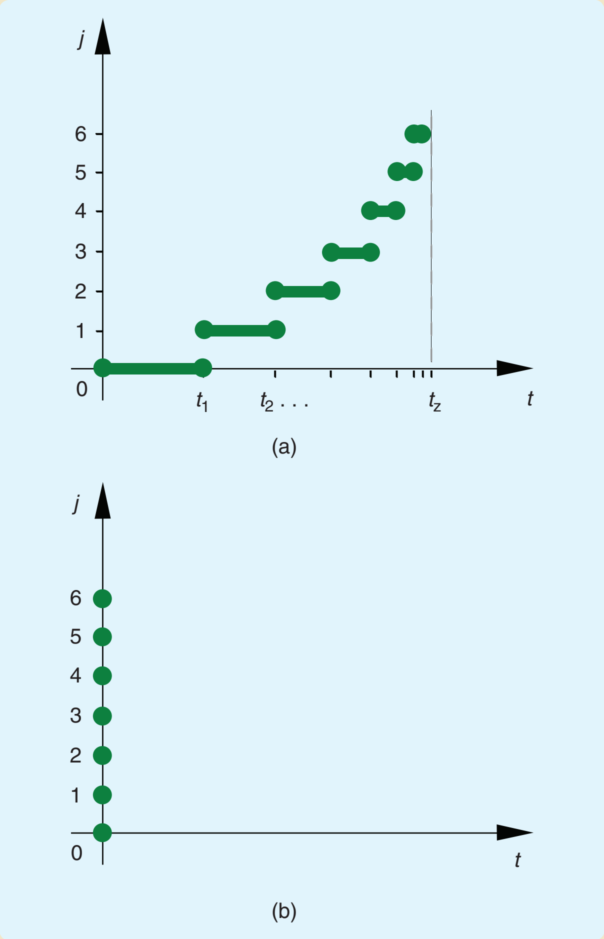

FIGURE S11 Hybrid time domains associated to two solutions of the Achilles system. (a) The hybrid time domain for a solution starting from the condition $\alpha < \tau$ and $\rho \in [\alpha, \tau]$. The hybrid time domain is unbounded in the j direction and bounded in the t direction by the Zeno time, which is denoted by t_Z. Solutions with domains of this type are called Zeno solutions. (b) The hybrid time domain for a solution starting from the condition $\alpha = \rho = \tau$. Its domain $\mathrm{dom}\, x$ is a subset of $\{0\} \times \mathbb{N}$. Solutions with domains of this type are called discrete solutions.

The bouncing ball model discussed in the main text is a physically based model that exhibits similar behavior, as illustrated in Figure S16.

In honor of Zeno and his Tortoise, sets $\mathcal{A}$ as in the Achilles system or the bouncing ball are often called Zeno attractors. Researchers have investigated sufficient conditions [36], [S50], [S51], and necessary conditions [45] for the existence of Zeno equilibria and have also grappled with the question of how to continue solutions past their "Zeno times." See, for example, [S52] and [18]. Some recent contributions to the characterization of Zeno equilibria can be found in [S54] and [S55].

Stabilizing hybrid feedback can induce Zeno equilibria or, more generally, solutions having a hybrid time domain bounded in the ordinary time direction. Such solutions sometimes exist in reset control systems [S53], [S58]. When asymptotically stable Zeno equilibria are induced by hybrid control, the equilibria can be made non-Zeno by introducing temporal regularization into the control algorithm. This operation can be carried out while preserving asymptotic stability in a practical sense. Results of this type, which follow from the robustness results discussed in the main text, are presented in [26]. Temporal regularization also appears in [S56].

The behavior of hybrid systems is rich and sometimes unexpected. Solutions with domains bounded in the ordinary time direction, including Zeno solutions, are a fascinating example of this behavior.

REFERENCES

[S50] A. D. Ames, A. Abate, and S. Sastry, "Sufficient conditions for the existence of Zeno behavior," in *Proc. IEEE Conf. Decision and Control and European Control Conf.*, Seville, Spain, 2005, pp. 696–701.
[S51] A. D. Ames, A. Abate, and S. Sastry, "Sufficient conditions for the existence of Zeno behavior in a class of nonlinear hybrid systems via constant approximations," in *Proc. 46th IEEE Conf. Decision and Control*, New Orleans, LA, Dec. 2007, pp. 4033–4038.
[S52] A. D. Ames, H. Zheng, R. D. Gregg, and S. Sastry, "Is there life after Zeno? Taking executions past the breaking (Zeno) point," in *Proc. American Control Conf.*, Minneapolis, MN, 2006, pp. 2652–2657.
[S53] O. Beker, C. V. Hollot, and Y. Chait, "Fundamental properties of reset control systems," *Automatica*, vol. 40, pp. 905–915, 2004.
[S54] R. Goebel and A. R. Teel, "Lyapunov characterization of Zeno behavior in hybrid systems," in *Proc. 47th IEEE Conf. Decision and Control*, Cancun, Mexico, Dec. 2008, pp. 2752–2757.
[S55] R. Goebel and A. R. Teel, "Zeno behavior in homogeneous hybrid systems," in *Proc. 47th IEEE Conf. Decision and Control*, Cancun, Mexico, Dec. 2008, pp. 2758–2763.
[S56] K. H. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry, "On the regularization of Zeno hybrid automata," *Syst. Control Lett.*, vol. 38, no. 3, pp. 141–150, 1999.
[S57] D. Konstan, *Simplicius: On Aristotle's Physics 6* (Transl.). London: Duckworth, 1989.
[S58] D. Nesic, L. Zaccarian, and A. R. Teel, "Stability properties of reset systems," *Automatica*, vol. 44, no. 8, pp. 2019–2026, 2008.
[S59] J. Sachs, *Aristotle's Physics: A Guided Study*. New Brunswick, NJ: Rutgers Univ. Press, 1995.

R3') For each $c > 0$ and $c_2 > 0$, there are no complete solutions to the systems

$$\dot{x} = F(x), \quad x \in \{x=(\xi,\eta) \in L_V(c) : V_2(x) \le \rho(V_1(\xi))\}$$

and

$$\dot{x} = F(x),$$
$$x \in \{x = (\xi,\eta) \in L_V(c) \cap L_{V_2}(c_2) : V_2(x) \ge \rho(V_1(\xi))\}.$$

The fact that R3') implies R3) follows from Proposition 24 with $W := -V_2$ and the fact that the set $\{x = (\xi,\eta) : V_2(x) \ge \rho(V_1(\xi))\}$ is forward invariant for the system

$$\dot{x} = F(x), \quad x \in \{x \in L_V(c) : \langle \nabla V_2(x), F(x)\rangle \ge 0\}.$$

Finally, note that when the set D satisfies R3), condition R3) is also satisfied for any closed set that contains D and does not intersect $\{(\xi,\eta) : \eta = \eta_*(\xi)\}$ except at the origin. So, for example, jumps might be enabled when x belongs to the set D defined in (37) and also when x belongs to the set $\{x=(\xi,\eta): V_2(x)\geq V_1(\xi)\}$. This observation is related to the thermodynamic stabilization technique described in [29] and [30]. ■

STABILITY ANALYSIS THROUGH LIMITED EVENTS

Background and Definitions

The concepts of average dwell-time switching [34] and multiple Lyapunov functions [6], [19], which are applicable to switched systems, extend to hybrid systems. These results for switching systems are based on how frequently switches occur or on the value of a Lyapunov function when switches occur. A switch between systems can be generalized to the occurrence of an event in a hybrid system, where an event is a particular type of jump. For example, in a networked control system, an event might correspond to a packet dropout, which can be modeled as a jump that is different from a jump that corresponds to successful communication. For more details, see "Example 27 Revisited." As another example, in a multimodal sampled-data controller, a timer state jump, which captures the sampled-data nature of the system, would typically not count as an event, whereas a jump corresponding to a change in the mode may correspond to an event. See Example 29.

We discuss below when pre-asymptotic stability of a compact set for a hybrid system can be deduced from pre-asymptotic stability of the set for the system with its events removed.

Events are defined by an event indicator, which is an outer semicontinuous set-valued mapping $\mathcal{E}:\mathbb{R}^n\times\mathbb{R}^n\rightrightarrows\{0\}$. An event is a pair (g,x) such that $\mathcal{E}(g,x)$ is empty. The number of events experienced by a hybrid arc x is given by the cardinality of the set

$$\{j: \text{for some } t\geq 0,\ (t,j),(t,j+1)\in\operatorname{dom}x,\\ \mathcal{E}(x(t,j+1),x(t,j))=\varnothing\}.$$

Example 27 Revisited: Networked Control Systems with Transmission Dropouts

Consider a networked control system with the possibility of communication dropouts, modeled by the hybrid system with data $C:=\mathbb{R}^{n_1+n_2}\times[0,T]\times\{0,1\}$, $D:=\mathbb{R}^{n_1+n_2}\times\{T\}\times\{0,1\}$,

$$F(x):=\begin{bmatrix}F_1(x_1,x_2)\\F_2(x_1,x_2)\\1\\0\end{bmatrix}\quad\text{for all } x\in C,$$

$$G(x):=\left\{\begin{bmatrix}x_1\\ g_4G_2(x_1,x_2)+(1-g_4)x_2\\ [0,\varepsilon T]\\ g_4\end{bmatrix}: g_4\in\{0,1\}\right\}$$

$$\text{for all } x\in D,$$

where $\varepsilon\in[0,1)$, $g_4=1$ corresponds to successful transmission and $g_4=0$ corresponds to unsuccessful transmission. If we associate events with unsuccessful transmission then the event indicator is given by $\mathcal{E}(g,x)=0$ when $g_4=1$ and $\mathcal{E}(g,x)=\varnothing$ otherwise. Solutions that do not experience events behave like the networked control systems analyzed in Example 27. ■

Example 29: Mode Transitions as Events

Consider a hybrid system with state $x=(\xi,q)\in\mathbb{R}^n\times\mathbb{R}$, and data (C,F,D,G), where $C\bigcup D\subset\mathbb{R}^n\times Q$ and Q is a finite set. Suppose we want to identify jumps that change the value of q as events. For $x=(\xi,q)\in D$ and $(g_\xi,g_q)\in G(x)$, we let $\mathcal{E}(g,x)=0$ if $g_q=q$, and we let $\mathcal{E}(g,x)=\varnothing$ otherwise.

A refinement is to associate events with jumps that change the mode q only when the state is outside of a target compact set $\mathcal{A}\times Q$. In this case, we let $\mathcal{E}(g,x)=0$ if $g_q=q$ or $x=(\xi,q)\in\mathcal{A}\times Q$. Otherwise, we let $\mathcal{E}(g,x)=\varnothing$. ■

For a hybrid system $\mathcal{H}:=(C,F,D,G)$, the eventless hybrid system $\mathcal{H}^0:=(C,F,D^0,G^0)$, with data contained in the data of $\mathcal{H}$, is constructed by removing events from (G,D) through the definitions

$$G^0(x):=G(x)\cap\{v:\mathcal{E}(v,x)=\{0\}\}\ \text{for all } x\in D,\tag{38}$$

$$D^0:=D\cap\{x:G^0(x)\neq\varnothing\}.\tag{39}$$

The data of $\mathcal{H}^0$ satisfies the Basic Assumptions and the solutions to $\mathcal{H}^0$ experience no events.

Example 30: Jumps Out of $C\bigcup D$

For a hybrid system $\mathcal{H}=(C,F,D,G)$, it is possible that $g\notin C\bigcup D$ for some $x\in D$ and $g\in G(x)$. These jumps can be inhibited with an event indicator $\mathcal{E}$ satisfying $\mathcal{E}(g,x)=0$ if $g\in C\bigcup D$ and $\mathcal{E}(g,x)=\varnothing$ otherwise. In this case, the definition (38) gives $G^0(x)=G(x)\cap(C\bigcup D)$ for each $x\in D$. Each solution of $\mathcal{H}$ experiences no more than one event, since a solution cannot be extended from a point $g\notin C\bigcup D$.

