Hybrid Dynamical Systems

RAFAL GOEBEL,
RICARDO G. SANFELICE,
and ANDREW R. TEEL

ROBUST STABILITY AND CONTROL FOR SYSTEMS THAT COMBINE CONTINUOUS-TIME AND DISCRETE-TIME DYNAMICS

Many dynamical systems combine behaviors that are typical of continuous-time dynamical systems with behaviors that are typical of discrete-time dynamical systems. For example, in a switched electrical circuit, voltages and currents that change continuously according to classical electrical network laws also change discontinuously due to switches opening or closing. Some biological systems behave similarly, with continuous change during normal operation and discontinuous change due to an impulsive stimulus. Similarly, velocities in a multibody system change continuously according to Newton's second law but undergo instantaneous changes in velocity and momentum due to collisions. Embedded systems and, more generally, systems involving both digital and analog components form another class of examples. Finally, modern control algorithms often lead to both kinds of behavior, due to either digital components used in implementation or logic and decision making encoded in the control algorithm. These examples fit into the class of hybrid dynamical systems, or simply hybrid systems.

This article is a tutorial on modeling the dynamics of hybrid systems, on the elements of stability theory for hybrid systems, and on the basics of hybrid control. The presentation and selection of material is oriented toward the analysis of asymptotic stability in hybrid systems and the design of stabilizing hybrid controllers. Our emphasis on the robustness of asymptotic stability to data perturbation, external disturbances, and measurement error distinguishes the approach taken here from other

Digital Object Identifier 10.1109/MCS.2008.931718

1066-033X/09/$25.00©2009IEEE
Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

NESTOR DUB

approaches to hybrid systems. While we make some connections to alternative approaches, this article does not aspire to be a survey of the hybrid system literature, which is vast and multifaceted.

The interaction of continuous- and discrete-time dynamics in a hybrid system leads to rich dynamical behavior and phenomena not encountered in purely continuous-time systems. Consequently, several challenges are encountered on the path to a stability theory for hybrid systems and to a methodology for robust hybrid control design. The approach outlined in this article addresses these challenges, by using mathematical tools that go beyond classical analysis, and leads to a stability theory that unifies and extends the theories developed for continuous- and discrete-time systems. In particular, we give necessary and sufficient Lyapunov conditions for asymptotic stability in hybrid systems, show uniformity and robustness of asymptotic stability, generalize the invariance principle to the hybrid setting and combine it with Barabasin-Krasovskii techniques, and show the utility of such results for hybrid control design. Despite their necessarily more technical appearance, these results parallel what students of nonlinear systems are familiar with.

We now present some background leading up to the model of hybrid systems used in this article. A widely used model of a continuous-time dynamical system is the first-order differential equation $\dot{x} = f(x)$, with x belonging to an n-dimensional Euclidean space $\mathbb{R}^n$. This model can be expanded in two directions that are relevant for hybrid systems. First, we can consider differential equations with state constraints, that is, $\dot{x} = f(x)$ and $x \in C$, where C is a subset of $\mathbb{R}^n$. For example, the set C might indicate that the force of gravity cannot push a ball through the floor. Alternatively, the set C might indicate a set of physically meaningful initial conditions of the system. Second, we can consider the situation where the right-hand side of the differential equation is replaced by a set that may depend on x. For example, when the force applied to a particle varies with time in an unknown way in the interval $[a, b]$, we can model the derivative of the velocity as belonging to $[a, b]$. Another reason for considering set-valued right-hand sides is to account for the effect of perturbations, such as measurement error in a feedback control system, on a differential equation. Both situations lead to the differential inclusion $\dot{x} \in F(x)$, where F is a set-valued mapping.

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

The examples of hybrid control systems provided in this article only scratch the surface of what is possible using hybrid feedback control.

Combining the two generalizations leads to constrained differential inclusions $\dot{x} \in F(x)$, $x \in C$.

A typical model of a discrete-time dynamical system is the first-order equation $x^+ = g(x)$, with $x \in \mathbb{R}^n$. The notation x^+ indicates that the next value of the state is given as a function of the current state x through the value $g(x)$. As for differential equations, it is a natural extension to consider constrained difference equations and difference inclusions, which leads to the model $x^+ \in G(x)$, $x \in D$, where G is a set-valued mapping and D is a subset of $\mathbb{R}^n$.

Since a model of a hybrid dynamical system requires a description of the continuous-time dynamics, the discrete-time dynamics, and the regions on which these dynamics apply, we include both a constrained differential inclusion and a constrained difference inclusion in a general model of a hybrid system in the form

$$\dot{x} \in F(x), \quad x \in C, \tag{1}$$
$$x^+ \in G(x), \quad x \in D. \tag{2}$$

The model (1), (2) captures a wide variety of dynamic phenomena including systems with logic-based state components, which take values in a discrete set, as well as timers, counters, and other components. Examples in this article demonstrate how to cast hybrid automata and switched systems, as well as sampled-data and networked control systems, into the form (1), (2). We refer to a hybrid system in the form (1), (2) as $\mathcal{H}$. We call C the flow set, F the flow map, D the jump set, and G the jump map.

For many systems, the generality provided by the inclusions in (1), (2) is not needed. Thus, the reader may replace the set-valued mappings and the corresponding inclusions in (1), (2) with equations and proceed confidently. It is often the geometry of sets C and D that produces the rich dynamical phenomena in a hybrid system rather than the multivaluedness of the mappings F and G. However, this article does justify, beyond the sake of generality, the use of differential inclusions and difference inclusions.

We provide examples of hybrid models in the following section. Subsequently, we make precise the meaning of a solution to a hybrid dynamical system and describe basic mathematical properties of the space of solutions. Afterward, we present results on asymptotic stability in hybrid systems, with an emphasis on robustness. Initially, we focus on Lyapunov functions as the primary stability analysis tool and show how Lyapunov functions are used in hybrid control design. Finally, we present tools for stability analysis based on limited events in hybrid systems and show how these tools are related to hybrid feedback control algorithms.

The main developments of the article are complemented by several supporting discussions. "Hybrid Automata" and "Switching Systems" relate systems in the form (1), (2) to hybrid automata and switching systems, respectively. "Related Mathematical Frameworks" presents other mathematical descriptions of systems where features of both continuous- and discrete-time dynamical systems are present. "Existence, Uniqueness, and other Well-Posedness Issues" discusses basic properties of solutions to hybrid systems in the form (1), (2). "Set Convergence" and "Robustness and Generalized Solutions" introduce mathematical tools from beyond classical analysis and motivate the assumptions placed on the data of a hybrid system $\mathcal{H}$, as given by (C, F, D, G). "Motivating Stability of Sets" and "Why 'Pre'-Asymptotic Stability?" explain distinct features of the asymptotic stability concept used in the article. "Converse Lyapunov Theorems" and "Invariance" state and discuss main tools used in the stability analysis. "Zeno Solutions" describes a phenomenon unique to hybrid dynamical systems. "Simulation in Matlab/Simulink" presents an approach to simulation of hybrid systems. The notation used throughout this article is defined in "List of Symbols."

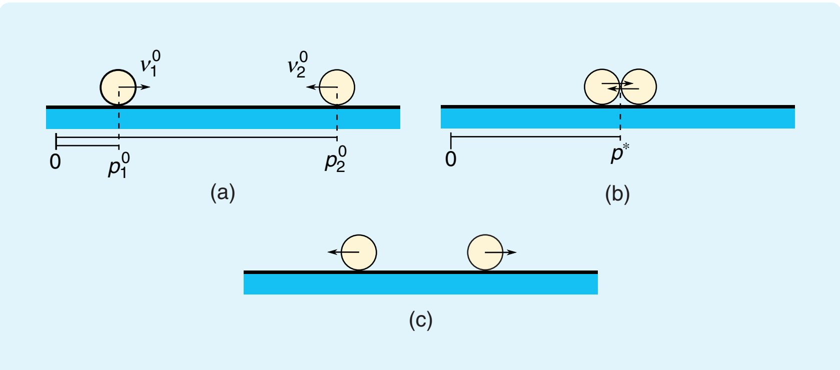

FIGURE 1 Collision between two particles. (a) Two particles are initialized to positions p_1^0 and p_2^0 and with velocities v_1^0 and v_2^0. (b) An impact between the particles occurs at the position p^*. (c) The direction of the motion of each particle is reversed after the impact.

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

HYBRID PHENOMENA AND MODELING

Colliding Masses

Many engineering systems experience impacts [8], [56]. Walking and jumping robots, juggling systems, billiards, and a bouncing ball are examples. Continuous-time equations of motion describe the behavior of these systems between impacts, whereas discrete dynamics approximate what happens during impacts.

Consider two particles that move toward each other, collide, and then move away from each other, as shown in Figure 1. Before and after the collision, the position and velocity of each particle are governed by Newton's second law. At impact, the velocity evolution is modeled by an instantaneous change in the velocities but no change in the positions of the particles.

The combined continuous and discrete behavior of the particles can be modeled as a hybrid system with a differential equation governing the continuous dynamics, constraints describing where the continuous dynamics apply, a difference equation governing the discrete dynamics, and constraints describing where the discrete dynamics apply. The state is the vector $x = (p, v) = (p_1, p_2, v_1, v_2)$, where p_1 and p_2 denote the particles' positions and v_1 and v_2 denote the particles' velocities. The state vector changes continuously if, as shown in Figure 1, the first particle's position is at or to the left of the second particle's position. This condition is described by the flow set $C := \{(p, v) : p_1 \le p_2\}$. Assuming no friction, the flow map obtained applying Newton's second law and using (1) is $F(p, v) = (v_1, 0, v_2, 0)$.

An impact occurs when the positions of the particles are identical and their velocities satisfy $v_1 \ge v_2$. These conditions define the jump set $D := \{(p, v) : p_1 = p_2,\ v_1 \ge v_2\}$. Letting v_1^+ and v_2^+ indicate the velocities after an impact, we have the conservation of momentum equation

$$m_1 v_1^+ + m_2 v_2^+ = m_1 v_1 + m_2 v_2 \tag{3}$$

and the energy dissipation equation

$$v_1^+ - v_2^+ = -\rho(v_1 - v_2), \tag{4}$$

where m_1 and m_2 are the masses of the particles and the constant $\rho \in (0, 1)$ is a restitution coefficient. Solving (3) and (4) for v_1^+ and v_2^+ and using (2) yields the jump map

$$G(p_1, p_2, v_1, v_2) = (p_1, p_2, v_1 - m_2\lambda(v_1 - v_2), v_2 + m_1\lambda(v_1 - v_2)),$$

where $\lambda = (1 + \rho)/(m_1 + m_2)$.

