition. Some examples are given in Figure 6. ∎

## 2.2 Graphical Representation

It is often convenient to represent hybrid automata as directed graphs, $(\mathbf{Q}, E)$. The vertices, $\mathbf{Q}$, of the graph correspond to values of the discrete state, while the edges, $E \subseteq \mathbf{Q} \times \mathbf{Q}$, represent possible transitions between the discrete states,

$$E = \{(q, q') \in \mathbf{Q} \times \mathbf{Q} \mid (q', x') \in R(q, x) \text{ for some } x, x' \in \mathbf{X}\}.$$

With each vertex $q \in \mathbf{Q}$, we associate a set of continuous initial states

$$\text{Init}_q = \{x \in \mathbf{X} \mid (q, x) \in \text{Init}\} \subseteq \mathbf{X},$$

a differential equation, $f_q : \mathbf{X} \to 2^{\mathbf{X}}$, given by

$$F_q(x) = F(q, x),$$

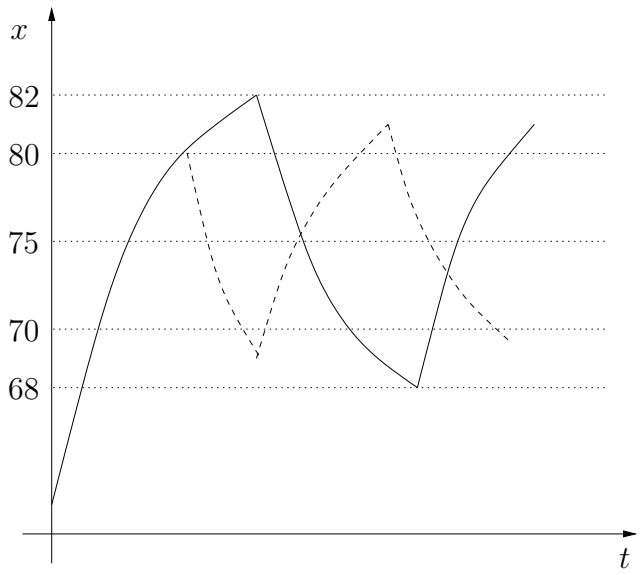


Figure 6: Examples of executions of the thermostat hybrid automaton.

and a domain,

$$\mathrm{Dom}_q = \{x \in \mathbf{X} \mid (q, x) \in \mathrm{Dom}\} \subseteq \mathbf{X}.$$

With each edge, $(q, q') \in E$, we associate a guard,

$$\mathrm{Guard}_{(q,q')} = \{x \in \mathbf{X} \mid (q', x') \in R(q, x) \text{ for some } x' \in \mathbf{X}\} \subseteq \mathbf{X},$$

and a reset relation, $\mathrm{Reset}_{(q,q')} : \mathbf{X} \to 2^{\mathbf{X}}$, given by

$$\mathrm{Reset}_{(q,q')}(x) = \{x' \in \mathbf{X} \mid (q', x') \in R(q, x)\}.$$

**Example (Water Tank and Thermostat (continued))** The thermostat and the water tank automata can be represented by the directed graphs of Figure 7a and Figure 7b respectively. In each vertex of the graph, we specify the value of the discrete state, a differential equation (implied by the vector field), and the domain. We write the continuous part of the initial state on an arrow pointing to the vertex corresponding to the discrete part of the initial state. The edges are represented by arrows pointing from the starting state to the destination state. The guard is written near the beginning of the arrow, and the reset near the end. Assignment is denoted by := (deterministic assignment) or :∈ (non-deterministic assignment). ∎

To simplify the figures, the reset will be suppressed for edges $(q, q') \in E$ with $\mathrm{Reset}_{(q,q')}(x) = \{x\}$ (as in the above examples). Likewise, the initial state arrow is suppressed for nodes $q \in \mathbf{Q}$ with $\mathrm{Init}_q = \emptyset$.

Notice that the graph associated with a hybrid automaton contains the same information as the definition of the automaton itself. The graphs, therefore, can also be treated as formal definitions of hybrid automata. We will take advantage of this fact in the examples

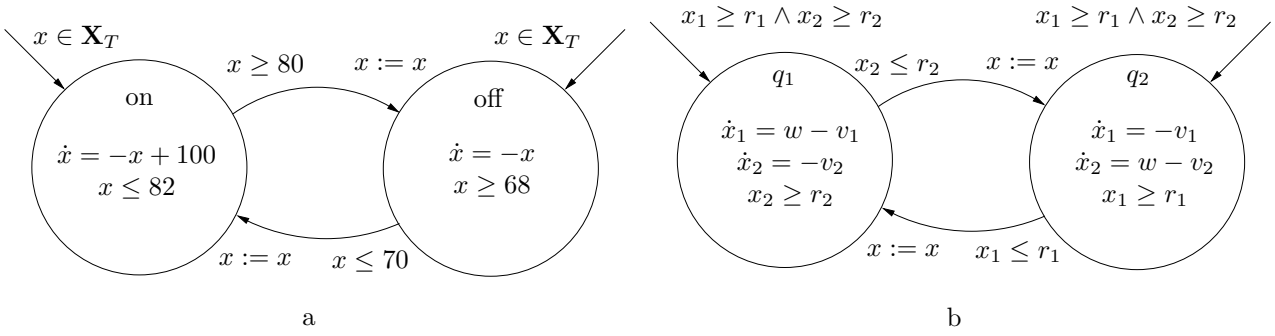


Figure 7: Graphical representation of the thermostat and water tank hybrid automata.

introduced in subsequent sections and use the more intuitive graphical representation to replace the formal (but occasionally cumbersome) definitions. In fact, graphs with guards, reset relations, etc. are commonly used in the hybrid systems literature as fundamental objects of modeling frameworks.

A subtle point here is that the correspondence of graphs to hybrid automata is not unique. The graphical representation may contain redundant elements. For example, one can define an edge $(q, q') \in E$ with $\text{Guard}_{(q,q')} = \emptyset$, or allow $\text{Reset}_{(q,q')}(x) = \emptyset$ for some $x \in \text{Guard}_{(q,q')}$. Situations like these are impossible in graphs generated from autonomous hybrid automata. Such graphs have the property that for all $(q, q') \in E$, $\text{Guard}_{(q,q')} \neq \emptyset$ and $\text{Reset}_{(q,q')}(x) \neq \emptyset$ for all $x \in \text{Guard}_{(q,q')}$. Graphs with these properties can be thought of as some form of "canonical" representations.

# References


[1] A. F. Filippov. *Differential equations with discontinuous right hand sides.* Kluwer Academic Publishers, Boston, 1988.

[2] V. I. Utkin. *Sliding Modes in Control Optimization.* Springer Verlag, Berlin, 1992.

[3] K. J. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry. On the regularization of zeno hybrid automata. *Systems and Control Letters*, 38:141–150, 1999.

[4] R. Alur and T. A. Henzinger. Modularity for timed and hybrid systems. In A. Mazurkiewicz and J. Winkowski, editors, *CONCUR 97: Concurrency Theory*, LNCS 1243, pages 74–88. Springer Verlag, 1997.

[5] M. S. Branicky. Multiple Lyapunov functions and other tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.

[6] Claire J. Tomlin. *Hybrid Control of Air Traffic Management Systems.* PhD thesis, Department of Electrical Engineering, University of California, Berkeley, 1998.