According to Theorem 31, the compact set $\mathcal{A}$ is globally pre-asymptotically stable for the hybrid system (C,F,D,G) whenever $G(D\cap\mathcal{A})\subset\mathcal{A}$, and $\mathcal{A}$ is globally pre-asymptotically stable for the hybrid system (C,F,D^0,G^0), where $G^0(x)=G(x)\cap(C\bigcup D)$ for each $x\in D$ and D^0 is the set of points $x\in D$ for which $G^0(x)\neq\varnothing$. ■

The stability properties of $\mathcal{H}^0=(C,F,D^0,G^0)$, with D^0 and G^0 defined in (38)–(39), can be used to derive stability properties for $\mathcal{H}=(C,F,D,G)$ under a variety of conditions. Below, we consider the cases where 1) the solutions of $\mathcal{H}$ experience a finite number of events or 2) there is a relationship between the values of a Lyapunov function when different events occur. Other results can be formulated that involve an average dwell-time requirement on events, like for switched systems discussed in "Switching Systems" or

Pre-asymptotic stability for compact sets is equivalent to uniform pre-asymptotic stability, which is sometimes called KL-stability.

that combine dwell-time conditions and Lyapunov conditions, as in [33].

Finite Number of Events

An approach used in adaptive supervisory control to establish convergence properties is to establish that the number of switches experienced by a solution is finite [58]. A related result for asymptotic stability in hybrid systems is given in the next theorem.

Theorem 31

Suppose that the hybrid system $\mathcal{H} = (C, F, D, G)$ with state $x \in \mathbb{R}^n$ satisfies the Basic Assumptions. Let the compact set $\mathcal{A} \subset \mathbb{R}^n$ satisfy $G(D \cap \mathcal{A}) \subset \mathcal{A}$, and assume that $\mathcal{A}$ is globally pre-asymptotically stable for the eventless hybrid system $\mathcal{H}^0 = (C, F, D^0, G^0)$, where D^0 and G^0 are defined in (38)–(39). Also suppose that, for the hybrid system $\mathcal{H}$ and each compact set $K \subset \mathbb{R}^n$, there exists $N > 0$ such that each solution starting in K experiences no more than N events. Then the set $\mathcal{A}$ is globally pre-asymptotically stable for the system $\mathcal{H}$.

Theorem 31 is behind the proof of asymptotic stability for the mode-switching controller for the disk drive system in Example 1. As shown for "Example 1 Revisited," the origin of the closed-loop system is pre-asymptotically stable when jumps are eliminated, and when jumps are allowed the maximum number of jumps is two.

An outline of the proof of Theorem 31 is as follows. Boundedness of maximal solutions and global pre-attractivity of $\mathcal{A}$ for $\mathcal{H}$ follow from the fact that the number of events is finite and that the maximal solutions of $\mathcal{H}^0$ are bounded and the complete ones converge to $\mathcal{A}$. Stability can be established by concatenating trajectories of the system $\mathcal{H}^0$, which can include jumps but not events, with jumps from G that correspond to events, and repeating this process up to N times. In particular, let $\varepsilon =: \varepsilon_0 > 0$ be given. Since $G(D \cap \mathcal{A}) \subset \mathcal{A}$ and G is outer semicontinuous and locally bounded, there exists $\delta_0 > 0$ such that $G(D \cap (\mathcal{A} + \delta_0 \mathbb{B})) \subset \mathcal{A} + \varepsilon_0 \mathbb{B}$. Since $\mathcal{A}$ is stable for $\mathcal{H}^0$, there exists $\varepsilon_1 > 0$ such that, for each solution x_1 to $\mathcal{H}^0$, $|x_1(0,0)|_{\mathcal{A}} \le \varepsilon_1$ implies $|x_1(t,j)|_{\mathcal{A}} \le \delta_0$ for all $(t,j) \in \operatorname{dom} x_1$. Repeating this construction N times and letting ε_j generate ε_{j+1} for $j \in \{0, \ldots, N-1\}$, it follows that $|x_1(0,0)|_{\mathcal{A}} \le \varepsilon_N$ implies $|x_1(t,j)|_{\mathcal{A}} \le \varepsilon_0 = \varepsilon$ for all solutions x of $\mathcal{H}_N$ and all $(t,j) \in \operatorname{dom} x$.

Multiple Lyapunov Functions

Stability results involving multiple Lyapunov functions [6], [19] for switched systems, as well as related results for discontinuous dynamical systems in [90] and for differential equations in [1], extend to hybrid systems. The stability conditions are expressed in terms of an event indicator $\mathcal{E}$, the eventless system with events inhibited, as in (38)–(39), and a decomposition of the event indicator into a finite number of event types.

Theorem 32

For a hybrid system $\mathcal{H} = (C, F, D, G)$ satisfying the Basic Assumptions with an event indicator $\mathcal{E}$, the compact set $\mathcal{A}$ is globally pre-asymptotically stable if and only if the following conditions are satisfied:

1) $G(D \cap \mathcal{A}) \subset \mathcal{A}$,
2) $\mathcal{A}$ is globally pre-asymptotically stable for the system without events $\mathcal{H}^0 := (C, F, D^0, G^0)$.
3) There exist a function $W: C \cup D \to \mathbb{R}_{\ge 0}$, and class-$\mathcal{K}_\infty$ functions α_1, α_2, α_3, such that
$$\alpha_1(|x|_{\mathcal{A}}) \le W(x) \le \alpha_2(|x|_{\mathcal{A}}) \text{ for all } x \in C \cup D,$$
$$W(g) \le \alpha_3(W(x)) \text{ for all } x \in D,\ g \in G(x).$$
4) There exists a positive integer ℓ and outer semicontinuous mappings $\mathcal{E}_i: \mathbb{R}^n \times \mathbb{R}^n \rightrightarrows \{0\}$, $i = 1, \ldots, \ell$, such that, for each (v,x) satisfying $\mathcal{E}(v,x) = \varnothing$, there exists $i \in \{1, \ldots, \ell\}$ satisfying $\mathcal{E}_i(v,x) = \varnothing$, and there exists a continuous function α_4 satisfying $0 < \alpha_4(s) < s$ for all $s > 0$ and such that, for each $i \in \{1, \ldots, \ell\}$ and each solution x,
$$W(x(t_2, j_2)) \le \alpha_4(W(x(t_1, j_1))),$$
where $(t_k, j_k) \in \operatorname{dom} x$, $k \in \{1, 2\}$, $t_2 + j_2 > t_1 + j_1$, are the two smallest elements of $\operatorname{dom} x$ satisfying $\mathcal{E}_i(x(t_k, j_k), x(t_k, j_k - 1)) = \varnothing$.

Figure 17 indicates a possible evolution of $W(x(t,j))$. Some jumps, such as the one from $(t_2, 1)$ to $(t_2, 2)$, do not correspond to events. When a particular type of event occurs twice, the value of W at the second occurrence is required to be strictly less than the value of W at the first occurrence of the event. In the figure, one type of event is indicated in blue while the other is indicated in red. Blue events occur at $(t_1, 1)$ and $(t_4, 4)$. Thus, the value of $W(x(t_4, 4))$ is required to be less than the value of $W(x(t_1, 1))$.

Consider Theorem 32 in the context of multiple Lyapunov functions V_q, where $q \in \{1, \ldots, \ell\}$, for a switching system with state z. We take $W(x) := V_q(z)$, where $x = (z, q)$. Sometimes it is assumed that V_q does not increase during flows [6], although this property is not necessary [90], and is not assumed in Theorem 32. For $i \in \{1, \ldots, \ell\}$, we take $\mathcal{E}_i(v,x) = 0$ for switches that turn the state q into a value $j \ne i$, otherwise we take $\mathcal{E}_i(v,x) = \varnothing$. Then the third assumption

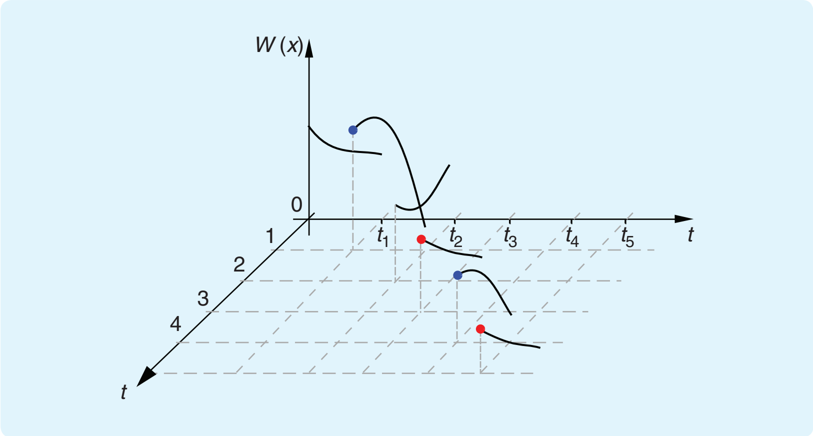

FIGURE 17 The evolution of a candidate function W for Theorem 32 along a trajectory of a hybrid system. The blue and red dots indicate different types of events. The blue dots, at $(t_1, 1)$ and $(t_4, 4)$, denote points $v \in G(x)$ at which $\mathcal{E}_1(v, x) = \emptyset$, whereas the red dots, at $(t_3, 3)$ and $(t_5, 5)$, denote points $v \in G(x)$ at which $\mathcal{E}_2(v, x) = \emptyset$. Some jumps, such as the one from $(t_2, 1)$ to $(t_2, 2)$, do not correspond to events. When a particular type of event occurs twice, the value of W at the second occurrence must be less than the value of W at the first occurrence of the event. For example, the value of $W(x(t_4, 4))$ must be less than the value of $W(x(t_1, 1))$.

of Theorem 32 asks that, when comparing W at an event where q changes to i to the value of W at the previous event where q changes to i, we see a decrease in W. The case of q belonging to a compact set can be handled in a similar way by generating a finite covering of the compact set, to generate a finite number of event indicators $\mathcal{E}_i$, and using extra continuity properties for the function $(z, q) \mapsto V_q(z)$ to ensure a decrease when an event repeats.

The necessity in Theorem 32 follows from the results discussed in "Converse Lyapunov Theorems," which show that W can always be taken to be smooth. See also [90]. The sufficiency, which doesn't use continuity properties for W, follows using the same argument used for multiple Lyapunov functions in [6]. We sketch the idea here.

For stability, note that due to the first two assumptions and Theorem 31, there exists $\alpha \in \mathcal{K}_\infty$ such that, for each solution x and each $(t, j) \in \operatorname{dom} x$ such that only one event has occurred

$$W(x(t,j)) \le \alpha(W(x(0,0))).$$

Without loss of generality, we associate the first event with the event indicator $i = 1$, and we use condition 3) of Theorem 32 to conclude that, for all event times (t_{k_1}, j_{k_1}) where event $i = 1$ occurs, that is, $\mathcal{E}_1(x(t_{k_1}, j_{k_1}), x(t_{k_1}, j_{k_1} - 1)) = \emptyset$, we have $W(x(t_{k_1}, j_{k_1})) \le \alpha(W(x(0,0)))$. We associate the first event that is different from events with $i = 1$ with event indicator $i = 2$ and conclude that, for all event times (t_{k_2}, j_{k_2}) at which event $i = 2$ occurs, that is, $\mathcal{E}_2(x(t_{k_2}, j_{k_2}), x(t_{k_2}, j_{k_2} - 1)) = \emptyset$, we have $W(x(t_{k_2}, j_{k_2})) \le \alpha^2(W(x(0,0)))$. Continuing in this way, we conclude that, for each solution x and each $(t, j) \in \operatorname{dom} x$

$$W(x(t,j)) \le \alpha^{\ell}(W(x(0,0))).$$

With the first assumption of the theorem, we conclude that the set $\mathcal{A}$ is stable.