Impulsive Behavior in Biological Systems

Synchronization in groups of biological oscillators occurs in swarms of fireflies [10], groups of crickets [88], ensembles of neuronal oscillators [32], and groups of heart muscle cells [61]. Detailed treatments include [62] and [78]. The discussion below is related to [10] and [55], where models of a collection of nonlinear clocks with impulsive coupling are studied. A model of two linear clocks with impulsive coupling is used in [61] to analyze the synchronization of heart muscle cells.

List of Symbols

$\dot{x}$	The derivative, with respect to time, of the state of a hybrid system
x^+	The state of a hybrid system after a jump
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^n$	The n-dimensional Euclidean space
$\mathbb{R}_{\ge 0}$	The set of nonnegative real numbers, $\mathbb{R}_{\ge 0} = [0, \infty)$
$\mathbb{Z}$	The set of all integers
$\mathbb{N}$	The set of nonnegative integers, $\mathbb{N} = \{0, 1, \ldots\}$
$\mathbb{N}_{\ge k}$	$\{k, k+1, \ldots\}$ for a given $k \in \mathbb{N}$
$\emptyset$	The empty set
$\overline{\Sigma}$	The closure of the set Σ
$\mathrm{con}\, \Sigma$	The convex hull of the set Σ
$\overline{\mathrm{con}}\, \Sigma$	The closure of the convex hull of a set Σ
$\Sigma_1 \setminus \Sigma_2$	The set of points in Σ_1 that are not in Σ_2
$\Sigma_1 \times \Sigma_2$	The set of ordered pairs (σ_1, σ_2) with $\sigma_1 \in \Sigma_1,\ \sigma_2 \in \Sigma_2$
$x^\top$	The transpose of the vector x
(x, y)	Equivalent notation for the vector $[x^\top y^\top]^\top$
$\lvert x \rvert$	The Euclidean norm of a vector $x \in \mathbb{R}^n$
$\mathbb{B}$	The closed unit ball, of appropriate dimension, in the Euclidean norm
$\lvert x \rvert_\Sigma$	$\inf_{y \in \Sigma} \lvert x - y \rvert$ for a set $\Sigma \subset \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$
$\mathbb{S}^n$	The set $\{x \in \mathbb{R}^{n+1} : \lvert x \rvert = 1\}$
$f : \mathbb{R}^m \to \mathbb{R}^n$	A function from $\mathbb{R}^m$ to $\mathbb{R}^n$
$F : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$	A set-valued mapping from $\mathbb{R}^m$ to $\mathbb{R}^n$
$R(\cdot)$	The rotation matrix $R(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$
$F(\Sigma)$	$\bigcup_{x \in \Sigma} F(x)$ for the set-valued mapping $F : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ and a set $\Sigma \subset \mathbb{R}^m$
$T_\Sigma(\eta)$	The *tangent cone* to the set $\Sigma \subset \mathbb{R}^n$ at $\eta \in \overline{\Sigma}$. $T_\Sigma(\eta)$ is the set of all vectors $w \in \mathbb{R}^n$ for which there exist $\eta_i \in \Sigma$, $\tau_i > 0$, for all $i = 1, 2, \ldots$ such that $\eta_i \to \eta$, $\tau_i \searrow 0$, and $(\eta_i - \eta)/\tau_i \to w$ as $i \to \infty$
$\mathcal{K}_\infty$	The class of functions from $\mathbb{R}_{\ge 0}$ to $\mathbb{R}_{\ge 0}$ that are continuous, zero at zero, strictly increasing, and unbounded
$L_V(\mu)$	The μ-level set of the function $V : \mathrm{dom}\, V \to \mathbb{R}$, which is the set of points $\{x \in \mathrm{dom}\, V : V(x) = \mu\}$

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

Hybrid Automata

Some models of hybrid systems explicitly partition the state of a system into a continuous state ξ and a discrete state q, the latter describing the mode of the system. For example, the values of q may represent modes such as "working" and "idle." In a temperature control system, q may stand for "on" or "off," while ξ may represent the temperature. By its nature, the discrete state can change only during a jump, while the continuous state often changes only during flows but sometimes may jump as well. These systems are called differential automata [82], hybrid automata [S2], [51], or simply hybrid systems [S1], [9]. All of these systems can be cast as a hybrid system of the form (1), (2).

The data of a hybrid automaton are usually given by

- a *set of modes* Q, which in most situations can be identified with a subset of the integers
- a *domain map* Domain: $Q \rightrightarrows \mathbb{R}^n$, which gives, for each $q \in Q$, the set Domain(q) in which the continuous state ξ evolves
- a *flow map* $f: Q \times \mathbb{R}^n \to \mathbb{R}^n$, which describes, through a differential equation, the continuous evolution of the continuous state variable ξ
- a *set of edges* Edges $\subset Q \times Q$, which identifies the pairs (q, q') such that a transition from the mode q to the mode q' is possible
- a *guard map* Guard : Edges $\rightrightarrows \mathbb{R}^n$, which identifies, for each edge $(q, q') \in$ Edges, the set Guard(q, q') to which the continuous state ξ must belong so that a transition from q to q' can occur
- a *reset map* Reset : Edges $\times \mathbb{R}^n \to \mathbb{R}^n$, which describes, for each edge $(q, q') \in$ Edges, the value to which the continuous state $\xi \in \mathbb{R}^n$ is set during a transition from mode q to mode q'. When the continuous state variable ξ remains constant at a jump from q to q', the map Reset$(q, q', \cdot)$ can be taken to be the identity.

Figure S1 depicts part of a state diagram for a hybrid automaton. The continuous dynamics of two modes are shown, together with the guard conditions and reset rules that govern transitions between these modes.

We now show how a hybrid automaton can be modeled as a hybrid system in the form (1), (2). First, we reformulate a hybrid automaton as a hybrid system with explicitly shown modes. For each $q \in Q$, we take

$$C_q = \text{Domain}(q), \quad D_q = \bigcup_{(q,q') \in \text{Edges}} \text{Guard}(q, q'),$$

$$F_q(\xi) = f(q, \xi), \quad \text{for all } \xi \in C_q,$$

$$G_q(\xi) = \bigcup_{\{q' : \xi \in \text{Guard}(q,q')\}} (\text{Reset}(q, q', \xi), q'), \quad \text{for all } \xi \in D_q.$$

When ξ is an element of two different guard sets Guard(q, q') and Guard(q, q''), $G_q(\xi)$ is a set consisting of at least two points. Hence, G_q can be set valued. In fact, G_q is not necessarily a function even when every Reset$(q, q', \cdot)$ is the identity map. With C_q, F_q, D_q, and G_q defined above, we consider the hybrid system with state $(\xi, q) \in \mathbb{R}^n \times \mathbb{R}$ and representation

$$\dot{\xi} = F_q(\xi), \qquad q \in Q, \ \xi \in C_q,$$

$$(\xi^+, q^+) \in G_q(\xi), \qquad q \in Q, \ \xi \in D_q.$$

Example S1: Reformulation of a Hybrid Automaton

Consider the hybrid automaton shown in figures S2 and S3, with the set of modes $Q = \{1, 2\}$; the domain map given by

$$\text{Domain}(1) = \mathbb{R}_{\le 0} \times \mathbb{R}, \quad \text{Domain}(2) = \{0\} \times \mathbb{R};$$

the flow map, for all $\xi \in \mathbb{R}^2$, given by

$$f(1, \xi) = (1, 1), \quad f(2, \xi) = (0, -1);$$

the set of edges given by Edges $= \{(1, 1), (1, 2), (2, 1)\}$; the guard map given by

$$\text{Guard}(1, 1) = \mathbb{R}_{\ge 0} \times \mathbb{R}_{\le 0},$$
$$\text{Guard}(1, 2) = \mathbb{R}^2_{\ge 0},$$
$$\text{Guard}(2, 1) = \{0\} \times \mathbb{R}_{\le 0};$$

and the reset map, for all $\xi \in \mathbb{R}^2$, given by

$$\text{Reset}(1, 1, \xi) = (-5, 0),$$
$$\text{Reset}(1, 2, \xi) = \xi,$$
$$\text{Reset}(2, 1, \xi) = 2\xi.$$

The sets Guard$(1, 1)$ and Guard$(1, 2)$ overlap, indicating that, in mode 1, a reset of the state ξ to $(-5, 0)$ or a switch of the mode to 2 is possible from points

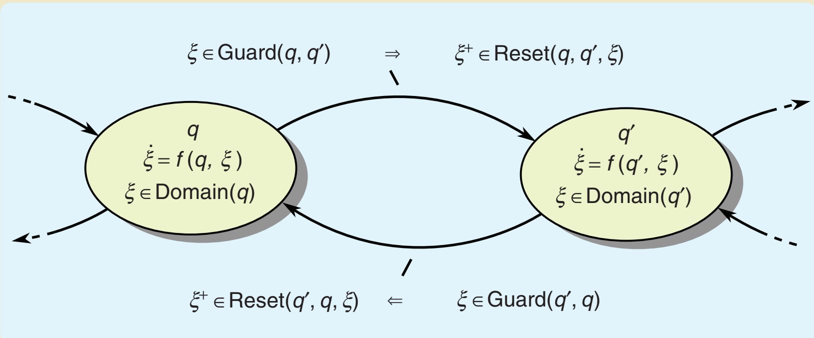

FIGURE S1 Two modes, q and q', of a hybrid automaton. In mode q, the state ξ evolves according to the differential equation $\dot{\xi} = f(q, \xi)$ in the set Domain(q). A transition from mode q to mode q' can occur when, in mode q, ξ is in the set Guard(q, q'). During the transition, ξ changes to a value ξ^+ in Reset(q, q', ξ). Transitions from mode q to other modes, not shown in the figure, are governed by similar rules.

