General Hybrid Dynamical Systems: Modeling, Analysis, and Control

Michael S. Branicky

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

Abstract. Complex systems typically possess a hierarchical structure, characterized by continuous-variable dynamics at the lowest level and logical decision-making at the highest. Virtually all control systems today perform computer-coded checks and issue logical as well as continuous-variable control commands. Such are “hybrid” systems. In this paper, we introduce a formal notion of such systems: “general hybrid dynamical systems”; they are interacting collections of dynamical systems, evolving on continuous-variable state spaces, and subject to continuous controls and discrete phenomena. We discuss modeling issues, giving definitions and conditions for hybrid trajectories and providing a taxonomy for hybrid systems models. We review our hybrid systems analysis results, including topological issues, complexity and computation, stability tools, and analyzed examples. We summarize our hybrid control results, including optimal control theory, control algorithms, and solved examples.

1 Introduction

Hybrid systems involve both continuous-valued and discrete variables. Their evolution is given by equations of motion that generally depend on all variables. In turn these equations contain mixtures of logic, discrete-valued or digital dynamics, and continuous-variable or analog dynamics. The continuous dynamics of such systems may be continuous-time, discrete-time, or mixed (sampled-data), but is generally given by differential equations. The discrete-variable dynamics of hybrid systems is generally governed by a digital automaton, or input-output transition system with a countable number of states. The continuous and discrete dynamics interact at “event” or “trigger” times when the continuous state hits certain prescribed sets in the continuous state space. See Fig. 1(a).

Hybrid control systems are control systems that involve both continuous and discrete dynamics and continuous and discrete controls. The continuous dynamics of such a system is usually modeled by a controlled vector field or difference equation. Its hybrid nature is expressed by a dependence on some discrete phenomena, corresponding to discrete states, dynamics, and controls. The result is a system as in Fig. 1(b).

Real-World Examples. The prototypical hybrid systems are digital controllers, computers, and subsystems modeled as finite automata coupled with controllers and plants modeled by partial or ordinary differential equations or difference equations. Thus, such systems arise whenever one mixes logical decision-
making with the generation of continuous control laws. More specifically, real-world examples of hybrid systems include systems with relays, switches, and hysteresis, computer disk drives, transmissions, stepper motors, and other motion controllers, constrained robotic systems, intelligent vehicle/highway systems (IVHS), modern flexible manufacturing and flight control systems. Each of these, plus other examples, are discussed in some detail in [15]. Other important application areas for hybrid systems theory include embedded systems and analog/digital circuit co-design and verification.

![Diagram](image)

Fig. 1. (a) Hybrid System. (b) Hybrid Control System

**Paradigms.** We see four basic paradigms for the study of hybrid systems (summarizing from [15]):

1. **Aggregation:** suppress the continuous dynamics so that the hybrid system is a finite automaton or discrete-event dynamical system [1, 3, 26].

2. **Continuation:** suppress the discrete dynamics so that the hybrid system becomes a differential equation. This original idea of Prof. Sanjoy Mitter and the author is to convert hybrid models into purely continuous ones (modeled by ODEs) using differential equations that simulate automata [16, 20].

3. **Automatization or automata approach.** Treat the constituent systems as a network of interacting automata. The focus is on the input-output or language behavior. Automatization was pioneered in full generality by Nerode and Kohn [33].

4. **Systemization or systems approach.** Treat the constituent systems as interacting dynamical systems. The focus is on the state-space. Systemization was developed in full generality by the author [15], summarized here.

**Research Areas.** Research into hybrid systems may be broken down into four broad categories:

- **Modeling:** formulating precise models that capture the rich behavior of hybrid systems. i.e., How do we “fill in the boxes” in Fig. 1? What is their
dynamics? How can we classify their rich structure and behavior—and sort through the myriad hybrid systems models appearing?

- Analysis: developing tools for the simulation, analysis, and verification of hybrid systems, i.e., *How do we analyze systems as in Fig. 1(a)? What does continuity mean? How is their complexity? How do they differ from continuous dynamical systems? What do we test their stability? or analyze examples?*

- Control: synthesizing hybrid controllers—which issue continuous controls and make discrete decisions—that achieve certain prescribed safety and performance goals for hybrid systems, i.e., *How do we control a plant as in Fig. 1(b) with a controller as in Fig. 1(b)? How can we synthesize such hybrid controllers?*

- Design: conceiving new schemes and structures leading to easier modeling, verification, and control.