Pre-attractivity is established as follows. Complete solutions that experience a finite number of events converge to $\mathcal{A}$ according to the first assumption of Theorem 32. For complete solutions that experience an infinite number of events, at least one type of event $\mathcal{E}_i$ does not cease happening and thus, using this i in the third assumption of Theorem 32, $W(x(t, j))$ converges to zero, that is, the solution converges to $\mathcal{A}$. Thus, the set $\mathcal{A}$ is globally pre-asymptotically stable.

We refer the reader to [6] and [19] for examples that illustrate Theorem 32 in the context of multiple Lyapunov functions. An illustration is also given at the end of the next section.

HYBRID FEEDBACK CONTROL BASED ON LIMITED EVENTS

In this section, we illustrate how stability analysis based on limited events motivates hybrid control algorithms for the nonlinear system (5). The control algorithms have a flow set C_c, flow map f_c, jump set D_c, jump map G_c, and feedback law κ_c, resulting in a closed-loop hybrid system with data given in (6) and (7).

Supervising a Family of Hybrid Controllers

The literature contains an extensive list of papers on supervisory control. For example, in the context of discrete-event systems see [68], in the setting of adaptive control see [57], and for hybrid systems see [41].

We discuss the design of a supervisor for a family of hybrid controllers $\widetilde{\mathcal{H}}_{c,q}$, where $q \in Q := \{1, \ldots, k\}$, each with state $\eta \in \mathbb{R}^m$ and data $(\widetilde{C}_q, \widetilde{f}_q, \widetilde{D}_q, \widetilde{G}_q, \widetilde{\kappa}_q)$. Classical controllers correspond to the special case where $\widetilde{C}_q = \mathbb{R}^{n+m}$, $\widetilde{D}_q = \emptyset$, and $\widetilde{G}_q(x) = \emptyset$ for all $x \in \mathbb{R}^{n+m}$.

Following [83], a supervisor $\mathcal{H}_c$ of these hybrid controllers is specified by indicating closed sets $C_q \subset \widetilde{C}_q$ and $D_q \subset \widetilde{D}_q$ in which flowing and jumping, respectively, using controller q is allowed, and a set $H_q \subset \mathbb{R}^{n+m}$, in which switching from controller q to another controller is allowed. In addition to the sets C_q, D_q, and H_q, the supervisor specifies a rule for how η and q change when the supervisor switches authority from controller q to a different controller. This rule is given through the set-valued mappings $J_q : \mathbb{R}^n \times \mathbb{R}^m \rightrightarrows \mathbb{R}^{m+1}$, $q \in Q$. Without loss of generality, we take $J_q(x_p, \eta) = \emptyset$ for all $(x_p, \eta) \notin H_q$. We also define f_q and κ_q to be the restrictions of $\widetilde{f}_q$ and $\widetilde{\kappa}_q$ to C_q, and we define G_q to be the set-valued mapping that is equal to $\widetilde{G}_q$ on $\overline{D_q \setminus H_q}$ and is empty otherwise.

The hybrid supervisor $\mathcal{H}_c$ has state $x_c = (\eta, q)$, input $x_p \in \mathbb{R}^n$, and is defined by the data

$$\begin{aligned} f_c(x_p, x_c) &= \begin{bmatrix} f_q(x_p, \eta) \\ 0 \end{bmatrix}, \; C_c = \bigcup_{q \in Q}(C_q \times \{q\}), \\ G_c(x_p, x_c) &= \begin{bmatrix} G_q(x_p, \eta) \\ q \end{bmatrix} \bigcup J_q(x_p, \eta), \\ D_c &= \bigcup_{q \in Q}((D_q \bigcup H_q) \times \{q\}), \end{aligned} \tag{40}$$

and $\kappa_c(x_p, x_c) = \kappa_q(x_p, \eta)$.

The goal of the supervisor is to achieve global asymptotic stability of the set $\mathcal{A} \times Q$, where $\mathcal{A} \subset \mathbb{R}^{n+m}$ is compact, while satisfying $C_c \bigcup D_c = \Theta \times Q$, where $\Theta \subset \mathbb{R}^{n+m}$ is closed, and to have that all maximal solutions starting in $C_c \bigcup D_c$ are complete. To guarantee that a supervisor can be constructed to achieve the goal, as in [83] we assume the following.

(Supervisor Assumption)

There exist closed sets $C_q \subset \widetilde{C}_q \cap \Theta$ and $D_q \subset \widetilde{D}_q \cap \Theta$ so that the following conditions hold:

1) There exist closed sets $\Psi_q \subset C_q \bigcup D_q$ such that $\bigcup_{q \in Q} \Psi_q = \Theta$.
2) For each $q \in Q$, the interconnection of the plant (5) and the controller $\mathcal{H}_{c,q} = (C_q, f_q, D_q, G_q, \kappa_q)$, having closed-loop data given in (6), (7), is such that, using the definition $\Phi_q := \bigcup_{\{i \in Q,\, i > q\}} \Psi_i$, the following conditions are satisfied:
   a) The set $\mathcal{A}$ is globally pre-asymptotically stable.
   b) No solution starting in Ψ_q reaches $\overline{\Theta \setminus (C_q \bigcup D_q \bigcup \Phi_q)} \setminus \mathcal{A}$.
   c) Each maximal solution is either complete or ends in $\Phi_q \bigcup \overline{\Theta \setminus (C_q \bigcup D_q \bigcup \Phi_q)}$.

Condition 2c) is a type of local existence of solutions condition. It is assumed to guarantee that a supervisor can be constructed to achieve not only global pre-asymptotic stability but also that all maximal solutions are complete.

The combination of conditions 2b) and 2c) imply that solutions starting in Ψ_q are either complete or end in $\mathcal{A}$ or Φ_q. This property enables building a hybrid supervisor to guarantee that each solution experiences no more than $k + 1$ events, where events are defined to be jumps in the value of q when $(x_p, \eta) \notin \mathcal{A}$.

The sets involved in the "Supervisor Assumption" for the problem of stabilizing the inverted position of a pendulum on a cart are depicted in Figure 20.

Given the sets C_q, D_q, Ψ_q, and Φ_q from the "Supervisor Assumption," we define the sets H_q and mappings J_q in (40) to be

$$H_q = \Phi_q \bigcup \overline{\Theta \setminus (C_q \bigcup D_q \bigcup \Phi_q)} \tag{41}$$

and in (42), shown at the bottom of the page. By construction, the mapping J_q is outer semicontinuous.

For the closed-loop system, events are defined to be jumps from $D \setminus (\mathcal{A} \times Q)$ that change the value of q. In other words, events are jumps that are due to J_q. Due to the definition of the map J_q, if a solution experiences one event then there is a time where the state (x_p, η, q) satisfies $(x_p, \eta) \in \Psi_q$. Thereafter, due to the "Supervisor Assumption," until the state reaches the set $\mathcal{A} \times Q$, the index q jumps monotonically. Therefore, the number of events experienced by a solution will be bounded by $k + 1$. Moreover, by the "Supervisor Assumption," the corresponding eventless system has the set $\mathcal{A} \times Q$ globally pre-asymptotically stable. These observations lead to the following corollary of Theorem 31.

Corollary 33

Under the "Supervisor Assumption" on the family of controllers $\mathcal{H}_{c,q}$, $q \in Q$, the closed-loop interconnection of the plant (5) and the hybrid supervisor $\mathcal{H}_c$, with data given in (40)–(42), satisfies the Basic Assumptions and has the compact set $\mathcal{A} \times Q$ globally asymptotically stable.

For the disk drive in Example 1 and for additional examples reported below, the "Supervisor Assumption" holds and the supervisory control algorithm indicated in Corollary 33 is used.

Some asymptotically controllable nonlinear control systems cannot be robustly stabilized to a point using classical, time-invariant state feedback. Examples include systems that fail Brockett's necessary condition for time-invariant, continuous stabilization [7], like the nonholonomic mobile robot, and other systems that satisfy Brockett's condition, like the system known as Artstein's circles [2]. The topological obstruction to robust stabilization in Artstein's circles is the same one encountered when trying to globally stabilize a point on a circle and is related to the issues motivating the development of topological complexity in the context of motion planning in [21]. We now illustrate how hybrid feedback, and supervisory control in particular, makes robust stabilization possible.

$$J_q(x_p, \eta) := \begin{cases} \begin{bmatrix} \{x_c\} \\ \{i \in Q : (x_p, \eta) \in \Psi_i\} \end{bmatrix}, & (x_p, \eta) \in \overline{\Theta \setminus (C_q \bigcup D_q \bigcup \Phi_q)}, \\ \begin{bmatrix} \{x_c\} \\ \{i \in Q : i > q, \;\; (x_p, \eta) \in \Psi_i\} \end{bmatrix}, & (x_p, \eta) \in H_q \setminus (\overline{\Theta \setminus (C_q \bigcup D_q \bigcup \Phi_q)}). \end{cases} \tag{42}$$

Simulation in Matlab/Simulink

Numerous software tools have been developed to simulate dynamical systems, including Matlab/Simulink, Modelica [S61], Ptolemy [S63], Charon [S60], HYSDEL [S66], and HyVisual [S62]. Hybrid systems $\mathcal{H} = (C, f, D, g)$ can be simulated in Matlab/Simulink with an implementation such as the one shown in Figure S12. This implementation uses the reset capabilities of integrator blocks in Matlab/Simulink.

Four basic blocks are used to input the data (C, f, D, g) of the hybrid system $\mathcal{H}$ for simulation by means of *Matlab function* blocks. The outputs of these blocks are connected to the input of the *Integrator System*, which integrates the flow map and executes the jumps. The *Integrator System* generates both the time variables t and j as well as the state trajectory x. Additionally, the *Integrator System* provides the value of the state before a jump occurs, which is denoted as x'. The four blocks are defined as follows:

- The f block uses the user function *f.m* to compute the flow map. Its input is given by the state of the system x, and its output is the value of the flow map f.
- The C block executes the user function *C.m*, which codes the flow set C. Its input is given by x'. The output of this block is set to one if x' belongs to the set C and is set to zero otherwise.
- The g block executes the user function *g.m*, which contains the jump map information. Its input is x', while its output is the value of the jump map g. Its output is passed through an IC block to specify the initial condition of the simulation.
- The D block evaluates the jump condition coded in the user function *D.m*. Its input is given by x', while its output is set to one if x' belongs to D, otherwise is set to zero.

For example, to simulate the bouncing ball system in Example 3 with gravity constant $\gamma = 9.8$, the flow map f and the flow set C can be coded into the functions *f.m* and *C.m*, respectively, as follows:

```
function out = f(u)
% state
x1 = u(1);
x2 = u(2);
% flow map
x1dot = x2;
x2dot = –9.8;
out = [x1dot; x2dot];                    (S9)
```

```
function out = C(u)
% state
x1 = u(1);
x2 = u(2);
% flow condition
if (x1 >= 0)
  out = 1;
else
  out = 0;
end                                      (S10)
```

The subsystems of the *Integrator System*, which are depicted in Figure S13, are used to compute the flows and jumps. These subsystems are the following.

The integrator block, labeled $1/s$, is the main block of the *Integrator System*. Its configuration uses the following settings: *external reset* set to "level hold" (setting for Matlab/Simulink R2007a and beyond; see [S65] for details about this setting for previous Matlab/Simulink versions), *initial condition source* set to "external" and *show state port* set to "checked." Simulink's zero-crossing detection algorithm for numerical computations is globally disabled.