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

where $\xi_1 \geq 0$ and $\xi_2 = 0$. Formulating this hybrid automaton as a hybrid system with explicitly shown modes leads to $D_1 = \mathrm{Guard}(1, 1) \cup \mathrm{Guard}(1, 2) = \mathbb{R}_{\geq 0} \times \mathbb{R}$ and the set-valued jump map G_1 given by

$$G_1(\xi) = \begin{cases} (-5, 0, 1), & \text{if } \xi_1 \geq 0, \xi_2 < 0, \\ (-5, 0, 1) \cup (\xi_1, \xi_2, 2), & \text{if } \xi_1 \geq 0, \xi_2 = 0, \\ (\xi_1, \xi_2, 2), & \text{if } \xi_1 \geq 0, \xi_2 > 0. \end{cases}$$

■

To formulate a hybrid automaton in the form (1), (2), we define $x = (\xi, q) \in \mathbb{R}^{n+1}$ and

$$C = \bigcup_{q \in Q} (C_q \times \{q\}), \quad F(x) = F_q(\xi) \times \{0\} \quad \text{for all } x \in C,$$

$$D = \bigcup_{q \in Q} (D_q \times \{q\}), \quad G(x) = G_q(\xi) \quad \text{for all } x \in D.$$

When the domains and guards are closed sets, the flow and jump sets C and D are also closed. Similarly, when the flow and reset maps are continuous, the flow map F and the jump map G satisfy the Basic Assumptions.

Example S1 Revisited: Solutions to a Hybrid Automaton

Consider the hybrid system modeling the hybrid automaton of Example S1. The initial condition $\xi = (3, 3)$, in mode $q = 1$, of the hybrid automaton corresponds to the initial condition (3, 3, 1) for the hybrid system. The maximal solution to the hybrid system starting from (3, 3, 1), denoted x_a, has domain $\mathrm{dom}\, x_a = (0, 0) \cup (0, 1)$ and is given by $x_a(0, 0) = (3, 3, 1)$, $x_a(0, 1) = (3, 3, 2)$. The solution jumps once, the jump takes x_a outside of both the jump set and the flow set, and thus cannot be extended.

The hybrid system has multiple solutions from the initial point $(0, 0, 1)$. One maximal solution starting from $(0, 0, 1)$, denoted x_b, is complete and never flows. This solution has $\mathrm{dom}\, x_b = \{0\} \times \mathbb{N}$ and is given by $x_b(0, j) = 1.5 - .5(-1)^j$. That is, the solution x_b switches back and forth between mode 1 and mode 2 infinitely many times. Another solution starting from $(0, 0, 1)$, denoted x_c, has $\mathrm{dom}\, x_c = (0, 0) \cup ([0, 5], 1) \cup ([5, 10], 2) \cup (10, 3)$ and is given by

$$x_c(t, j) = \begin{cases} (0, 0, 1), & \text{if } t = 0, j = 0, \\ (-5 + t, t, 1), & \text{if } t \in [0, 5], j = 1, \\ (0, 5 - (t - 5), 2), & \text{if } t \in [5, 10], j = 2, \\ (0, 0, 1), & \text{if } t = 10, j = 3. \end{cases}$$

In the language of hybrid automata, this solution undergoes a reset of the state without a switch of the mode, flows for five units of time until it hits a guard, switches the mode without resetting the state, flows for another five units of time until it hits another guard, and switches the mode without resetting the state. This solution is not maximal since it can be extended in several ways. One way is by concatenating x_c with x_b, that is, by setting $x_c(t + 10, j + 3) = x_b(t, j)$ for $(t, j) \in \mathrm{dom}\, x_b$. In other words, x_c can be extended by back and forth switches between modes. The solution x_c can be also extended to be periodic. We can consider $x_c(t, j) = (-5 + (t - 10), t - 10, 1)$ for $t \in [10, 15]$, $j = 4$, $x_c(t, j) = (0, 5 - (t - 15), 2)$ for $t \in [15, 20]$, $s = 5$, $x_c(20, 6) = (0, 0, 1)$, and repeat.

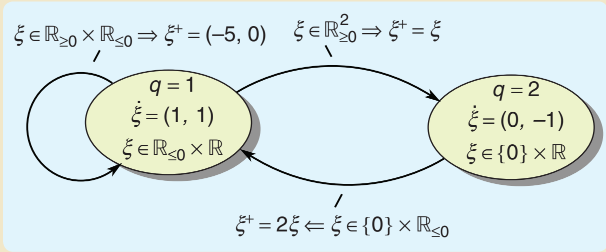

FIGURE S2 Two modes of the hybrid automaton in Example S1. In mode 1, the state ξ evolves according to the differential equation $\dot{\xi} = (1, 1)$ in the set $\mathrm{Domain}(1) = \mathbb{R}_{\leq 0} \times \mathbb{R}$. A transition from mode 1 to mode 2 occurs when ξ is in the guard set $\mathrm{Guard}(1, 2) = \mathbb{R}^2_{\geq 0}$ but not in the guard set $\mathrm{Guard}(1, 1) = \mathbb{R}_{\geq 0} \times \mathbb{R}_{\leq 0}$. During the transition, ξ does not change its value. Also in mode 1, a jump in ξ to the value $(-5, 0)$ occurs when ξ is in the guard set $\mathrm{Guard}(1, 1) = \mathbb{R}_{\geq 0} \times \mathbb{R}_{\leq 0}$ but not in the guard set $\mathrm{Guard}(1, 2) = \mathbb{R}^2_{\geq 0}$. When ξ belongs to $\mathrm{Guard}(1, 2)$ and $\mathrm{Guard}(1, 1)$, either the transition to mode 2 or the jump of ξ can occur. In mode 2, the state ξ evolves according to the differential equation $\dot{\xi} = (0, -1)$ in the set $\mathrm{Domain}(2) = \{0\} \times \mathbb{R}$. A transition from mode 2 to mode 1 can occur when ξ is in the guard set $\mathrm{Guard}(2, 1) = \{0\} \times \mathbb{R}_{\leq 0}$. During the transition, ξ changes to the value 2ξ.

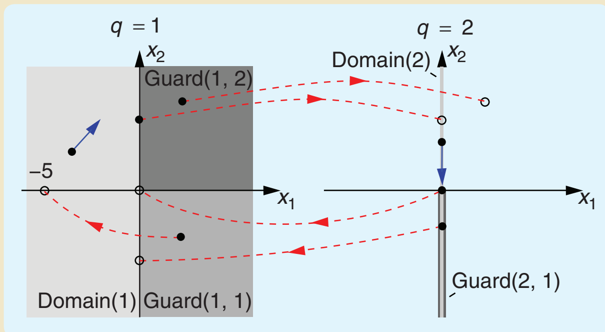

FIGURE S3 Data for the hybrid automaton in Example S1. Solid arrows indicate the direction of flow in Domain(1) and Domain(2). Dashed arrows indicate jumps from Guard(1, 1) and Guard(2, 1).

REFERENCES

[S1] M. S. Branicky, V. S. Borkar, and S. K. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Trans. Automat. Contr.*, vol. 43, no. 1, pp. 31–45, 1998.

[S2] T. A. Henzinger, "The theory of hybrid automata," in *Proc. of the 11th Annu. Symp. Logic in Computer Science*, IEEE CS Press, 1996, pp. 278–292.

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

Switching Systems

A switching system is a differential equation whose right-hand side is chosen from a family of functions based on a switching signal [49], [S4]. For each switching signal, the switching system is a time-varying differential equation. As in [S3], we study the properties of a switching system not under a particular switching signal but rather under various classes of switching signals.

In the framework of hybrid systems, information about the class of switching signals often can be embedded into the system data by using timers and reset rules, which can be viewed as an autonomous, that is, time-invariant hybrid subsystem. Results for switching systems, including converse Lyapunov theorems and invariance principles [25], can be then deduced from results obtained for hybrid systems.

A switched system can be written as

$$\dot{\xi} = f_q(\xi), \tag{S1}$$

where, for each $q \in Q = \{1, 2, \ldots, q_{\max}\}$, $f_q : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous function. A *complete solution* to the system (S1) consists of a locally absolutely continuous function $\xi : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and a function $q : \mathbb{R}_{\geq 0} \to Q$ that is piecewise constant, has a finite number of discontinuities in each compact time interval, and satisfies $\dot{\xi}(t) = f_{q(t)}(\xi(t))$ for almost all $t \in \mathbb{R}_{\geq 0}$. In what follows, given a complete solution (ξ, q) to (S1), let l be the number of discontinuities of q, with the possibility of $l = \infty$, and let $t_0 = 0$ and $\{t_i\}_{i=1}^{l}$ be the increasing sequence of times at which q is discontinuous. For simplicity, we discuss complete solutions only.

A solution (ξ, q) to (S1) is a *dwell-time solution with dwell time* $\tau_D > 0$ if $t_{i+1} - t_i \geq \tau_D$ for all $i = 1, 2, \ldots, l-1$. That is, switches are separated by at least an amount of time τ_D. Each dwell-time solution can be generated as part of a solution to the hybrid system with state $x = (\xi, q, \tau) \in \mathbb{R}^{n+2}$ given by

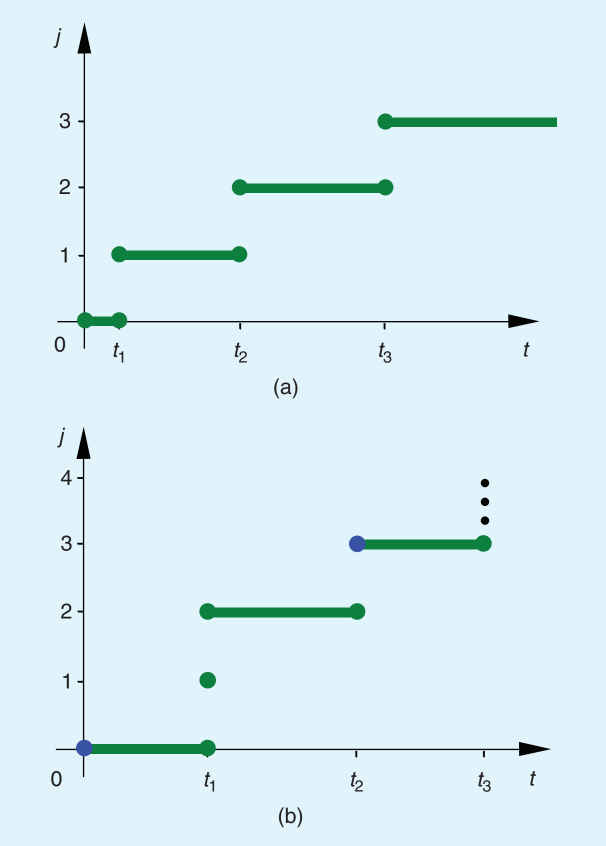

FIGURE S4 Hybrid time domain for a solution under dwell-time switching and average dwell-time switching. (a) Hybrid time domain for a dwell-time solution with dwell-time constant τ_D larger or equal than $\min\{t_2 - t_1, t_3 - t_2, \ldots\}$. (b) Hybrid time domain for an average dwell-time solution for parameters (δ, N) satisfying the average dwell-time condition in (S2). For example, parameters $(\delta, N) = (4/\min\{t_1, t_2 - t_1\}, 2)$ and $(\delta, N) = (4/t_2, 4)$ satisfy (S2). The domain repeats periodically, as denoted by the blue dot.