Outline of Paper. In the next two sections, we concentrate on modeling and its related questions above. In particular, we introduce general hybrid dynamical systems as interacting collections of dynamical systems, each evolving on continuous state spaces, and subject to continuous and discrete controls, and some other discrete phenomena. We give explicit instructions for computing the orbits and trajectories of general hybrid dynamical systems, including sufficient conditions for existence and uniqueness. We introduce a hierarchy of such systems and provide a taxonomy of them based on their structure and the discrete phenomena they exhibit. In Sections 4 and 5, we quickly summarize our work on analysis and control, answering the questions above. In all cases, the reader is referred to [15] for more details and complete references.

2 Hybrid Dynamical Systems

The notion of dynamical system has a long history as an important conceptual tool in science and engineering. It is the foundation of our formulation of hybrid dynamical systems. Briefly, a dynamical system [36] is a system \( \Sigma = [X, \Gamma, \phi] \), where \( X \) is an arbitrary topological space, the state space of \( \Sigma \). The transition semigroup \( \Gamma \) is a topological semigroup with identity. The (extended) transition map \( \phi : X \times \Gamma \to X \) is a continuous function satisfying the identity and semigroup properties [37]. A transition system is a dynamical system as above, except that \( \phi \) need not be continuous.

Examples of dynamical systems abound, including autonomous ODEs, autonomous difference equations, finite automata, pushdown automata, Turing machines, Petri nets, etc. As seen from these examples, both digital and analog systems can be viewed in this formalism. The utility of this has been noted since the earliest days of control theory [34].

We also denote by "dynamical system" the system \( \Sigma = [X, \Gamma, f] \), where \( X \) and \( \Gamma \) are as above, but the transition function \( f \) is the generator of the extended transition function \( \phi \).\(^1\) We may also refine the above concept by

\(^1\) In the case of \( \Gamma = \mathbb{Z}, f : X \to X \) is given by \( f \equiv \phi(-, 1) \). In the case of \( \Gamma = \mathbb{R} \),
introducing dynamical systems with initial and final states, input and output, and timing maps.\footnote{\textit{f} : \textit{X} \to \textit{TX} is given by the vector fields \( f(x) = d \phi(x,t)/dt |_{t=0} \).}

Briefly, a hybrid dynamical system is an indexed collection of dynamical systems along with some map for “jumping” among them (switching dynamical system and/or resetting the state). This jumping occurs whenever the state satisfies certain conditions, given by its membership in a specified subset of the state space. Hence, the entire system can be thought of as a sequential patching together of dynamical systems with initial and final states, the jumps performing a reset to a (generally different) initial state of a (generally different) dynamical system whenever a final state is reached.

More formally, a \textbf{general hybrid dynamical system (GHDS)} is a system \( H = [Q, \Sigma, A, G] \), with its constituent parts defined as follows.

- \( Q \) is the set of \textbf{index states}, also referred to as \textbf{discrete states}.
- \( \Sigma = \{ \Sigma_q \}_{q \in Q} \) is the collection of \textbf{constituent} dynamical systems, where each \( \Sigma_q = [X_q, \Gamma_q, \phi_q] \) (or \( \Sigma_q = [X_q, \Gamma_q, f_q] \)) is a dynamical system as above. Here, the \( X_q \) are the \textbf{continuous state spaces} and \( \phi_q \) (or \( f_q \)) are called the \textbf{continuous dynamics}.
- \( A = \{ A_q \}_{q \in Q} \), \( A_q \subset X_q \) for each \( q \in Q \), is the collection of \textbf{autonomous jump sets}.
- \( G = \{ G_q \}_{q \in Q}, G_q : A_q \to \bigcup_{q \in Q} X_q \times \{ q \} \), is the collection of \textbf{(autonomous) jump transition maps}.

These are also said to represent the \textbf{discrete dynamics} of the HDS.