The *CT dynamics* subsystem computes the flow map of the hybrid system as well as updates the parameters t and j during flows. Its output is integrated by the integrator block. This computation is accomplished by setting the dynamics of the integrator block to be $\dot{t} = 1$, $\dot{j} = 0$, $\dot{x} = f(x)$.

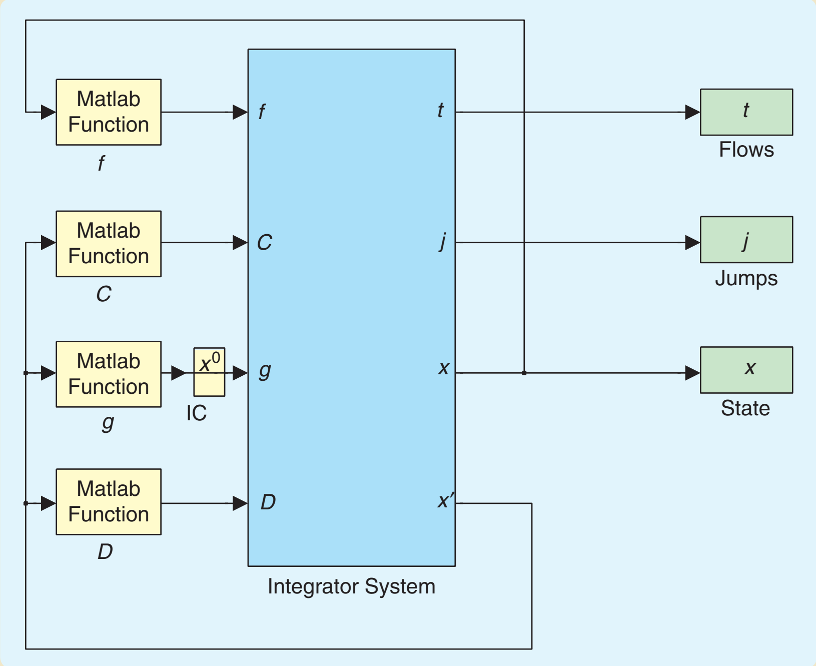

FIGURE S12 Matlab/Simulink implementation of a hybrid system $\mathcal{H} = (C, f, D, g)$. From the output of the Matlab functions, the *Integrator System* takes the values of the flow map f and jump map g as well as the indications of whether or not the state belongs to the flow set C and jump set D. The output of the *Integrator System* consists of t, j, and $x(t, j)$ at every simulation time step. The output x', which corresponds to the value of the state x before the jump instant, is used in the computations. The initial condition for the simulation is specified by x^0 in the *IC* block.

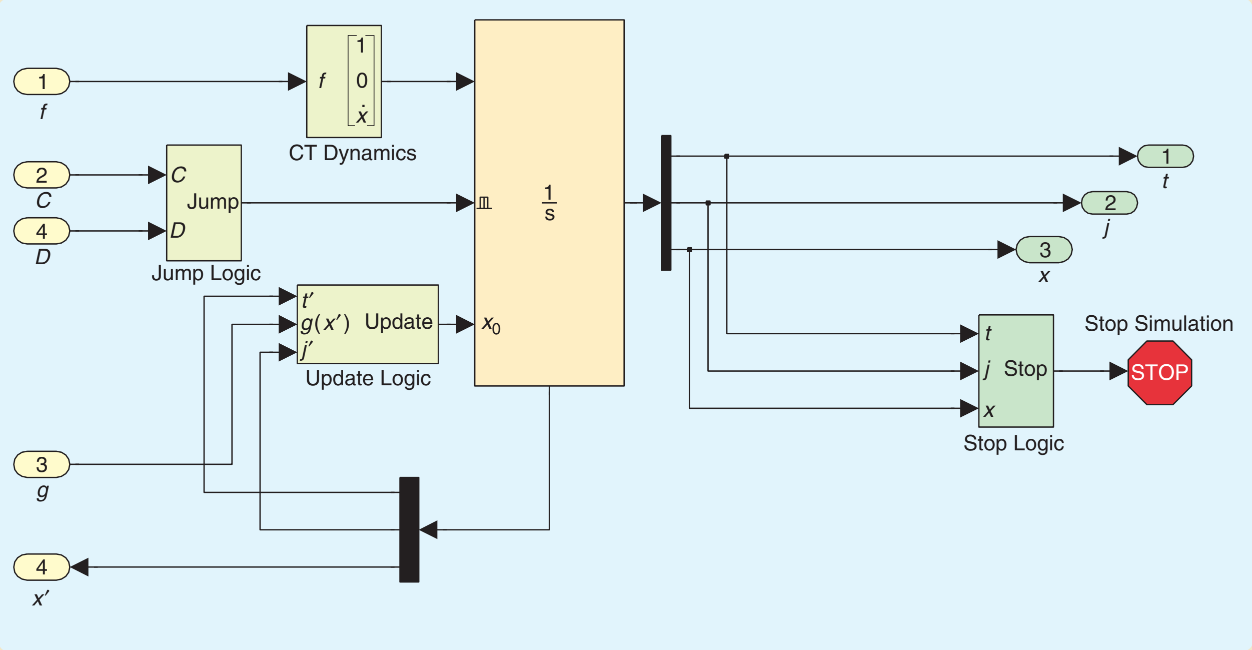

FIGURE S13 Internal diagram of the *Integrator System*. The CT *dynamics* subsystem computes the flow of the hybrid system and updates the parameter t. The *Jump Logic* subsystem triggers the jumps when the state x belongs to the jump set D. The *Update Logic* subsystem updates the value of the simulation as well as the pair (t, j) at jumps. The *Stop Simulation* subsystem stops the simulation when either the flow time is larger than or equal to the maximum flow time specified, the jump time is larger than or equal to the maximum number of jumps specified, or the simulation is in neither C nor D.

The *Jump Logic* subsystem triggers the jumps. Its inputs, which are the outputs of the blocks C and D, indicate whether the simulation is in either the flow set or the jump set. With this information, the *Jump Logic* subsystem generates the reset input of the integrator block as follows: when the simulation belongs only to the flow set, no reset is generated; when the simulation belongs only to the jump set, a reset is generated; and when the simulation belongs to both sets, a reset is generated depending on the value of a flag, which is denoted as *priority*.

The *Update Logic* subsystem updates the value of the simulation as well as the pair (t, j) at jumps. By using the state port of the integrator block, which reports the value of the state of the integrator at the exact instant that the reset condition becomes true, the value of the simulation is updated by $g(x')$. The flow time t is kept constant at jumps, and j is incremented by one. That is, at jumps the update law is $t^+ = t'$, $j^+ = j' + 1$, $x^+ = g(x')$.

The *Stop Logic* subsystem stops the simulation under any of the following events:

- The flow time is larger than or equal to the maximum flow time, which is specified by the parameter T.
- The jump time is larger than or equal to the maximum number of jumps, which is specified by the parameter J.
- The simulation is neither in C nor in D.

Under any of these events, the output of the logic operator connected to the *Stop logic* becomes one and the simulation is stopped.

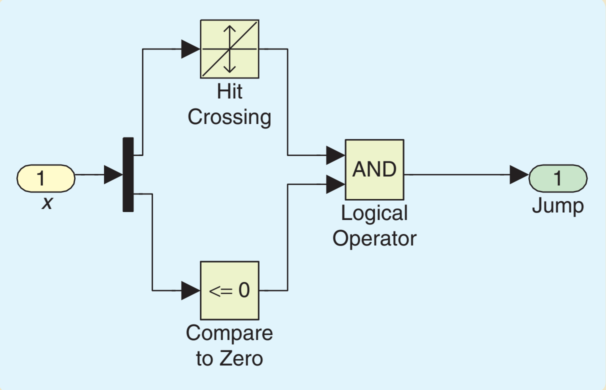

FIGURE S14 Alternative implementation in Matlab/Simulink of the jump set for the bouncing ball system in Example 3, which is given by $\{x : x_1 = 0, x_2 \le 0\}$. The "Hit Crossing" block detects when the state component x_1 crosses zero, while the "Compare to Zero" block is employed to report when the state component x_2 satisfies the condition $x_2 \le 0$. When the outputs of the "Hit Crossing" and "Compare to Zero" blocks are equal to one, a jump is triggered.

The following parameters are used in a simulation: initial condition for x, which is given by x^0; simulation horizon for the flows and jumps, which are given by T and J, respectively; and flag *priority* for flows/jumps, where $priority = 1$ indicates that jumps have the highest priority (forced jumps) and $priority = 2$ indicates

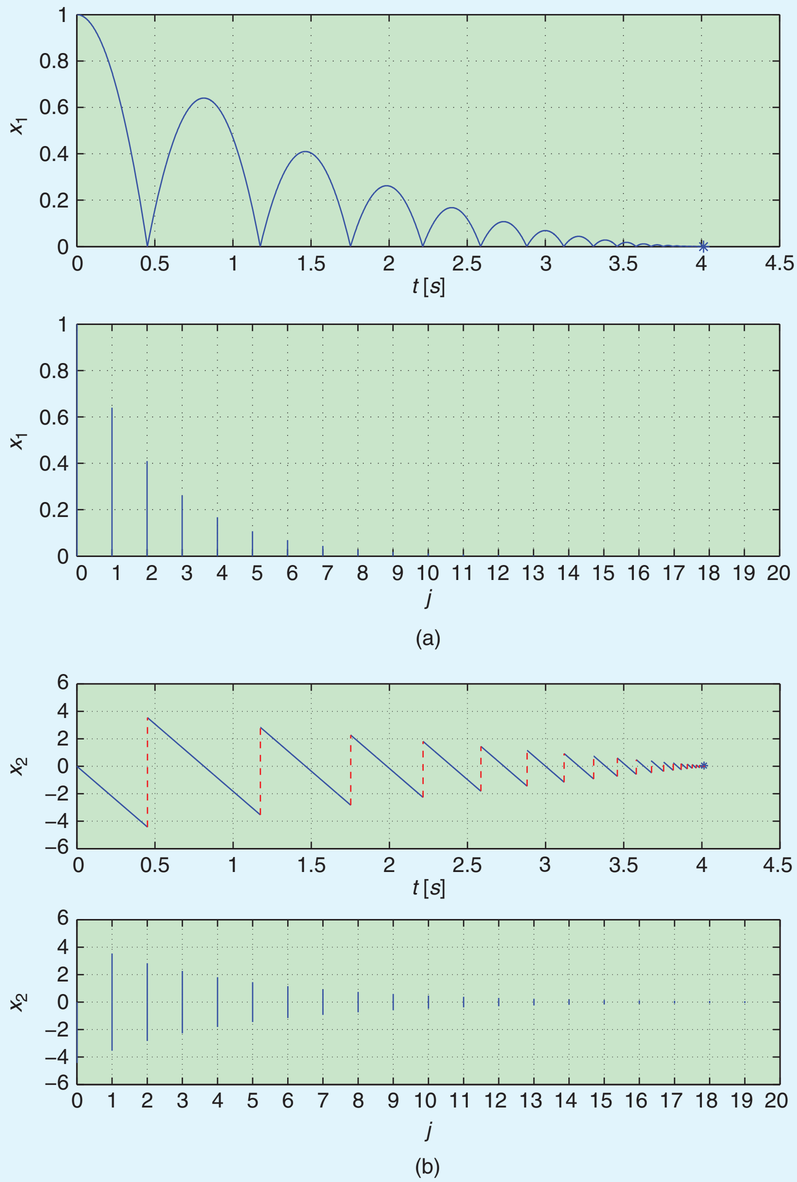

FIGURE S15 Simulation of the bouncing ball system in Matlab/Simulink for an initial condition with height $x_1 = 1$ and vertical velocity $x_2 = 0$. The maximum flow time is $T = 10$ s, the maximum number of jumps is $J = 20$, and the flag *priority* is set to one, which indicates that priority is given to the jump condition coded in *D.m.* (a) shows the height of the ball, which is denoted by x_1, while (b) shows the velocity of the ball, which is denoted by x_2, projected to the t axis and to the j axis of its hybrid time domain. The plots indicate that the solution jumps more frequently as the height of the ball approaches zero, which occurs when the flow time approaches the Zeno time of the bouncing ball system. For the given initial condition and parameters, the Zeno time is given by $t_Z = 4.066$ s.

that flows have the highest priority (forced flows). Matlab/Simulink documentation recommends using numerical solvers with variable step size for simulations of systems with discontinuities/jumps, such as ode45, which is a one-step solver based on an explicit Runge-Kutta (4, 5) formula. This setting, which is the default, is recommended as a first-try solver [S64].