Consider a group of fireflies, each of which has an internal clock state. Suppose each firefly's clock state increases monotonically until it reaches a positive threshold, assumed to be the same for each firefly. When a firefly's clock reaches its threshold, the clock resets to zero and the firefly flashes, which causes the other fireflies' clocks to jump closer to their thresholds. In this way, the flash of one firefly affects the internal clocks of the other fireflies.

Figure 2 depicts the evolution of the internal clocks of two fireflies coupled through flashes. The time units are normalized so that each firefly's internal clock state takes values in the interval $[0, 1]$, and thus the threshold for flashing is one for each firefly.

A hybrid model for a system of n fireflies, each with the same clock characteristics, has the state $x = (x_1, \ldots, x_n)$, flow map $F(x) = (f(x_1), \ldots, f(x_n))$, where $f : [0, 1] \to \mathbb{R}_{>0}$ is continuous, and flow set $C = [0, 1]^n$, where $[0, 1]^n$ indicates the set of points x in $\mathbb{R}^n$ for which each component x_i belongs to the interval $[0, 1]$. The function f governs the rate at which each clock state evolves in the interval $[0, 1]$.

Since jumps in the state of the system are allowed when any one of the fireflies' clocks reaches its threshold, the jump set is $D = \{x \in C : \max_i x_i = 1\}$. One way to model the impulsive changes in the clock states is through a rule that instantaneously advances a clock state by a factor $1 + \varepsilon$, where $\varepsilon > 0$, when this action does not take the clock state past its threshold. Otherwise, the clock state is reset to zero, just as if it had reached its threshold. The corresponding jump map G does not satisfy the regularity condition (A3) of the Basic Assumptions imposed in the section "Basic Mathematical Properties." The algorithm for defining generalized solutions in "Robustness and

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

$$\left.\begin{array}{rcl}\dot{\xi} &=& f_q(\xi)\\ \dot{q} &=& 0\\ \dot{\tau} &\in& [0, 1/\tau_D]\end{array}\right\} =: F(x),\ x \in C := \mathbb{R}^n \times Q \times [0,1],$$

$$\left.\begin{array}{rcl}\xi^+ &=& \xi\\ q^+ &\in& Q\\ \tau^+ &=& 0\end{array}\right\} =: G(x), x \in D := \mathbb{R}^n \times Q \times \{1\}.$$

Note that it takes at least an amount of time τ_D for the timer state τ to increase from zero to one with velocity $\dot{\tau} \in [0, 1/\tau_D]$. Therefore, τ ensures that jumps of this hybrid system occur with at least τ_D amount of time in between them. In fact, there is a one-to-one correspondence between dwell-time solutions to (S1) and solutions to the hybrid system initialized with $\tau = 1$ for which $\dot{\tau} = 1$ when $\tau \in [0, 1)$ and $\dot{\tau} = 0$ when $\tau = 1$.

A solution (ξ, q) to (S1) is an *average dwell-time solution* if the number of switches in a compact interval is bounded by a number that is proportional to the length of the interval, with proportionality constant $\delta \geq 0$, plus a positive constant N [34]. In the framework of hybrid systems, each average dwell-time solution has a hybrid time domain E such that, for each pair (s, i) and (t, i) belonging to E and satisfying with $s \leq t$ and $i \leq j$,

$$j - i \leq (t - s)\delta + N. \tag{S2}$$

Dwell-time solutions are a special case, corresponding to $\delta = 1/\tau_D$ and $N = 1$. Every hybrid time domain that satisfies (S2) can be generated by the hybrid subsystem with compact flow and jump sets given by

$$\dot{\tau} \in [0, \delta], \quad \tau \in [0, N], \tag{S3}$$
$$\tau^+ = \tau - 1, \quad \tau \in [1, N]. \tag{S4}$$

The time domain for each solution of this hybrid system satisfies the constraint (S2). Furthermore, for every hybrid time domain E satisfying (S2) there exists a solution of (S3), (S4), starting at $\tau = N$, and defined on E [S5], [14]. In turn, switching systems under an average dwell-time constraint with parameters (δ, N) are captured by the hybrid system with state $x = (\xi, q, \tau) \in \mathbb{R}^{n+2}$ given by

$$\left.\begin{array}{rcl}\dot{\xi} &=& f_q(\xi)\\ \dot{q} &=& 0\\ \dot{\tau} &\in& [0, \delta]\end{array}\right\} =: F(x), \quad x \in C := \mathbb{R}^n \times Q \times [0, N],$$

$$\left.\begin{array}{rcl}\xi^+ &=& \xi\\ q^+ &\in& Q\\ \tau^+ &=& \tau - 1\end{array}\right\} =: G(x), \quad x \in D := \mathbb{R}^n \times Q \times [1, N].$$

Figure S4 depicts a hybrid time domain for a dwell-time solution and an average dwell-time solution to a switching system.

More elaborate families of solutions to switching systems can be modeled by means of hybrid systems. We briefly mention one such family. A solution (x, q) to (S1) is a *persistent dwell-time solution with persistent dwell time* $\tau_D > 0$ *and period of persistence* $T > 0$ if there are infinitely many intervals of length at least τ_D on which no switches occur, and such intervals are separated by at most an amount of time T [S3]. A hybrid system that models such solutions involves two timers. One timer ensures that periods with no switching last at least an amount of time τ_D; the other timer ensures that periods of arbitrary switching do not last more than an amount of time T. The hybrid system also involves a differential inclusion $\dot{\xi} \in \Phi(\xi)$, where $\Phi(\xi) := \overline{\text{con}}\bigcup_{q \in Q} f_q(\xi)$, to describe solutions to the switching system during periods of arbitrary switching.

REFERENCES

[S3] J. P. Hespanha, "Uniform stability of switched linear systems: Extensions of LaSalle's invariance principle," *IEEE Trans. Automat. Contr.*, vol. 49, pp. 470–482, 2004.

[S4] D. Liberzon and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Syst. Mag.*, vol. 19, pp. 59–70, 1999.

[S5] S. Mitra and D. Liberzon, "Stability of hybrid automata with average dwell time: An invariant approach," in *Proc. 43rd IEEE Conf. Decision and Control*, Bahamas, Dec. 2004, pp. 1394–1399.

Generalized Solutions" motivates the modified jump map $G(x) = (g(x_1), \ldots, g(x_n))$, where

$$g(x_i) = \begin{cases} (1+\varepsilon)x_i, & \text{when } (1+\varepsilon)x_i < 1,\\ 0, & \text{when } (1+\varepsilon)x_i > 1,\\ \{0, 1\}, & \text{when } (1+\varepsilon)x_i = 1,\end{cases}$$

which does satisfies the regularity condition (A3). This jump map advances the clock state x_i by the factor $1 + \varepsilon$ when this action keeps x_i below the threshold, and it resets x_i to zero when multiplying x_i by $1 + \varepsilon$ produces a value greater than the threshold value. Either resetting the clock state to zero or advancing the clock state by $1 + \varepsilon$ can occur when $(1 + \varepsilon)x_i = 1$.

A group of fireflies can exhibit almost global synchronization, meaning that, from almost every initial condition, the state vector tends to the set where all of the clock states are equal [55]. Synchronization analysis for the case $n = 2$ is given in Example 25.

Explicit Zero-Crossing Detection

Zero-crossing detection (ZCD) algorithms for sinusoidal signals are crucial for estimating phase and frequency as well as power factor in electric circuits. ZCD algorithms employ a discrete state, which remembers the most recent sign of the signal and is updated when the signal crosses zero, as indicated in Figure 3.

We cast a simple ZCD algorithm for a sinusoidal signal in terms of a hybrid system. Let the sinusoid be generated as the output of the linear system $\dot{\xi}_1 = \omega\xi_2$, $\dot{\xi}_2 = -\omega\xi_1$, $y = \xi_1$, where $\omega > 0$, and let q denote a discrete state taking values in $Q := \{-1, 1\}$ corresponding to the sign of ξ_1. The state of the hybrid system is $x = (\xi, q)$, while the flow map is $F(x) = (\omega\xi_2, -\omega\xi_1, 0)$.

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

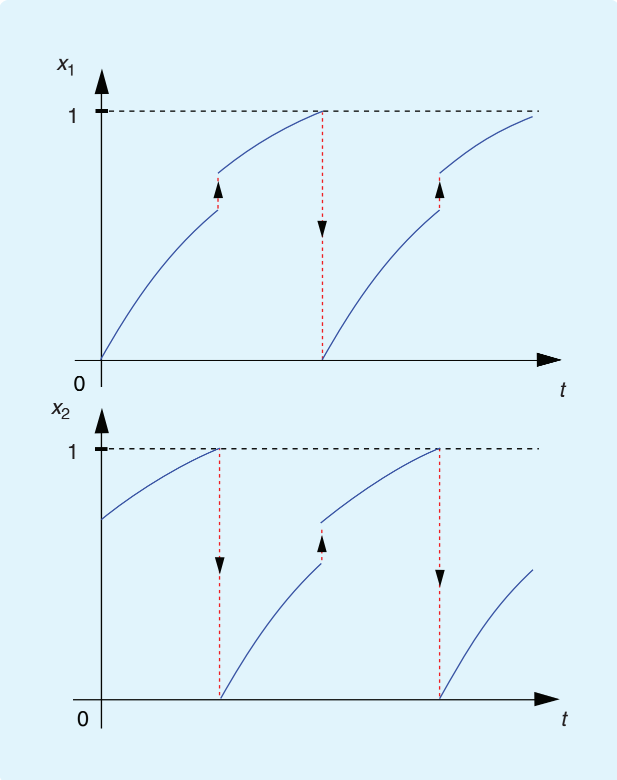

FIGURE 2 Trajectories of the internal clocks of two fireflies with impulsive coupling. When either clock state x_1 or x_2 reaches the unit threshold, both states experience a jump. When a state reaches the threshold, it is reset to zero. At the same time, the other state is increased by a factor $1 + \varepsilon$ if this increase does not push the state past the threshold; otherwise, this state is also reset to zero.