Thus, \( S = \bigcup_{q \in Q} X_q \times \{ q \} \) is the \textbf{hybrid state space} of \( H \). For convenience, we use the following shorthand. \( S_q = X_q \times \{ q \} \) and \( A = \bigcup_{q \in Q} A_q \times \{ q \} \) is the \textbf{autonomous jump set} \( G : A \to S \) is the \textbf{autonomous jump transition map}, constructed componentwise in the obvious way. The \textbf{jump destination sets} \( D = \{ D_q \}_{q \in Q} \) are given by \( D_q = \pi_i (G(A) \cap S_q) \), where \( \pi_i \) is projection onto the \( i \)th coordinate. The \textbf{switching or transition manifolds}, \( M_{q,p} \subset A_q \) are given by \( M_{q,p} = G_q^{-1}(p, D_p) \), i.e., the set of states from which transitions from index \( q \) to index \( p \) can occur.

A GHDS can be pictured as an automaton. Here, each node is a constituent dynamical system, with the index the name of the node. Each edge represents a possible transition between constituent systems, labeled by the appropriate condition for the transition’s being “enabled” and the update of the continuous state.

Roughly,\footnote{\textit{Timing maps} provide the aforementioned mechanism for reconciling different “time scales,” by giving a uniform meaning to different transition semigroups in a hybrid system. See below.} the dynamics of the GHDS \( H \) are as follows. The system is assumed to start in some hybrid state in \( S \setminus A \), say \( s_0 = (x_0, q_0) \). It evolves according to \( \phi_{q_0}(x_0, \cdot) \) until the state enters—if ever—\( A_{q_0} \) at the point \( s^-_1 = (x^-_1, q_0) \). At
this time it is instantly transferred according to transition map to \( G_{q_1}(x_1, q_1) \equiv s_1 \), from which the process continues.

**Dynamical Systems.** \(|Q| = 1\) and \( A = \emptyset \) recovers all dynamical systems.

**Hybrid Systems.** The case \(|Q|\) finite, each \( X_q \) a subset of \( \mathbb{R}^n \), and each \( \Gamma_q = \mathbb{R} \) largely corresponds to the usual notion of a hybrid system, viz. a coupling of finite automata and differential equations [16, 17, 27]. Herein, a **hybrid system** is a GHDS with \( Q \) countable, and with \( \Gamma_q \equiv \mathbb{R} \) (or \( \mathbb{R}_+ \)) and \( X_q \subset \mathbb{R}^{d_q}, \; d_q \in \mathbb{Z}_+ \), for all \( q \in Q \); \[ [Q, \{X_q\}_{q \in Q}, \mathbb{R}_+, \{f_q\}_{q \in Q}, A, G] \], where \( f_q \) is a vector field on \( X_q \subset \mathbb{R}^{d_q} \).

**Changing State Space.** The state space may change. This is useful in modeling component failures or changes in dynamical description based on autonomous—and later, controlled—events which change it. Examples include the collision of two inelastic particles or an aircraft mode transition that changes variables to be controlled [32]. We also allow the \( X_q \) to overlap and the inclusion of multiple copies of the same space. This may be used, for example, to take into account overlapping local coordinate systems on a manifold [4].

**Refinements.** We may refine the concept of GHDS \( H \) by adding:

- inputs, including control inputs, disturbances, or parameters (see controlled HDS below).
- outputs, including **state-output** for each constituent system as for dynamical systems [15, 37] and **edge-output**: \( H = [Q, \Sigma, A, G, O, \eta] \), where \( \eta : A \rightarrow O \) produces an output at each jump time.
- \( \Delta : A \rightarrow \mathbb{R}_+ \), the jump delay map, which can be used to account for the time which abstracted-away, lower-level transition dynamics actually take.\(^5\)
- Marked states (including initial or final states), timing, or input and output for any constituent system.

**Example 1 — Reconciling Time Scales.** Suppose that each constituent dynamical system \( \Sigma_q \) of \( H \) is equipped with a timing map. That is \( \tau_q = \{\tau_q\}_{q \in Q} \) where \( \tau_q : X_q \times \Gamma_q \rightarrow \mathbb{R}_+ \). Then, we may construct **trajectories** for \( H \), i.e., a function from "real-time" to state. This is discussed below.