Additionally, this implementation of a simulation of a hybrid system (C, f, D, g) permits replacing the Matlab functions that define the data of the hybrid system by a Simulink subsystem, and, hence, it enables using general purposes blocks in Simulink's library. For instance, this implementation permits the use of Simulink blocks to detect signal crossings of a given threshold with the "Hit Crossing" block, as illustrated for the bouncing ball example below, and quantizing a signal with the "Quantizer" block.

To simulate the bouncing ball system in Example 3 in Matlab/Simulink using the implementation in Figure S12, we employ the functions *f.m* and *C.m* defined in (S9) and (S10) for the flow map and flow set, respectively. The jump map and jump set are implemented with restitution coefficient $\rho = 0.8$ in the functions *g.m* and *D.m,* respectively, as follows:

```
function out = g(u)
% state
x1 = u(1);
x2 = u(2);
% jump map
x1plus = –0.8*x1;
x2plus = –0.8*x2;
out = [x1plus; x2plus];

function out = D(u)
% state
x1 = u(1);
x2 = u(2);
% jump condition
if (x1 <= 0 && x2 <= 0)
```

Example 34: Robust Global Asymptotic Stabilization of a Point on a Circle

Consider a plant with state ξ that is constrained to evolve on the unit circle with angular velocity u, where positive angular velocity corresponds to counterclockwise rotation. Moving on the unit circle constrains the velocity vector to be perpendicular to the position vector, as generated by rotating the position vector using the

```
    out = 1;
else
    out = 0;
end
```

Note that due to numerical integration error, the condition $x_1 = 0$ may never be satisfied. Then, solutions obtained from simulations can miss jumps as well as leave the set $C \cup D$, causing a premature stop of the simulation. To overcome these issues, instead of using the jump set in Example 4, which is given by $\{x : x_1 = 0, x_2 \leq 0\}$, the jump set $D = \{x : x_1 \leq 0, x_2 \leq 0\}$ is used. This jump set has no effect on solutions starting from the set $\{x : x_1 \geq 0\}$. Another alternative is to replace the function *D.m* by a Simulink subsystem implementing the condition $x_1 = 0$ in the jump set $\{x : x_1 = 0, x_2 \leq 0\}$ with a Simulink block that is capable of detecting zero crossings of x_1 as shown in Figure S14. This implementation uses three Simulink blocks, namely, a “Hit Crossing” block, a “Compare to Zero” block, and an “AND Logical Operator” block. The last block combines the outputs of the first two blocks. The “Hit Crossing” block is used to detect when the state component x_1 crosses zero. For this purpose, this block is required to have enabled Simulink’s zero-crossing detection algorithm for numerical computations. Then, when x_1 crosses zero, this algorithm sets back the simulation step to a time instant close enough to the time at which x_1 is zero, recomputes the state, and sets its output to one. The “Compare to Zero” block is used to report when the state component x_2 satisfies the condition $x_2 \leq 0$ by setting its output to one. Then, when the outputs of the “Hit Crossing” and “Compare to Zero” blocks are equal to one, which indicates that $x_1 = 0$ and $x_2 \leq 0$ are satisfied, a jump is reported to the *Integrator System.*

Figure S15 depicts the height and velocity of a solution to the bouncing ball system projected to the t axis and j axis of its hybrid time domain. The solution is generated from the initial condition $(1, 0)$ with maximum flow time $T = 10$ s, maximum number of jumps $J = 20$, and highest priority for jumps, that is, $priority = 1$. It is easy to verify that the solutions to the bouncing ball system are Zeno and that the Zeno time, which is given by $t_Z = \sup\{t : (t, j) \in \operatorname{dom} x\}$, is given by

$$t_Z = \frac{x_2(0,0) + \sqrt{x_2(0,0)^2 + 2\gamma x_1(0,0)}}{\gamma} + \frac{2\rho\sqrt{x_2(0,0)^2 + 2\gamma x_1(0,0)}}{\gamma(1-\rho)}.$$

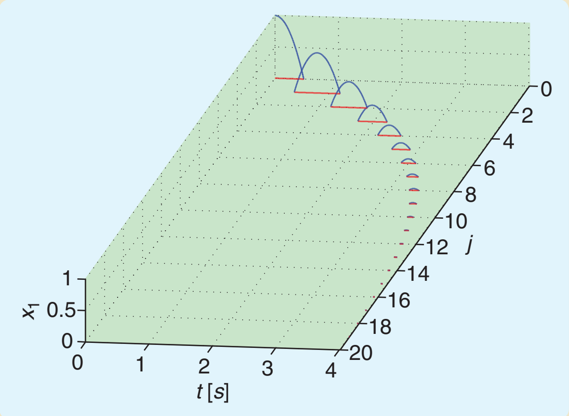

FIGURE S16 Hybrid arc and hybrid time domain corresponding to the height component of the simulation to the bouncing ball system in Matlab/Simulink depicted in Figure S15. The hybrid arc is shown in blue, while the hybrid time domain is shown in red.

For the initial conditions $x_1(0,0) = 1$, $x_2(0,0) = 0$ and the parameters $\gamma = 9.8\ \mathrm{m/s}^2$ and $\rho = 0.8$, the simulation yields $t_Z = 4.066$ *s*. The solution obtained from the numerical simulation indicates that as the flow time approaches t_Z , the height and velocity of the ball as well as the time elapsed between jumps approach zero. Finally, Figure S16 depicts the hybrid arc and hybrid time domain for the height component of the solution obtained from the numerical simulation.

The Matlab/Simulink files for this implementation and the functions used for plotting solutions as well as several examples are available online [S65].

REFERENCES

[S60] R. Alur, T. Dang, J. Esposito, Y. Hur, F. Ivancic, V. Kumar, I. Lee, P. Mishra, G. J. Pappas, and O. Sokolsky, “Hierarchical modeling and analysis of embedded systems,” *Proc. IEEE*, vol. 91, pp. 11–28, 2003.

[S61] H. Elmqvist, S. E. Mattsson, and M. Otter, “Modelica: The new object-oriented modeling language,” in *Proc. 12th European Simulation Multiconf.*, Manchester, U.K., 1998.

[S62] E. A. Lee and H. Zheng, “Operational semantics for hybrid systems,” in *Proc. 8th Int. Workshop Hybrid Systems: Computation and Control*, Zurich, Switzerland, 2005, pp. 25–53.

[S63] J. Liu and E. A. Lee, “A component-based approach to modeling and simulating mixed-signal and hybrid systems,” *ACM Trans. Model. Comput. Simul.*, vol. 12, no. 4, pp. 343–368, 2002.

[S64] The MathWorks. (2008). Matlab Simulink R2008 HelpDesk [Online]. Available: http://www.mathworks.com/access/helpdesk/help/toolbox/simulink/simulink.shtml

[S65] R. G. Sanfelice. (2008). Simulating hybrid systems in Matlab/Simulink [Online]. Available: http://www.u.arizona.edu/,sricardo/software.html

[S66] F. D. Torrisi and A. Bemporad, “HYSDEL—A tool for generating computational hybrid models for analysis and synthesis problems,” *IEEE Trans. Contr. Syst. Technol.*, vol. 12, pp. 235–249, 2004.

rotation matrix $R(-\pi/2)$. The constrained equations of motion are

$$\dot{\xi} = uR(-\pi/2)\xi \quad \xi \in \mathbb{S}^1.$$

For this system, global asymptotic stabilization of the point $(1, 0)$ with a continuous state-feedback law is impossible. Global asymptotic stabilization of this point by discontinuous feedback is possible, although the resulting

With the use of hybrid time domains and the notion of graphical convergence, sequential compactness of the space of solutions and semicontinuous dependence of solutions on initial conditions and perturbations can be established under mild conditions.

global closed-loop behavior is not robust to arbitrarily small measurement noise. For an explanation, see "Robustness and Generalized Solutions." We now describe a hybrid controller that achieves robust, global asymptotic stabilization of the point $(1, 0)$.

Consider a hybrid supervisor for classical controllers $\widetilde{\mathcal{H}}_q$, $q \in Q = \{1, 2\}$, that are static and are given by $\widetilde{\kappa}_q(\xi) = \xi_q$. The "Supervisor Assumption" is satisfied for $\mathcal{A} = (1, 0)$ and $\Theta = \mathbb{S}^1$ by taking $D_1 = D_2 = \varnothing$ and

$$\begin{aligned}\Psi_1 = C_1 &:= \{\xi \in \mathbb{S}^1 : \xi_1 \leq -1/3\},\\ C_2 &:= \{\xi \in \mathbb{S}^1 : \xi_1 \geq -2/3\},\\ \Psi_2 &:= \{\xi \in \mathbb{S}^1 : \xi_1 \geq -1/3\}.\end{aligned}$$

Following the prescription of the components of a hybrid supervisor in (41) and (42), we get $H_1 = \Psi_2$, $H_2 = \overline{\mathbb{S}^1 \setminus C_2} = \{\xi \in \mathbb{S}^1 : \xi_1 \leq -2/3\}$ and $J_q = 3 - q$. Then, using the definition of a supervisory controller in (40), we get a hybrid controller with state $x_c = q$, data $C_c = (C_1 \times \{1\}) \cup (C_2 \times \{2\})$, $f_c = 0$, $D_c = (H_1 \times \{1\}) \cup (H_2 \times \{2\})$, $G_c = 3 - q$, and $\kappa_c(\xi, q) = \xi_q$. See Figure 18.

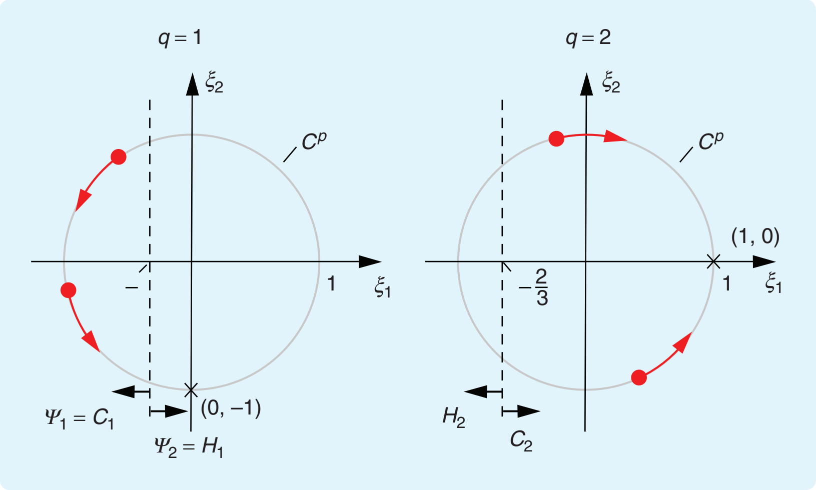

FIGURE 18 Illustration of the hybrid controller, for robust global stabilization of the point $(1, 0)$ on a circle $\mathbb{S}^1$, in Example 34. The sets C_1, Ψ_1, H_1 and C_2, Ψ_2, H_2 are subsets of the circle, contained in the half-planes as indicated by the arrows. These sets define the flow and jump sets in modes $q = 1$ and $q = 2$, respectively. In mode $q = 1$, the controller drives the state toward the point $(0, -1)$ from every point in C_1, as indicated by the arrows, and the switch to mode $q = 2$ happens when the state is in H_1. In mode $q = 2$, the controller drives the state toward the point $(1, 0)$ from every point in C_2, as indicated by the arrows, and the switch to mode $q = 1$ happens when the state is in H_2. This controller results in robust asymptotic stability of $(1, 0)$ on the circle. Note that the solutions starting from $\xi_1 = -1/3$, $\xi_2 > 0$, $q = 1$ are not unique. This initialization results in one solution that jumps immediately to $q = 2$ and then flows to $(1, 0)$ and another solution that flows to $\xi_1 = -1/3$, $\xi_2 < 0$, $q = 1$ before jumping. Similarly, the solutions starting from $\xi_1 = -2/3$, $\xi_2 > 0$, $q = 2$ are not unique.