As ξ_1 changes sign, a zero-crossing event occurs. We model the detection of this event as a toggling of the state q through the rule $q^+ = -q$. In a more elaborate model, either a counter that keeps track of the number of zero-crossing events can be incremented, or a timer state that monitors the amount of time between zero-crossing events can be reset. The state ξ does not change during jumps. The jump map is thus $G(x) = (\xi, -q)$.

When q and ξ_1 have the same sign, that is, $q\xi_1 \geq 0$, flows are allowed. This behavior corresponds to the flow set $C = \bigcup_{q\in Q}(C_q \times \{q\})$, where $C_q := \{\xi \in \mathbb{R}^2 : q\xi_1 \geq 0\}$. In other words, the flow set C is the union of two sets. One set corresponds to points where $q = 1$ and $\xi_1 \geq 0$, while the other set corresponds to points where $q = -1$ and $\xi_1 \leq 0$.

When $\xi_1 = 0$ and the sign of q is opposite to the sign of the derivative of ξ_1, that is, $q\xi_2 \leq 0$, the value of q is toggled. This behavior corresponds to the jump set $D = \bigcup_{q\in Q}(D_q \times \{q\})$, where $D_q := \{\xi \in \mathbb{R}^2 : \xi_1 = 0, q\xi_2 \leq 0\}$. Thus, the jump set D is the union of two sets. One set corresponds to points where $q = 1$, $\xi_1 = 0$, and $\xi_2 \leq 0$, while the other set corresponds to points where $q = -1$, $\xi_1 = 0$, and $\xi_2 \geq 0$.

Figure 4 shows the flow and jump sets of the hybrid system. The figure also depicts the sinusoidal signal ξ_1 and the discrete state q obtained for initial conditions with ξ_1 starting at one, ξ_2 starting at zero, and q starting at one.

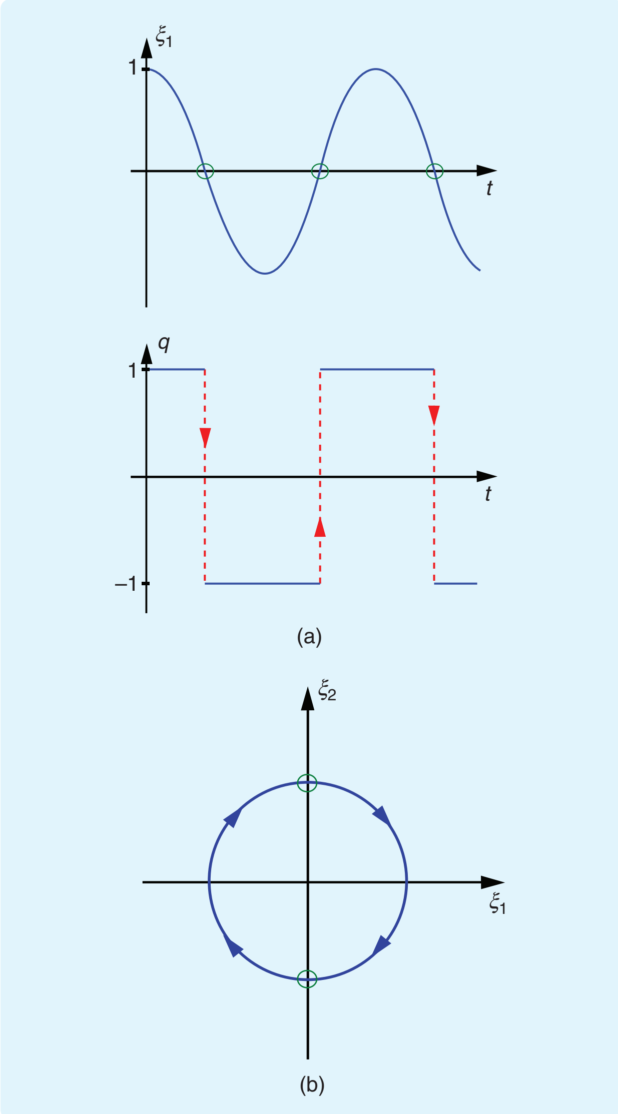

FIGURE 3 Detection of zero crossings of a sinusoidal signal. The sinusoidal signal ξ_1 is the output of the linear system $\dot{\xi}_1 = \omega\xi_2$, $\dot{\xi}_2 = -\omega\xi_1$, where $\omega > 0$. (a) The discrete state q is toggled at every zero crossing of the sinusoidal signal ξ_1. (b) The signal evolves in the $\xi = (\xi_1, \xi_2)$ plane.

Sample-and-Hold Control Systems

In a typical sample-and-hold control scenario, a continuous-time plant is controlled by a digital controller. The controller samples the plant's state, computes a control signal, and sets the plant's control input to the computed value. The controller's output remains constant between updates. Sample-and-hold devices perform analog-to-digital and digital-to-analog conversions.

As noted in [59], the closed-loop system resulting from this control scheme can be modeled as a hybrid system. Sampling, computation, and control updates in

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

sample-and-hold control are associated with jumps that occur when one or more timers reach thresholds defining the update rates. When these operations are performed synchronously, a single timer state and threshold can be used to trigger their execution. In this case, a sample-and-hold implementation of a control law samples the state of the plant and updates its input when a timer reaches the threshold $T > 0$, which defines the sampling period. During this update, the timer is reset to zero.

For the static, state-feedback law $u = \kappa(\xi)$ for the plant $\dot{\xi} = f(\xi, u)$, a hybrid model uses a memory state z to store the samples of u, as well as a timer state τ to determine when each sample is stored. The state of the resulting closed-loop system, which is depicted in Figure 5, is taken to be $x = (\xi, z, \tau)$.

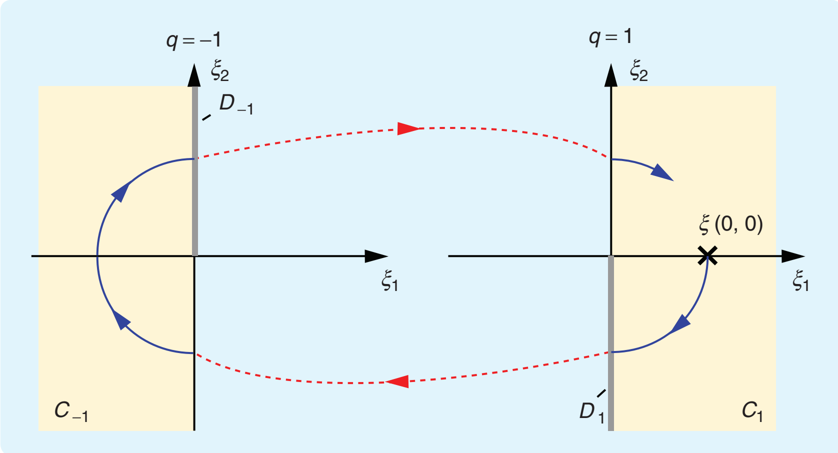

FIGURE 4 Flow and jump sets for each $q \in Q$ and trajectory to the hybrid system in "Explicit Zero-Crossing Detection." The trajectory starts from the initial condition at $(t, j) = (0, 0)$ given by $\xi_1(0, 0) = 1$, $\xi_2(0, 0) = 0$, $q(0, 0) = 1$. The jumps occur on the ξ_2 axis and toggle q. Flows are permitted in the left-half plane for $q = -1$ and in the right-half plane for $q = 1$.

During flow, which occurs until τ reaches the threshold T, the state of the plant evolves according to $\dot{\xi} = f(\xi, z)$, the value of z is kept constant, and τ grows at the constant rate of one. In other words, $\dot{z} = 0$ and $\dot{\tau} = 1$. This behavior corresponds to the flow set $C = \mathbb{R}^n \times \mathbb{R}^m \times [0, T]$, while the flow map is given by $F(x) = (f(\xi, z), 0, 1)$ for all $x \in C$.

When the timer reaches the threshold T, the timer state τ is reset to zero, the memory state z is updated to $\kappa(\xi)$, but the plant state ξ does not change. This behavior corresponds to the jump set $D := \mathbb{R}^n \times \mathbb{R}^m \times \{T\}$ and the jump map $G(x) := (\xi, \kappa(\xi), 0)$ for all $x \in D$.

Hybrid Controllers for Nonlinear Systems

Hybrid dynamical systems can model a variety of closed-loop feedback control systems. In some hybrid control applications the plant itself is hybrid. Examples include juggling [70], [73] and robot walking control [63]. In other applications, the plant is a continuous-time system that is controlled by an algorithm employing discrete-valued states. This type of control appears in a broad class of industrial applications, where programmable logic controllers and microcontrollers are employed for automation. In these applications, discrete states, as well as other variables in software, are used to implement control logic that incorporates decision-making capabilities into the control system.

Consider a plant described by the differential equation

$$\dot{x}_p = f_p(x_p, u), \tag{5}$$

where $x_p \in \mathbb{R}^n$, $u \in \mathbb{R}^r$, and f_p is continuous. A hybrid controller for this plant has state $x_c \in \mathbb{R}^m$, which can contain logic states, timers, counters, observer states, and other continuous-valued and discrete-valued states. A hybrid controller is defined by a flow set $C_c \subset \mathbb{R}^{n+m}$, flow map $f_c : C_c \to \mathbb{R}^n$, jump set $D_c \subset \mathbb{R}^{n+m}$, and a possibly set-valued jump map $G_c : \mathbb{R}^{n+m} \rightrightarrows \mathbb{R}^m$, together with a feedback law $\kappa_c : C_c \to \mathbb{R}^r$ that specifies the control signal u. Figure 6 illustrates this setup.