According to Corollary 33, this controller renders the set $\{(1, 0)\} \times Q$ globally asymptotically stable. In fact, it can be established that the point $(1, 0, 2)$ is globally asymptotically stable. ■

Example 35: Mobile Robot Control

Consider a stabilization problem for the control system

$$\left.\begin{aligned}\dot{z} &= \xi v\\ \dot{\xi} &= uR(-\pi/2)\xi\end{aligned}\right\} \ (z, \xi) \in \mathbb{R}^2 \times \mathbb{S}^1,$$

where $z \in \mathbb{R}^2$ corresponds to the position of a nonholonomic vehicle, $v \in \mathbb{R}$ denotes its velocity, which is treated as a control variable, $\xi \in \mathbb{S}^1$ denotes the orientation of the vehicle, and $u \in \mathbb{R}$ denotes angular velocity, which is the other control variable. The plant state is $x_p = (z, \xi)$. As in Example 34, positive angular velocity corresponds to rotation of ξ in the counterclockwise direction. The control objective is to asymptotically stabilize the point $x_p^* = (0, \xi^*)$, where $\xi^* \in \mathbb{S}^1$. We start by picking $v = -\rho(z^\top \xi)$, where ρ is a continuous function satisfying $s\rho(s) > 0$ for all $s \neq 0$. A discontinuous or set-valued choice for ρ is also possible. The condition $s\rho(s) > 0$ for all $s \neq 0$ ensures that, during flows, the size of z decreases except when $z^\top \xi = 0$ at which points $\dot{z} = 0$.

With the vehicle velocity specified, what remains is a control system with a single input u. Following the ideas in Example 34, a hybrid feedback algorithm for u exists that drives ξ to a given point on the unit circle. More generally, a comparable hybrid algorithm can be developed to track any continuously differentiable function, of the plant state z and perhaps of extra controller variables, that takes values on the unit circle. This task is accomplished by adding to the point stabilization algorithm an appropriate feedforward signal that renders the desired angular profile invariant.

Following ideas that are similar to those in [65], we consider the supervision of two hybrid controllers for the mobile robot system using a hybrid supervisor $\mathcal{H}_c$ with data given in (40)–(42).

The first controller is efficient at bringing the position of the robot close to the desired target while the second controller works to simultaneously regulate orientation and position. We give the two controllers a common state $\eta \in \mathbb{R}^m$ and suppose η takes values in a compact set Ω. The first hybrid controller drives ξ to $-z/|z|$. The data of the first controller satisfies

$$\widetilde{C}_1 \cup \widetilde{D}_1 = (\mathbb{R}^2 \setminus \{0\}) \times \mathbb{S}^1 \times \Omega.$$

This specification avoids the singularity in $-z/|z|$ at $z = 0$.

The second hybrid controller drives ξ to $R(\sigma(z)\chi_1)\xi^*$, where σ is a continuously differentiable function that is zero at zero and positive otherwise, and χ_1 is a state of the controller satisfying

$$\dot{\chi} = \omega R(\pi/2)\chi, \quad \chi \in \mathbb{S}^1,$$

where $\omega > 0$. This controller satisfies $\widetilde{C}_2 \cup \widetilde{D}_2 = \mathbb{R}^2 \times \mathbb{S}^1 \times \Omega$. This second controller is closely related to time-varying stabilization algorithms that have appeared previously in the literature. See, for example, [64] and [71].

To satisfy the "Supervisor Assumption" for $\mathcal{A} = \{0\} \times \{\xi^*\} \times \Omega$ and $\Theta = \mathbb{R}^2 \times \mathbb{S}^1 \times \Omega$, we let $0 < \varepsilon_1 < \varepsilon_2$ and take

$$\begin{aligned} C_1 &:= \widetilde{C}_1 \cap ((\overline{\mathbb{R}^2 \setminus \varepsilon_1\mathbb{B}}) \times \mathbb{S}^1 \times \Omega), \\ D_1 &:= \widetilde{D}_1 \cap ((\overline{\mathbb{R}^2 \setminus \varepsilon_1\mathbb{B}}) \times \mathbb{S}^1 \times \Omega), \\ \Psi_1 &:= C_1 \cup D_1 = (\overline{\mathbb{R}^2 \setminus \varepsilon_1\mathbb{B}}) \times \mathbb{S}^1 \times \Omega, \end{aligned}$$

and

$$\begin{aligned} C &:= \widetilde{C}_2 \cap (\varepsilon_2\mathbb{B} \times \mathbb{S}^1 \times \Omega), \\ D_2 &:= \widetilde{D}_2 \cap (\varepsilon_2\mathbb{B} \times \mathbb{S}^1 \times \Omega), \\ \Psi_2 &:= \varepsilon_1\mathbb{B} \times \mathbb{S}^1 \times \Omega. \end{aligned}$$

The components of the supervisor are then given by

$$\begin{aligned} H_1 &= \Psi_2 = \varepsilon_1\mathbb{B} \times \mathbb{S}^1 \times \Omega, \\ H_2 &= \overline{(\mathbb{R}^2 \setminus \varepsilon_2\mathbb{B})} \times \mathbb{S}^1 \times \Omega, \\ J_q &= 3 - q. \end{aligned}$$

With these specifications, according to Corollary 33 the hybrid controller (40)–(42) renders the set $\mathcal{A} \times Q$ globally asymptotically stable. ■

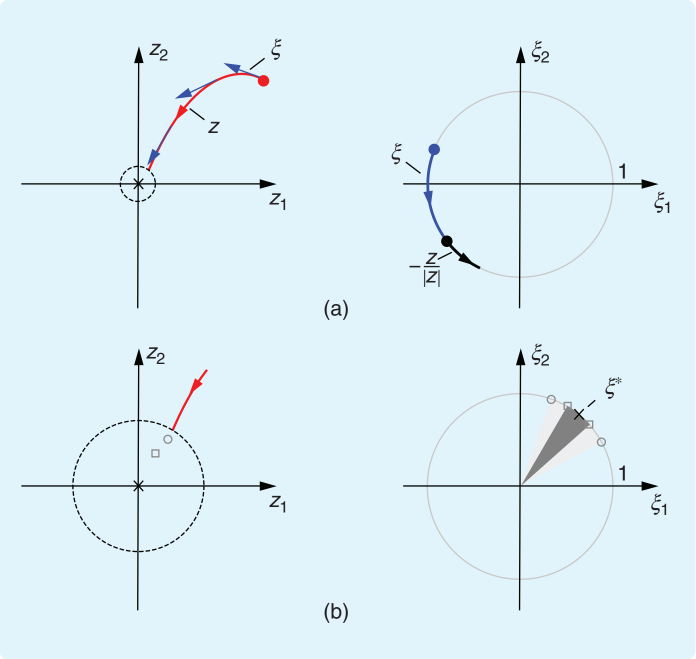

FIGURE 19 Position $z = (z_1, z_2)$ and orientation $\xi = (\xi_1, \xi_2)$ of the mobile robot with hybrid controller in Example 35. (a) The position, orientation, and ratio $-z/|z|$ with the first controller are shown on the z and ξ planes. The position z is in red, the orientation ξ is in blue, and $-z/|z|$ is in black. The first controller drives z to a neighborhood of $z = 0$ and ξ to $-z/|z|$. (b) The second controller drives the orientation to a cone centered at ξ^* with aperture depending on z. Cones associated to two points in a neighborhood of $z = 0$ are depicted. The two points, indicated by a square and a circle, along with the neighborhood around $z = 0$, indicated with dashed line, are depicted in the z plane, while their associated cones are denoted on the ξ plane. As z approaches the origin, the aperture of the cone approaches zero.

Example 36: Global Asymptotic Stabilization of the Inverted Position for a Pendulum on a Cart

Consider the task of robustly, globally asymptotically stabilizing the point $x_p^* := (0, 1, 0)$ for the system with state $x_p = (\xi, z) \in \mathbb{R}^3$ and dynamics

$$\begin{bmatrix} \dot{\xi} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} zR(-\pi/2)\xi \\ \xi_1 + \xi_2 u \end{bmatrix} =: f_p(x_p, u)$$
$$x_p \in \mathbb{S}^1 \times \mathbb{R}.$$

This model corresponds to a pendulum on a cart. The model includes an input transformation from force to acceleration of the cart, which is the control input u in the system above. Moreover, the ratio of the gravitational constant to the pendulum length has been normalized to one. The cart dynamics are ignored to simplify the presentation; however, global asymptotic stabilization of the full cart-pendulum system can be addressed with the same tools used below. The state ξ denotes the angle of the pendulum. The point $\xi = (0, 1)$ corresponds to the inverted position while $\xi = (0, -1)$ corresponds to the down position. The state z corresponds to the angular velocity, with positive velocity in the clockwise direction.

A feature of hybrid systems satisfying the Basic Assumptions is that pre-asymptotic stability is robust.

We construct a hybrid feedback stabilizer that supervises three classical controllers. The first controller steers the state x_p out of a neighborhood of the point $-x_p^*$. The second controller moves the system to a neighborhood of the point x_p^*. The third controller locally asymptotically stabilizes the point x_p^*. For an illustration of the data of these controllers, see Figure 20.

The third controller, a local asymptotic stabilizer for x_p^*, is simple to synthesize. For example, the idea of partial feedback linearization with "output" ξ_1 can be used since ξ_2 is positive and bounded away from zero in a neighborhood of x_p^*. Let $\kappa_3 : \mathbb{S}^1 \times \mathbb{R} \to \mathbb{R}$ denote this local asymptotic stabilizer, let C_3 be a compact neighborhood of the point x_p^* that is also a subset of the basin of attraction for x_p^* for the system $\dot{x}_p = f_p(x_p, \kappa_3(x_p))$, $x_p \in \mathbb{S}^1 \times \mathbb{R}$, and let Ψ_3 be a compact neighborhood of the point x_p^* with the property that solutions of $\dot{x}_p = f_p(x_p, \kappa_3(x_p))$ starting in Ψ_3 do not reach the boundary of C_3. Then, redefine C_3 and Ψ_3 by intersecting the original choices with $\mathbb{S}^1 \times \mathbb{R}$. The set Ψ_3 is indicated in green in Figure 20(d) while the set C_3 is the union of the green and yellow regions in the same figure.