During continuous-time evolution, which can occur when the composite closed-loop state $x = (x_p, x_c)$ belongs to the set C_c, the controller state satisfies $\dot{x}_c = f_c(x)$ and the control signal is generated as $u = \kappa_c(x)$. At jumps, which are allowed when the closed-loop state belongs to D_c, the state of the controller is reset using the rule $x_c^+ \in G_c(x)$. The closed-loop system is a hybrid system with state $x = (x_p, x_c)$, flow set $C = C_c$, jump set $D = D_c$, flow map

$$F(x) = \begin{bmatrix} f_p(x_p, \kappa_c(x)) \\ f_c(x) \end{bmatrix} \quad \text{for all } x \in C, \tag{6}$$

and jump map

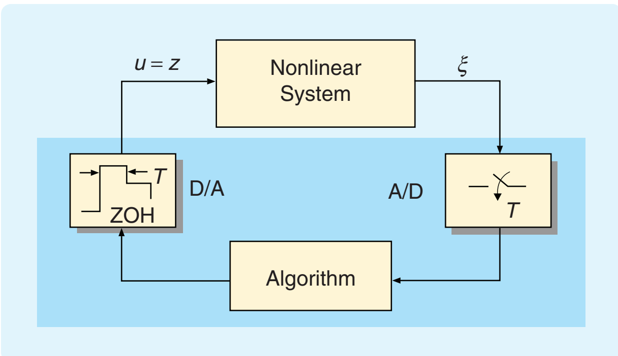

FIGURE 5 Digital control of a continuous-time nonlinear system with sample-and-hold devices performing the analog-to-digital (A/D) and digital-to-analog (D/A) conversions. Samples of the state ξ of the plant and updates of the control law $\kappa(\xi)$ computed by the algorithm are taken after each amount of time T. The controller state z stores the values of $\kappa(\xi)$.

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

The interaction of continuous- and discrete-time dynamics in a hybrid system leads to rich dynamical behavior and phenomena not encountered in purely continuous-time systems.

$$G(x) = \begin{bmatrix} x_p \\ G_c(x) \end{bmatrix} \quad \text{for all } x \in D. \tag{7}$$

One way hybrid controllers arise is through supervisory control. A supervisor oversees a collection of controllers and makes decisions about which controller to insert into the closed-loop system based on the state of the plant and the controllers. The supervisor associates to each controller a region of operation and a region where switching to other controllers is possible. These regions are subsets of the state space. In the region where changes between controllers are allowed, the supervisor specifies the controllers to which authority can be switched. In supervisory control, it is possible for the individual controllers to be hybrid controllers. Through this degree of flexibility, it is possible to generate hybrid control algorithms through a hierarchy of supervisors.

The following example features a supervisor for two state-feedback control laws.

Example 1: Dual-Mode Control for Disk Drives

Control of read/write heads in hard disk drives requires precise positioning on and rapid transitioning between tracks on a disk drive. To meet these dual objectives, some

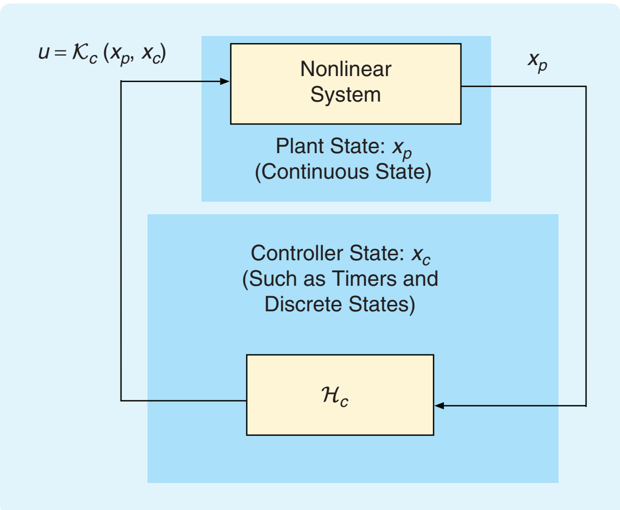

FIGURE 6 Closed-loop system consisting of a continuous-time nonlinear system and a hybrid controller. The nonlinear system has state x_p, which is continuous, and input u. The hybrid controller has state x_c, which has continuous state variables, such as timer states, and discrete state variables, such as logic modes. The control input $u = \kappa_c(x_p, x_c)$ to the nonlinear system is a function of the plant state x_p and the controller state x_c.

commercial hard disk drives use mode-switching control [27], [81], [87], which combines a track-seeking controller and a track-following controller. The track-seeking controller rapidly steers the magnetic head to a neighborhood of the desired track, while the track-following controller regulates position and velocity, precisely and robustly, to enable read/write operations. Mode-switching control uses the track-seeking controller to steer the magnetic head's state to a point where the track-following controller is applicable, and then switches the control input to the track-following controller. The control strategy results in a hybrid closed-loop system.

Let $p \in \mathbb{R}$ be the position and $v \in \mathbb{R}$ the velocity of the magnetic head in the disk drive. The dynamics can be approximated by the double integrator system $\dot{p} = v$, $\dot{v} = u$ [27], [87].

The hybrid controller for the magnetic head supervises both the track-seeking control law $u = \kappa_1(p, v, p^*)$ and the track-following control law $u = \kappa_2(p, v, p^*)$, where p^* is the desired position. We assume that the track-seeking control law globally asymptotically stabilizes the point $(p^*, 0)$, while the track-following control law locally asymptotically stabilizes the point $(p^*, 0)$. Let C_2 be a compact neighborhood of $(p^*, 0)$ that is contained in the basin of attraction for $(p^*, 0)$ when using the track-following control law, and let D_1 be a compact neighborhood of $(p^*, 0)$ such that solutions using the track-following control law that start in D_1 do not reach the boundary of C_2. Also define $C_1 = \overline{\mathbb{R}^2 \setminus D_1}$ and $D_2 = \overline{\mathbb{R}^2 \setminus C_2}$. Figure 7 illustrates these sets.

Let the controller state $q \in Q := \{1, 2\}$ denote the operating mode. The track-seeking mode corresponds to $q = 1$, while the track-following mode corresponds to $q = 2$. The mode-switching strategy uses the track-seeking controller when $(p, v) \in C_1$ and the track-following controller when $(p, v) \in C_2$. Figure 7 indicates the intersection of C_1 and C_2, where either controller can be used. To prevent chattering between the two controllers in the intersection of C_1 and C_2, the supervisor allows mode switching when $(p, v) \in D_q$. In other words, a switch from the track-seeking mode to the track-following mode can occur when $(p, v) \in D_1$, while a switch from the track-following mode to the track-seeking mode can occur when $(p, v) \in D_2$.

The hybrid controller executing this logic has the flow set $C_c = \bigcup_{q \in Q}(C_q \times \{q\})$, flow map $f_c(p, v, q) = 0$, jump set

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

$D_c := \bigcup_{q\in Q}(D_q \times \{q\})$, and jump map $G_c(p, v, q) = 3 - q$, which toggles q in the set $Q = \{1, 2\}$.

The idea behind this control construction applies to arbitrary nonlinear control systems and state-feedback laws [66]. ■

CONCEPT OF A SOLUTION

Generalized Time Domains

Solutions to continuous-time dynamical systems are parameterized by $t \in \mathbb{R}_{\geq 0}$, whereas solutions to discrete-time dynamical systems are parameterized by $j \in \mathbb{N}$. Parameterization by $t \in \mathbb{R}_{\geq 0}$ is possible for a continuous-time system even when solutions experience jumps, as long as at most one jump occurs at each time t. For example, parameterization by $t \in \mathbb{R}_{\geq 0}$ is used for switched systems [49] as well as for impulsive dynamical systems [43], [30]. However, parameterization by $t \in \mathbb{R}_{\geq 0}$ of discontinuous solutions to a dynamical system may be an obstacle for establishing sequential compactness of the space of solutions. For example, sequential compactness may require us to admit a solution with two jumps at the same time instant to represent the limit of a sequence of solutions for which the time between two consecutive jumps shrinks to zero. By considering a generalized time domain, we can overcome such obstacles and, furthermore, treat continuous- and discrete-time systems in a unified framework.

A subset E of $\mathbb{R}_{\geq 0} \times \mathbb{N}$ is a *hybrid time domain* [23], [26] if it is the union of infinitely many intervals of the form $[t_j, t_{j+1}] \times \{j\}$, where $0 = t_0 \leq t_1 \leq t_2 \leq \cdots$, or of finitely many such intervals, with the last one possibly of the form $[t_j, t_{j+1}] \times \{j\}$, $[t_j, t_{j+1}) \times \{j\}$, or $[t_j, \infty) \times \{j\}$. An example of a hybrid time domain is shown in Figure 8.

A hybrid time domain is called a hybrid time set in [17] and is equivalent to a generalized time domain [51] defined as a sequence of intervals, some of which may consist of one point. The idea of a hybrid time domain is present in the concept of a solution given in [4], which explicitly includes a nondecreasing sequence of jump times in the solution description.

More general time domains are sometimes considered. For details, see [54], [18], or the discussion of time scales in "Related Mathematical Frameworks." Some time domains make it possible to continue solutions past infinitely many jumps. For an initial exposition of hybrid system and for the analysis of many hybrid control algorithms, domains with this feature are not necessary.

Solutions

A solution to a hybrid system is a function, defined on a hybrid time domain, that satisfies the dynamics and constraints given by the data of the hybrid system. The data in (1), (2) has four components, which are the flow set, the flow map, the jump set, and the jump map. For

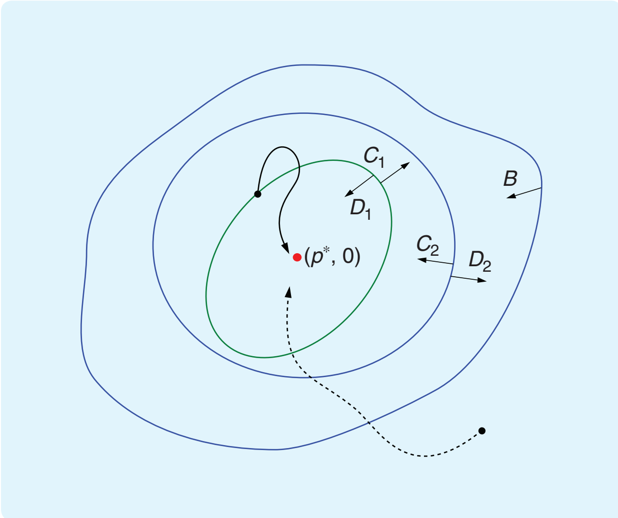

FIGURE 7 Sets of the hybrid controller for dual-mode control of disk drives. The flow and jump sets for the track-seeking mode $q = 1$ and the track-following mode $q = 2$ are constructed from the sets C_1, D_1 and C_2, D_2, respectively. The set $\mathcal{B}$ is the basin of attraction for $(p^*, 0)$ when the track-following controller is applied. In addition, this set contains the compact set D_1, from which solutions with the track-following controller do not reach the boundary of C_2, a compact subset of $\mathcal{B}$. This property is illustrated by the solid black solution starting from D_1. The dashed black solution is the result of applying the track-seeking controller, which steers solutions to the set D_1 in finite time.