For the second controller, let $0 < \delta < \varepsilon < 1$ and define

$$\begin{aligned} W(x_p) &:= \frac{1}{2}z^2 + 1 + \xi_2, \\ \Psi_2 &:= \overline{\{(\xi, z) \in \mathbb{S}^1 \times \mathbb{R} : W(x_p) \geq \varepsilon\} \setminus \Psi_3}, \\ C_2 &:= \overline{\{(\xi, z) \in \mathbb{S}^1 \times \mathbb{R} : W(x_p) \geq \delta\} \setminus \Psi_3}, \\ \kappa_2(x_p) &:= -z\xi_2(W(x_p) - 2) \quad \text{for all } x_p \in C_2. \end{aligned}$$

The set Ψ_2 is indicated in green in Figure 20(b) or, alternatively, in Figure 20(c). The set C_2 is the union of the green and yellow regions in the same figures.

For the first controller, define $C_1 := \overline{(\mathbb{S}^1 \times \mathbb{R}) \setminus (\Psi_2 \cup \Psi_3)}$, $\Psi_1 := C_1$, and $\kappa_1(x_p) := k$ for all $x_p \in C_1$, where $k > \sqrt{\delta(2-\delta)}/(1-\delta)$. The set Ψ_1 is indicated in green in Figure 20(a).

We now establish that the indicated C_q and Ψ_q satisfy the "Supervisor Assumption" for $\mathcal{A} := \{x_p^*\}, Q := \{1, 2, 3\}$, and $\Theta := \mathbb{S}^1 \times \mathbb{R}$. We take $\tilde{C}_q = \mathbb{S}^1 \times \mathbb{R}$ and $\tilde{D}_q = \varnothing$ for each $q \in Q$. First, by construction, $\Psi_q \subset C_q$ for each $q \in Q$. Also,

$$\begin{aligned} \Psi_1 \cup \Psi_2 \cup \Psi_3 &= \overline{(\mathbb{S}^1 \times \mathbb{R}) \setminus (\Psi_2 \cup \Psi_3)} \cup \Psi_2 \cup \Psi_3 \\ &= \mathbb{S}^1 \times \mathbb{R} = \Theta. \end{aligned}$$

Next, we check that $\mathcal{A}$ is globally pre-asymptotically stable for $\dot{x}_p = f_p(x_p, \kappa_q(x_p))$, $x_p \in C_q$ for each $q \in Q$. This property holds for $q = 3$ since C_3 is a subset of the basin of attraction for x_p^* for the system $\dot{x}_p = f_p(x_p, \kappa_3(x_p))$.

For $q = 1$, we note that $x_p \in C_1$ implies that $W(x_p) \leq \delta$. In particular, $|z| \leq \sqrt{2\delta} < \sqrt{2}$ and $\xi_2 \leq \delta - 1 < 0$. We use the Lyapunov function $V_1(x_p) := 2 + z$, which is positive on C_1, and we obtain

$$\begin{aligned} \langle \nabla V_1(x_p), f_p(x_p, \kappa_1(x_p)) \rangle &= \xi_1 + \xi_2 k \\ &\leq \sqrt{1 - \xi_2^2} - |\xi_2| k \\ &\leq \sqrt{\delta(2-\delta)} - (1-\delta)k \\ &< 0 \end{aligned}$$

for all $x_p \in C_q$. It follows from Theorem 20 that $\mathcal{A}$ is globally pre-asymptotically stable for $\dot{x}_p = f_p(x_p, \kappa_1(x_p))$, $x_p \in C_1$. This conclusion is equivalent to saying that there are no complete solutions that remain in the green region in Figure 20(a).

For $q = 2$, we use the Lyapunov function $V_2(x_p) := 1 + (W(x_p) - 2)^2$, which is positive on C_2. For all $x_p \in C_2$,

$$\langle \nabla V_2(x_p), f_p(x_p, \kappa_2(x_p)) \rangle = -(z\xi_2)^2(W(x_p) - 2)^2 \leq 0.$$

It follows from Theorem 23 that $\mathcal{A}$ is globally pre-asymptotically stable for $\dot{x}_p = f_p(x_p, \kappa_2(x_p))$, $x_p \in C_2$. This is seen by noting that $V_2(x_p) = \mu \geq 1$ implies

$$\langle \nabla V_2(x_p), f_p(x_p, \kappa_2(x_p)) \rangle = -(z\xi_2)^2(\mu - 1).$$

For $\mu > 1$, nontrivial solutions evolving in the set where $\langle \nabla V_2(x_p), f_p(x_p, \kappa_2(x_p)) \rangle = 0$ must either be the constant solution $x_p(t, j) = x_p^*$ or the constant solution $x_p(t, j) = -x_p^*$. Since neither x_p^* nor $-x_p^*$ belong to C_2, the constrained system admits no nontrivial solutions. When $\mu = 1$, solutions evolve in the set where $W(x_p) = 2$. Belonging to this set and also to the set C_2 implies that z is bounded away from zero. Otherwise, the state would be near x_p^*, which is not a part of C_2. When z is bounded away from zero, ξ must eventually approach the point $(0, 1)$. Then, because we are considering solutions satisfying the condition $W(x_p) = 2$, z must eventually approach zero as ξ approaches $(0, 1)$. In other words, x_p must eventually approach x_p^*. Since x_p^* does not belong to C_2, there are no complete solutions for the constrained system.

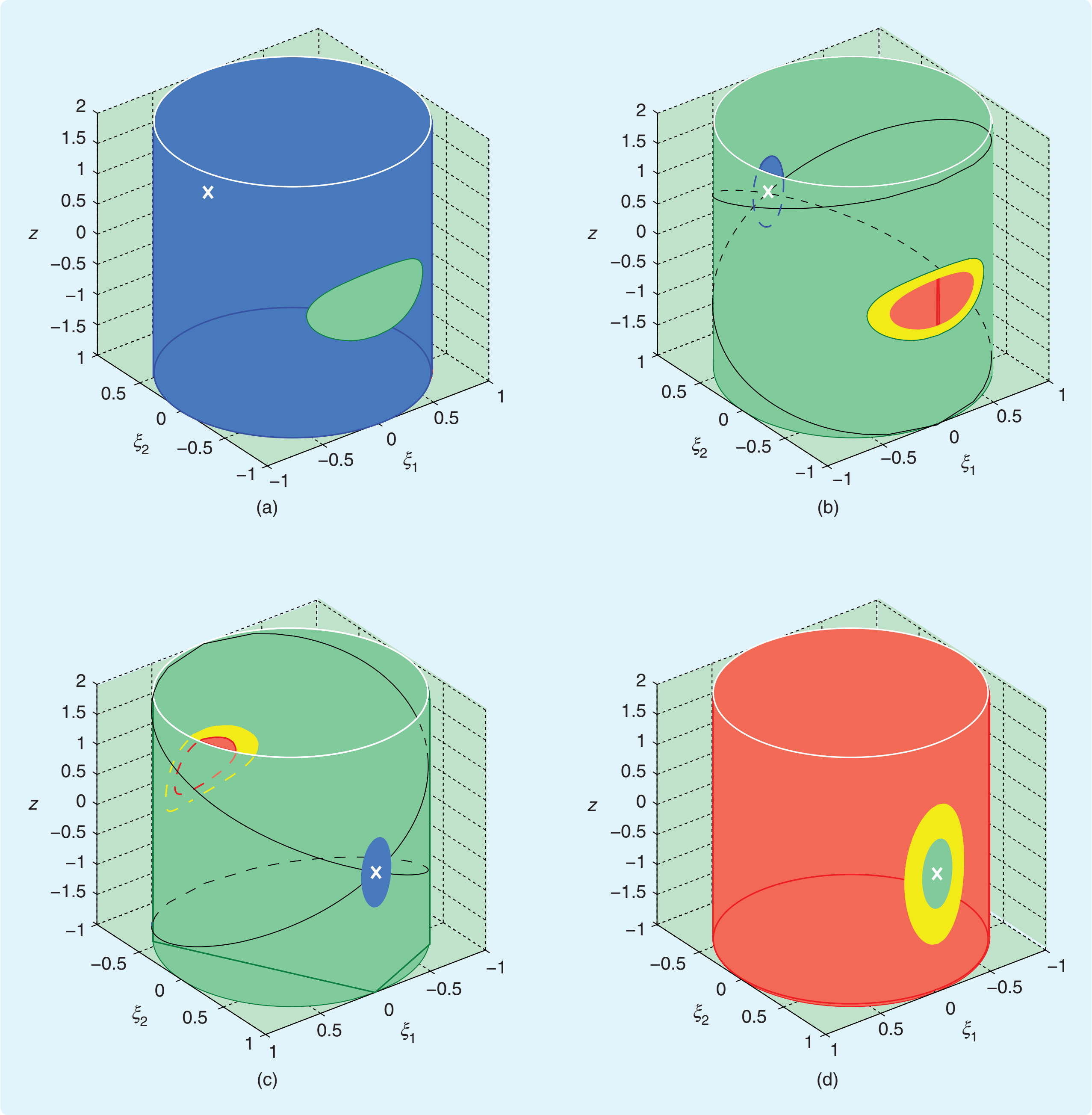

FIGURE 20 The sets used in the construction of the hybrid controller in Example 36 for $q = 1$, $q = 2$, and $q = 3$ are shown in (a), (b), and (d), respectively. In (c), the sets for $q = 2$ are depicted with perspective rotated by 180°. The state $x = (\xi, z)$ evolves on the cylinder $\mathbb{S}^1 \times \mathbb{R} \subset \mathbb{R}^3$, while $q \in \{1, 2, 3\}$. The vertical position $(\xi, z) = ((0, 1), 0)$ is indicated by a black **x**. The black curve represents the set of points where $W(x) := 0.5z^2 + 1 + \xi_2 = 2$. To meet the "Supervisor Assumption," the sets D_q can be taken to be empty. The sets $\Phi_q := \bigcup_{i>q}\Psi_q$ are indicated in blue, the sets $\Psi_q \subset C_q$ are indicated in green, the sets $C_q \backslash \Psi_q$ are indicated in yellow, and the sets $\overline{(\mathbb{S}^1 \times \mathbb{R})\backslash(C_q \cup (\bigcup_{i>q}\Psi_i))}$ are indicated in red. The sets H_q constructed in the supervisory control algorithm are the unions of the blue and red regions. The key properties are that, for each $q \in \{1, 2, 3\}$, the vertical position is globally pre-asymptotically stable for the system $\dot{x}_p = f_p(x_p, \kappa_q(x_p))$, $x_p \in C_q$, and each solution starting in a green region does not reach a red region. Taken together, these assumptions guarantee that solutions starting in a green region reach a blue region, for $q \in \{1, 2\}$.

This deduction is equivalent to saying that there are no complete solutions that remain in the union of the green and yellow regions in Figure 20(b) or, alternatively, Figure 20(c). This behavior is because trajectories in those regions converge to the set where $W(x_p) = 2$, which is indicated by the black curve, and moving along this set leads to points near x_p^* where the flow $\dot{x}_p = f_p(x_p, \kappa_2(x_p))$ cannot be continued in the union of the green and yellow regions.

Pre-asymptotically stable compact sets always admit Lyapunov functions.

Next we verify that, for each $q \in Q$, each solution of $\dot{x}_p = f_p(x_p, \kappa_q(x_p))$, $x_p \in C_q$ is either complete or ends in $\Phi_q \bigcup \overline{(\mathbb{S}^1 \times \mathbb{R}) \setminus C_q}$, where $\Phi_q = \bigcup_{i \in Q,\, i>q} \Psi_i$. This condition follows immediately from the fact that the second condition in Proposition 2 holds for each $x_p \in C_q \setminus \overline{(\mathbb{S}^1 \times \mathbb{R}) \setminus C_q}$. Thus, maximal solutions that are not complete must end in $\overline{(\mathbb{S}^1 \times \mathbb{R}) \setminus C_q}$. With respect to Figure 20, this conclusion amounts to the statement that the only places in green or yellow regions where flowing solutions cannot be continued are on the boundaries that touch red or blue regions.