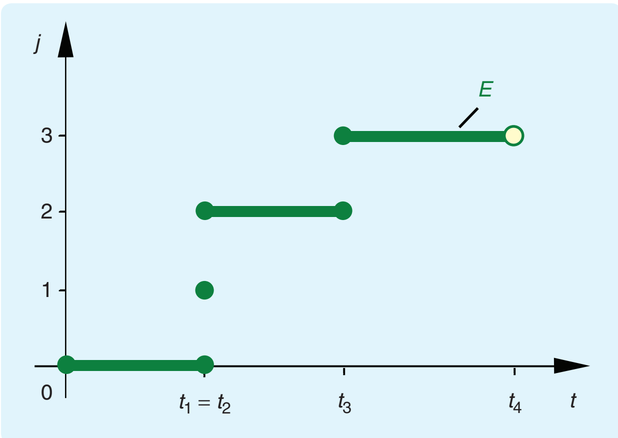

FIGURE 8 A hybrid time domain. The hybrid time domain, which is denoted by E, is given by the union of $[0, t_1] \times \{0\}$, $[t_2, t_2] \times \{1\}$, $[t_2, t_3] \times \{2\}$, and $[t_3, t_4) \times \{3\}$.

a hybrid system (1), (2) on $\mathbb{R}^n$, the flow set C is a subset of $\mathbb{R}^n$, the flow map is a set-valued mapping $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, the jump set D is a subset of $\mathbb{R}^n$, and the jump map is a set-valued mapping $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$. A set-valued mapping on $\mathbb{R}^n$ associates, to each $x \in \mathbb{R}^n$, a set in $\mathbb{R}^n$. A function is a set-valued mapping whose values can be viewed as sets that consist of one point.

A *hybrid arc* is a function $x : \operatorname{dom} x \to \mathbb{R}^n$, where $\operatorname{dom} x$ is a hybrid time domain and, for each

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

Related Mathematical Frameworks

Interest in hybrid systems grew rapidly in the 1990s with computer scientists and control systems researchers coming together to organize several international workshops. See [S8] and similar subsequent collections. Additional books dedicated to hybrid systems include [86] and [S10]. The legacy of the cooperative initiative with computer science is the successful conference "Hybrid Systems: Computation and Control (HSCC)," now a part of the larger "cyber-physical systems week," which includes real-time and embedded systems and information processing in sensor networks. At the same time, many mathematical frameworks related to hybrid systems have also appeared in the literature. Some of those frameworks are discussed below. Additional ideas appear in the concept of a discontinuous dynamical system, described in [S11] and [S12].

IMPULSIVE DIFFERENTIAL EQUATIONS

Impulsive differential equations consist of the classical differential equation $\dot{x}(t) = f(x(t))$, which applies at all times except the impulse times, and the equation $\Delta x(t_i) = I_i(x(t_i))$, which describes the impulsive behavior at impulse times. The impulse times are often fixed a priori for each particular solution and form an increasing sequence $t_1, t_2, \ldots$. In other words, a solution with the state $x(t_i)$ before the ith jump has the value $x(t_i) + I_i(x(t_i))$ after the jump. Solutions to impulsive differential equations are piecewise differentiable or piecewise absolutely continuous functions parameterized by time t. These functions cannot model multiple jumps at a single time instant.

An impulsive differential equation can be recast as a hybrid system in the case where the impulse times are determined by the condition $x(t) \in D$ for some set D. This situation requires some conditions on D and I_i to ensure that $x(t_i) + I_i(x(t_i)) \notin D$. For simplicity, consider the case where I_i is the same map I for each i. Then the corresponding hybrid system has the flow map f, the jump map $x \mapsto x + I(x)$, the jump set D, and the flow set C given by the complement of D.

Natural generalizations of impulsive differential equations include impulsive differential inclusions, where either f or I may be replaced by a set-valued mapping. For details, see [S6], [43], [S15], and [30].

MEASURE-DRIVEN DIFFERENTIAL EQUATIONS

The classical differential equation $\dot{x}(t) = f(x(t))$ can be rewritten as

$$dx(t) = f(x(t))\,dt.$$

Measure-driven differential equations are formulated as

$$dx(t) = f_1(x(t))\,dt + f_2(x(t))\,\mu(dt),$$

where f_1, f_2 are functions and μ is a nonnegative scalar or vector-valued Borel measure. Solutions to measure-driven differential equations are given by functions of bounded variation parameterized by t and are not necessarily differentiable, absolutely continuous, or even continuous. The discontinuous behavior is due to the presence of atoms in the measure μ. In control situations, the driving measure μ, in particular, the atoms of μ, can approximate time-dependent controls that take large values on short intervals.

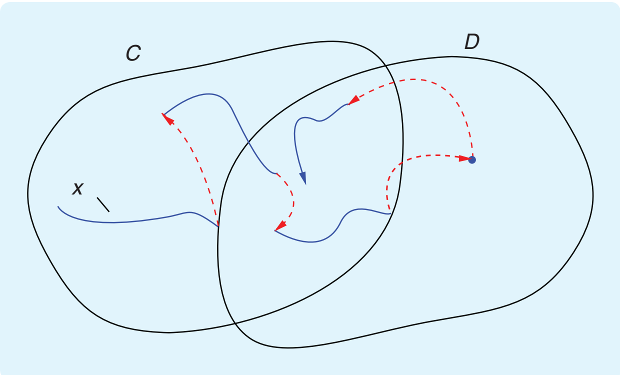

FIGURE 9 Evolution of a solution to a hybrid system. Flows and jumps of the solution x are allowed only on the flow set C and from the jump set D, respectively. The solid blue curves indicate flow. The dashed red arcs indicate jumps. The solid curves must belong to the flow set C. The dashed arcs must originate from the jump set D.

fixed j, $t \mapsto x(t,j)$ is a locally absolutely continuous function on the interval

$$I_j = \{t : (t,j) \in \operatorname{dom} x\}.$$

The hybrid arc x is a *solution to the hybrid system* $\mathcal{H} = (C, F, D, G)$ if $x(0,0) \in C \cup D$ and the following conditions are satisfied.

Flow Condition

For each $j \in \mathbb{N}$ such that I_j has nonempty interior,

$$\begin{aligned} \dot{x}(t,j) &\in F(x(t,j)) && \text{for almost all } t \in I_j, \\ x(t,j) &\in C && \text{for all } t \in [\min I_j, \sup I_j). \end{aligned}$$

Jump Condition

For each $(t,j) \in \operatorname{dom} x$ such that $(t, j+1) \in \operatorname{dom} x$,

$$x(t, j+1) \in G(x(t,j)),$$

$$x(t,j) \in D.$$

If the flow set C is closed and I_j has nonempty interior, then the requirement $x(t,j) \in C$ for all $t \in [\min I_j, \sup I_j)$ in the flow condition is equivalent to $x(t,j) \in C$ for

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

Natural generalizations, needed to analyze mechanical systems with friction or impacts [S13], include measure-driven differential inclusions, where f_1, f_2 are replaced by set-valued mappings. Formulating a robust notion of a solution to measure-driven differential inclusions is technically challenging [S9], [S14].

DYNAMICAL SYSTEMS ON TIME SCALES

A framework for unifying the classical theories of differential and difference equations is that of dynamical systems on time scales [S7]. Given a time scale $\mathbb{T}$, which is a nonempty closed subset of $\mathbb{R}$, a generalized derivative of a function $\phi:\mathbb{T}\to\mathbb{R}$ relative to $\mathbb{T}$ can be defined. This generalized derivative reduces to the standard derivative when $\mathbb{T}=\mathbb{R}$, and to the difference $\phi(n+1)-\phi(n)$ when evaluated at n and for $\mathbb{T}=\mathbb{N}$. As special cases, classical differential and difference equations can be written as systems on time scales. Systems on time scales can also be used to model populations that experience a repeated pattern consisting of continuous evolution followed by a dormancy [S7, Ex. 1.39].

Consider a time scale $\mathbb{T}$ that is unbounded to the right, and, for $t\in\mathbb{T}$, define $\sigma(t)=\inf\{s\in\mathbb{T}:s>t\}$. The generalized derivative of $\phi:\mathbb{T}\to\mathbb{R}$ at $t\in\mathbb{T}$ is the number $\phi^{\Delta}(t)$, if it exists, such that, for each $\varepsilon>0$ and each $s\in\mathbb{T}$ sufficiently close to t,

$$|[\phi(\sigma(t))-\phi(s)]-\phi^{\Delta}(t)[\sigma(t)-s]|\le\varepsilon|\sigma(t)-s|.$$

The function ϕ is differentiable if ϕ^{Δ} exists at every $t\in\mathbb{T}$. A dynamical system on the time scale $\mathbb{T}$ has the form

$$x^{\Delta}(t)=f(x(t))\quad\text{for every } t\in\mathbb{T}.$$

One advantage of the framework of dynamical systems on time scales is the generality of the concept of a time scale. A drawback, especially for control engineering purposes, is that a time scale is chosen a priori, and all solutions to a system are defined on the same time scale.

REFERENCES

[S6] D. D. Bainov and P. S. Simeonov, *Systems with Impulse Effect: Stability, Theory, and Applications*. Chichester, U.K.: Ellis Horwood, 1989.

[S7] M. Bohner and A. Peterson, *Dynamic Equations on Time Scales*. Cambridge, MA: Birkhaüser, 2001.

[S8] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. *Hybrid Systems*. New York: Springer-Verlag, 1993.

[S9] G. Dal Maso and F. Rampazzo, "On systems of ordinary differential equations with measures as controls," *Differ. Integr. Equ.*, vol. 4, pp. 739–765, 1991.