Finally, we verify that, for each $q \in Q$, no solution starting in Ψ_q reaches $\overline{(\mathbb{S}^1 \times \mathbb{R}) \setminus (C_q \bigcup \Phi_q)} \setminus \mathcal{A}$. In other words, in terms of Figure 20, no solution starting in a green region reaches a red region. For $q = 1$, the red set is empty and so there is nothing to check. For $q = 3$, the property follows from the definition of Ψ_3, which is taken sufficiently small to satisfy the condition. For $q = 2$, this property follows from the fact that $\langle \nabla W(x_p), f_p(x_p, \kappa_2(x_p)) \rangle \geq 0$ for x_p such that $\delta \leq W(x_p) \leq 2$ and the fact that $x_p \in \Psi_2$ implies $W(x_p) \geq \varepsilon > \delta$.

This establishes that the "Supervisor Assumption" holds. In turn, the hybrid supervisory control algorithm given in (40)–(42), which is expressed explicitly in terms of C_q, Ψ_q, and κ_q, robustly, globally asymptotically stabilizes the point x_p^* for the pendulum on a cart. ■

PATCHY CONTROL LYAPUNOV FUNCTIONS

Patchy control-Lyapunov functions (PCLFs) are defined in [24] as an extension of the now classical concept of a control-Lyapunov function (CLF) [77], [2]. A CLF for a given control system is a smooth function whose value decreases along the solutions to the system under an appropriate choice of controls. The existence of such a function guarantees, in most cases, the existence of a robust stabilizing feedback [48]. Unfortunately, some nonlinear control systems do not admit a CLF.

A PCLF for system (5) is, broadly speaking, an object consisting of several local CLFs, the domains of which cover $\mathbb{R}^n$ and have certain weak invariance properties. PCLFs exist for far broader classes of nonlinear systems than CLFs, especially when an infinite number of patches (that is, of local CLFs) is allowed. Moreover, PCLFs lead to a robustly stabilizing hybrid feedback.

A smooth PCLF for the nonlinear system (5) with respect to the compact set $\mathcal{A} \subset \mathbb{R}^n$ consists of a finite set $Q \subset \mathbb{Z}$ and a collection of functions V_q and sets Ω_q, Ω'_q for each $q \in Q$, such that:

i) $\{\Omega_q\}_{q \in Q}$ and $\{\Omega'_q\}_{q \in Q}$ are families of nonempty open subsets of $\mathbb{R}^n$ such that

$$\mathbb{R}^n = \bigcup_{q \in Q} \Omega_q = \bigcup_{q \in Q} \Omega'_q,$$

and, for each $q \in Q$, the unit outward normal vector to Ω_q is continuous on $\partial\Omega_q \setminus (\bigcup_{i>q} \Omega'_i \bigcup \mathcal{A})$, and

$$\overline{\Omega'}_q \setminus \mathcal{A} \subset \Omega_q.$$

ii) For each q, V_q is a smooth function defined on an open set containing $\overline{\Omega_q \setminus \bigcup_{i>q} \Omega'_i} \setminus \mathcal{A}$.

iii) For each $q \in Q$ there exist a continuous, positive definite function $\alpha_q : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, class-$\mathcal{K}_\infty$ functions $\underline{\gamma}_q$, $\overline{\gamma}_q$, and a positive and continuous function $\mu_q : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ such that

a) For all $\xi \in \Omega_q \setminus \bigcup_{i>q} \Omega'_i$,

$$\underline{\gamma}_q(|\xi|_{\mathcal{A}}) \leq V_q(\xi) \leq \overline{\gamma}_q(|\xi|_{\mathcal{A}}).$$

b) For all $\xi \in \Omega_q \setminus (\bigcup_{i>q} \Omega'_i \bigcup \mathcal{A})$, there exists $u_{q,\xi} \in \mathcal{U} \cap \mu_q(|\xi|_{\mathcal{A}})\mathbb{B}$, $\mathcal{U} \subset \mathbb{R}^r$, such that

$$\langle \nabla V_q(\xi), f_p(\xi, u_{q,\xi}) \rangle \leq -\alpha_q(|\xi|_{\mathcal{A}}).$$

c) For all $\xi \in \partial\Omega_q \setminus (\bigcup_{i>q} \Omega'_i \bigcup \mathcal{A})$, the $u_{q,\xi}$ of b) can be chosen such that

$$\langle n_q(\xi), f_p(\xi, u_{q,\xi}) \rangle \leq -\alpha_q(|\xi|_{\mathcal{A}}),$$

where $n_q(\xi)$ is the unit outward normal vector to Ω_q at ξ.

A stabilizing hybrid feedback can be constructed from a PCLF under an additional assumption. Suppose that, for each $\xi, v \in \mathbb{R}^n$, and $c \in \mathbb{R}$, the set $\{u \in \mathcal{U} : \langle v, f_p(\xi, u) \rangle \leq c\}$ is convex, which always holds if $\mathcal{U}$ is a convex set and $f_p(\xi, u)$ is affine in u. For each $q \in Q$, let

$$C_q = \overline{\Omega_q \setminus \bigcup_{i>q} \Omega'_i}, \qquad \Psi_q = \overline{\Omega'_q \setminus \bigcup_{i>q} \Omega'_i}, \qquad D_q = \varnothing.$$

It can be shown, in part through arguments similar to those used when constructing a feedback from a CLF, that for each

$q \in Q$ there exists a continuous mapping $k_q : C_q \setminus \mathcal{A} \to U$ such that, for all $\xi \in C_q \setminus \mathcal{A}$,

$$\langle \nabla V_q(\xi), f_p(\xi, k_q(\xi))\rangle \le -\alpha_q(|\xi|_{\mathcal{A}})/2,$$

which implies that the set $\mathcal{A}$ is globally pre-asymptotically stable for each controller $q \in Q$; moreover, for all $\xi \in \partial C_q \setminus \left(\mathcal{A} \cup \bigcup_{i>q} \Psi_i\right)$,

$$\langle n_q(\xi), f_p(\xi, u_{q,\xi})\rangle \le -\alpha_q(|\xi|_{\mathcal{A}})/2,$$

which implies that no maximal solution starting in Ψ_q reaches

$$\overline{\mathbb{R}^n \setminus \left(C_q \cup \bigcup_{i>q} \Psi_i\right)} \setminus \mathcal{A},$$

independently of how k_q is extended from $C_q \setminus \mathcal{A}$ to C_q. To preserve outer semicontinuity and local boundedness, set $k_q(\xi) = \mu_q(0)\mathbb{B}$ for $\xi \in C_q \cap \mathcal{A}$. The "Supervisor Assumption" holds. In particular, conditions 2a) and 2b) are addressed by the inequalities displayed above, while 2c) holds since $\Theta = \mathbb{R}^n$. By taking $\kappa_q(x_p, x_c) = k_q(x_p)$ for each $q \in Q$, the feedbacks k_q can be combined in a hybrid supervisor $\mathcal{H}_c$ with data given in (40). Although κ_q are set-valued on $\mathcal{A}$, the arguments used still apply. Furthermore, robustness results like Theorem 15 hold.

Example 37: Global Asymptotic Stabilization of a Point on the Three-Sphere

Consider the problem of globally asymptotically stabilizing the point $\xi^* := (0,0,0,1) \in \mathbb{S}^3 := \{x \in \mathbb{R}^4 : x^T x = 1\}$ for the system

$$\dot{\xi} = \Lambda(u)\xi \quad \xi \in \mathbb{S}^3,$$

where

$$\Lambda(u) = \begin{bmatrix} 0 & u_3 & -u_2 & u_1 \\ -u_3 & 0 & u_1 & u_2 \\ u_2 & -u_1 & 0 & u_3 \\ -u_1 & -u_2 & -u_3 & 0 \end{bmatrix}.$$

These equations can model the orientation kinematics of a rigid body expressed in terms of unit quaternions. In this case, ξ^* corresponds to a desired orientation and the inputs u_i correspond to angular velocities.

First we study the effect of the feedback controls $u_i := -\xi_i$, $i \in \{1,2,3\}$. Denote this feedback κ_2. Using the Lyapunov-function candidate $V(\xi) = 1 - \xi_4$, and noting that

$$\langle \nabla V(\xi), \Lambda(\kappa_2(\xi))\xi\rangle = 1 - \xi_4^2 \quad \text{for all } \xi \in \mathbb{S}^3,$$

it follows that the point ξ^* is rendered locally asymptotically stable with basin of attraction containing every point except $-\xi^*$.

Next we study the effect of the feedback controls $u_1 = -\xi_2$, $u_2 = \xi_1$, $u_3 = \xi_4$. Denote this feedback κ_1. Using the Lyapunov-function candidate $V(\xi) = 1 - \xi_3$, it follows that the point $(0,0,1,0)$ is rendered locally asymptotically stable with basin of attraction containing every point except $(0,0,-1,0)$.

Consider a hybrid supervisor of the classical controllers $\widetilde{\mathcal{H}}_q$, $q \in Q = \{1,2\}$, that are static and are given by $\widetilde{\kappa}_q(\xi) = \kappa_q(\xi)$. The "Supervisor Assumption" is satisfied for $\mathcal{A} = \{\xi^*\}$ and $\Theta = \mathbb{S}^3$ by taking $D_1 = D_2 = \varnothing$ and

$$\begin{aligned} \Psi_1 = C_1 &:= \{\xi \in \mathbb{S}^3 : \xi_4 \le -1/3\}, \\ C_2 &:= \{\xi \in \mathbb{S}^3 : \xi_4 \ge -2/3\}, \\ \Psi_2 &:= \{\xi \in \mathbb{S}^3 : \xi_4 \ge -1/3\}. \end{aligned}$$

Following the prescription of the components of a hybrid supervisor in (41) and (42), we get $H_1 = \Psi_2$, $H_2 = \overline{\mathbb{S}^3 \setminus C_2} = \{\xi \in \mathbb{S}^3 : \xi_4 \le -2/3\}$, and $J_q = 3 - q$. Then, using the definition of a supervisory controller in (40), we get a hybrid controller with state $x_c = q$, data $C_c = (C_1 \times \{1\}) \cup (C_2 \times \{2\})$, $f_c = 0$, $D_c = (H_1 \times \{1\}) \cup (H_2 \times \{2\})$, $G_c = 3 - q$, and $\kappa_c(\xi, q) = \kappa_q(\xi)$.

According to Corollary 33, this controller renders the set $\{\xi^*\} \times Q$ globally asymptotically stable. In fact, it can be established that the point $(\xi^*, 2)$ is globally asymptotically stable.

This example can also be interpreted in terms of the concept of patchy control-Lyapunov functions. Indeed, it can be verified that, with $Q = \{1,2\}$,

$$\begin{aligned} V_1(\xi) &= 1 - \xi_3, & \Omega_1' &= \Omega_1 := \mathbb{S}^3, \\ V_2(\xi) &= 1 - \xi_4, & \Omega_2' &:= \{\xi \in \mathbb{S}^3 : \xi_4 \ge -1/3\}, \\ & & \Omega_2 &:= \{\xi \in \mathbb{S}^3 : \xi_4 \ge -2/3\}, \end{aligned}$$

constitutes a patchy control-Lyapunov function. ■

Example 38: Nonholonomic Integrator

Consider the nonlinear control system

$$\left.\begin{aligned} \xi_1 &= u \\ \xi_2 &= u_2 \\ \xi_3 &= \xi_1 u_2 - \xi_2 u_1 \end{aligned}\right\} =: f_p(\xi, u)$$

known as the nonholonomic integrator. According to Brockett's necessary condition, the origin is not stabilizable by continuous, static-state feedback, nor it is robustly stabilizable by discontinuous, static state feedback. It is robustly stabilizable by time-varying feedback and by