[S10] A. S. Matveev and A. V. Savkin, *Qualitative Theory of Hybrid Dynamical Systems*. Cambridge, MA: Birkhaüser, 2000.

[S11] A. N. Michel, L. Wang, and B. Hu, *Qualitative Theory of Dynamical Systems: The Role of Stability Preserving Mappings*. New York: Marcel Dekker, 2001.

[S12] A. N. Michel, L. Hou, and D. Liu, *Stability of Dynamical Systems: Continuous, Discontinuous, and Discrete Systems*. Cambridge, MA: Birkhaüser, 2008.

[S13] J.-J. Moreau, "Unilateral contact and dry friction in finite freedom dynamics," in *Non-smooth Mechanics and Applications*, New York: Springer-Verlag, 1988, pp. 1–82.

[S14] G. N. Silva and R. B. Vinter, "Measure driven differential inclusions," *J. Math. Anal. Applicat.*, vol. 202, no. 3, pp. 727–746, 1996.

[S15] T. Yang, *Impulsive Control Theory*. Berlin: Springer-Verlag, 2001.

all $t\in I_j$ and is also equivalent to $x(t,j)\in C$ for almost all $t\in I_j$.

The solution x to a hybrid system is *nontrivial* if dom x contains at least one point different from $(0,0)$; *maximal* if it cannot be extended, that is, the hybrid system has no solution x' whose domain dom x' contains dom x as a proper subset and such that x' agrees with x on dom x; and complete if dom x is unbounded. Every complete solution is maximal.

Figure 9 shows a solution to a hybrid system flowing, as solutions to continuous-time systems do, at points in the flow set C, and jumping, as solutions to discrete-time systems do, from points in the jump set D. At points where D overlaps with the interior of C, solutions can either flow or jump. Thus, the jump set D enables rather than forces jumps. To force jumps from D, the flow set C can be replaced by either the set $C\setminus D$ or the set $\overline{C\setminus D}$.

The parameterization of a solution x by $(t,j)\in\operatorname{dom} x$ means that $x(t,j)$ represents the state of the hybrid system after t time units and j jumps. Figure 10 shows a

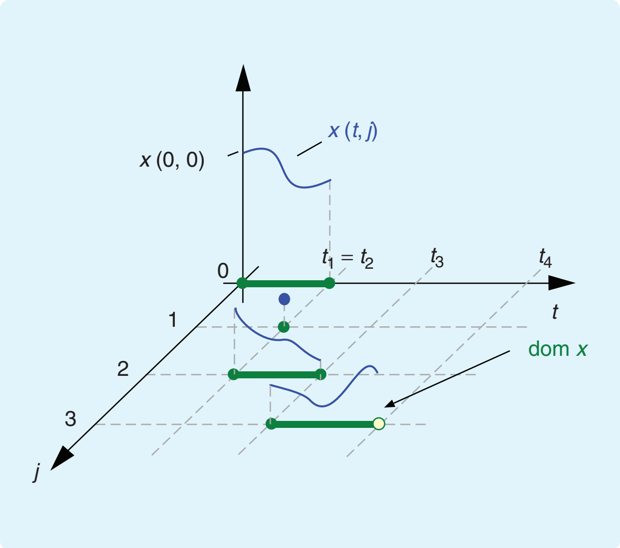

FIGURE 10 A solution to a hybrid system. The solution, which is denoted by x, has initial condition $x(0,0)$, is given by a hybrid arc, and has hybrid time domain dom x. The hybrid time domain E in Figure 8 is equal to dom x.

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.

solution to a hybrid system and illustrates the parameterization by (t, j).

Every solution x to a hybrid system has a hybrid time domain $\operatorname{dom} x$ associated with it. However, for a given hybrid system, not every hybrid time domain is the domain of a solution to this system. This phenomenon goes beyond what can happen in unconstrained continuous-time systems, where solutions may blow up in finite time, and, hence, may be defined on only a bounded subset of $[0, \infty)$. For example, for a hybrid system in which two jumps cannot occur at the same time instant, a hybrid time domain that contains (t, j), $(t, j+1)$, and $(t, j+2)$ for some $t \in \mathbb{R}_{\geq 0}$, $j \in \mathbb{N}$ is not a domain of any solutions. Hence, we do not pick a hybrid time domain and then look for a solution to a hybrid system on that time domain. Rather, the domain must be generated along with the solution. Example 2 shows how a hybrid system can have complete solutions with different domains.

Example 2: Solutions to a Hybrid System

Consider the hybrid system in $\mathbb{R}^2$ given by $D = \{x \in \mathbb{R}^2 : x_1 \leq x_2 \leq 3\}$, $C = \mathbb{R}^2 \setminus D$, $f(x) = (-1, 1)$ for all $x \in C$, and $g(x) = (2x_1^2, 0)$, as depicted in Figure 11. Solutions flow in C with velocity $(-1, 1)$ and jump from points $x = (x_1, x_2) \in D$ to $(2x_1^2, 0)$. We consider maximal solutions from points of the form $(z, 0)$ with $z \in \{8, 4, 2, 1, 0\}$.

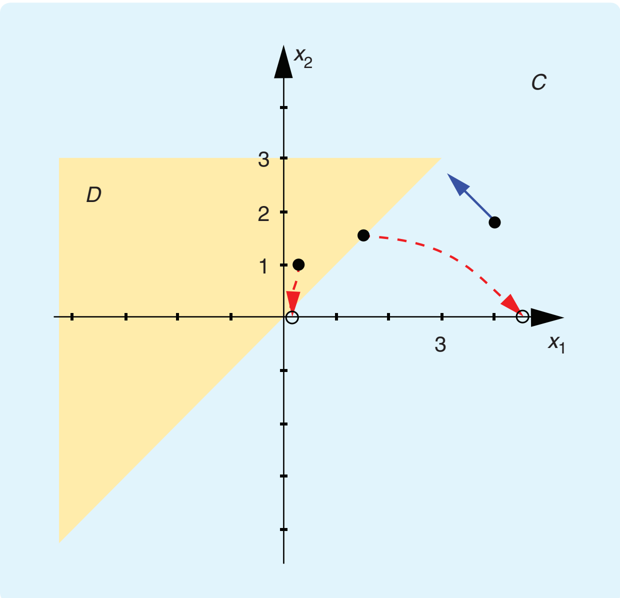

FIGURE 11 Data for the hybrid system in Example 2. The jump set $D = \{x \in \mathbb{R}^2 : x_1 \leq x_2 \leq 3\}$ is the shaded region. The flow set $C = \mathbb{R}^2 \setminus D$ is the complement of the jump set D. The flow map is $f(x) = (-1, 1)$ for all $x \in C$; hence solutions flow, when in C, with velocity $(-1, 1)$. The jump map is $g(x) = (2x_1^2, 0)$, where $x = (x_1, x_2)$; hence solutions jump from points $x \in D$ to $(2x_1^2, 0)$. A solution that starts in C with $x_2 \geq 3$ or with $x_1 + x_2 \geq 6$ flows and never jumps. Otherwise, a solution starting in C flows toward D, reaches D on the line $x_1 = x_2$, and then jumps to a point on the nonnegative x_1-axis. A solution that starts in D jumps to a point on the nonnegative x_1-axis.

The maximal solution starting from $(8, 0)$, denoted by x_a, flows and never hits the jump set D. More precisely, $\operatorname{dom} x_a = \mathbb{R}_{\geq 0} \times \{0\}$ and $x_a(t, 0) = (8 - t, t)$ for $t \geq 0$.

The maximal solution starting from $(4, 0)$, denoted by x_b, hits the jump set at the point $(2, 2)$, jumps to $(8, 0)$, and then flows without hitting the jump set D again. More precisely, $\operatorname{dom} x_b = [0, 2] \times \{0\} \cup [2, \infty) \times \{1\}$ and

$$x_b(t, j) = \begin{cases} (4 - t, t) & \text{for} \quad 0 \leq t \leq 2, j = 0, \\ (8 - (t - 2), t - 2) & \text{for} \quad 2 \leq t, j = 1. \end{cases}$$

The maximal solution starting from $(2, 0)$, denoted by x_c, hits the jump set at $(1, 1)$, jumps to $(2, 0)$, and repeats this behavior infinitely many times. More precisely, $\operatorname{dom} x_c = \bigcup_{j=0}^{\infty} [j, j+1] \times \{j\}$ and

$$x_c(t, j) = (2 - (t - j), t - j) \text{ for } j = 0, 1, 2, \ldots, \; t \in [j, j+1].$$

The maximal solution starting from $(1, 0)$, denoted by x_d, jumps infinitely many times and converges to the origin. More precisely, $\operatorname{dom} x_d = \bigcup_{j=0}^{\infty} [t_j, t_{j+1}] \times \{j\}$, where $a_0 = 1$, $a_j = a_{j-1}^2/2$ for $i = 1, 2, \ldots$, $t_j = \sum_{i=0}^{j-1} a_i/2$ and

$$x_d(t, j) = (a_j - (t - t_j), t - t_j) \quad \text{for } j = 0, 1, 2, \ldots, \; t \in [t_j, t_{j+1}].$$

Finally, the maximal solution starting from $(0, 0)$, denoted by x_e, never flows, has $\operatorname{dom} x_e = \{0\} \times \mathbb{N}$ and is given by $x_e(0, j) = (0, 0)$ for all $j \in \mathbb{N}$.

The solutions x_a, x_b, x_c, x_d, and x_e are complete. Note that solutions x_d and x_e are complete even though $\operatorname{dom} x_d$ and $\operatorname{dom} x_e$ do not contain points (t, j) with arbitrarily large t. ■

BASIC MATHEMATICAL PROPERTIES

Basic questions about solutions to dynamical systems concern existence, uniqueness, and dependence on initial conditions and other parameters. Existence and uniqueness questions for hybrid systems are addressed in "Existence, Uniqueness, and Other Well-Posedness Issues." The dependence of solutions on initial conditions, and compactness of the space of solutions to hybrid systems, are essential tools for developing a stability theory for hybrid systems. These tools depend on the concept of graphical convergence of hybrid arcs.

Authorized licensed use limited to: UNIVERSITY OF NOTTINGHAM. Downloaded on February 02,2026 at 10:11:14 UTC from IEEE Xplore. Restrictions apply.