A **controlled general hybrid dynamical system** (GCHDS) is a system \( H_c = [Q, \Sigma, A, G, V, C, F] \), with its constituent parts defined as follows.

- \( Q, A \), and \( S \) are defined as above.
- \( \Sigma = \{\Sigma_q\}_{q \in Q} \) is the collection of controlled dynamical systems, where each \( \Sigma_q = [X_q, \Gamma_q, f_q, U_q] \) (or \( \Sigma_q = [X_q, \Gamma_q, \phi_q, U_q] \)) is a controlled dynamical

\(^4\) Here, we may take the view that the system evolves on the state space \( \mathbb{R}^* \times Q \), where \( \mathbb{R}^* \) denotes the set of finite, but variable-length real-valued vectors. For example, \( Q \) may be the set of labels of a computer program and \( x \in \mathbb{R}^* \) the values of all currently-allocated variables. This then includes Smale's tame machines [8].

\(^5\) Think of modeling the closure time of a discretely-controlled hydraulic valve or trade mechanism imperfections in economic markets.
system as above with (extended) transition map parameterized by control set \( U_q \).
- \( G = \{ G_q \}_{q \in Q} \), where \( G_q : A_q \times V_q \to S \) is the autonomous jump transition map, parameterized by the transition control set \( V_q \), a subset of the collection \( V = \{ V_q \}_{q \in Q} \).
- \( C = \{ C_q \}_{q \in Q} \), \( C_q \subseteq X_q \), is the collection of controlled jump sets.
- \( F = \{ F_q \}_{q \in Q} \), where \( F_q : C_q \to 2^S \), is the collection of controlled jump destination maps.

As shorthand, \( G, C, F, V \) may be defined as above. Likewise, jump destination sets \( D_a \) and \( D_c \) may be defined. In this case, \( D = D_a \cup D_c \).

Again, a GCHDS has an automaton representation. See Fig. 2. The notation \(![\text{condition}]\) denotes that the transition \textbf{must} be taken when enabled. The notation \(?[\text{condition}]\) denotes an enabled transition that \textbf{may be taken} on command; ";:e\" means reassignment to some value in the given set.

![Automaton Associated with GCHDS](image)

Roughly, the dynamics of \( H_c \) are as follows. The system is assumed to start in some hybrid state in \( S \setminus A \), say \( s_0 = (x_0, q_0) \). It evolves according to \( \phi_{q_0}(\cdot, v, u) \) until the state enters—if ever—either \( A_{q_0} \) or \( C_{q_0} \) at the point \( s^-_1 = (x_1, q_0) \). If it enters \( A_{q_0} \), then it \textbf{must} be transferred according to transition map \( G_{q_0}(x_1, v) \) for some chosen \( v \in V_{q_0} \). If it enters \( C_{q_0} \), then we \textbf{may} choose to jump and, if so, we may choose the destination to be any point in \( F_{q_0}(x_1) \). In either case, we arrive at a point \( s_1 = (x_1, q_1) \) from which the process continues. See Fig. 3.

**Notes.** (1) Nondeterminism in transitions may be taken care of by partitioning ?[condition] into those which are controlled and uncontrolled (cf. [28]). Disturbances (and other nondeterminism) may be modeled by partitioning \( U \), \( V \), and \( C \) into portions that are under the influence of the controller or nature respectively. Systems with state-output, edge-output, and autonomous and controlled jump delay maps (\( \Delta_a \) and \( \Delta_c \), respectively) may be added as above. (2) The model includes the "unified" model posed by Branicky, Borkar, and Mitter.
(BBM; [17]) and thus several other previously posed hybrid systems models [3, 4, 21, 33, 39, 41]. It also includes systems with impulse effect [5] and hybrid automata [25]. (3) In particular, our unified BBM model is, briefly, a controlled hybrid system, with the form $[\mathbb{Z}_+, [\{\mathbb{R}^d\}_{i=0}^\infty, \mathbb{R}_+, \{f_i\}_{i=0}^\infty, \mathbb{U}], \mathbb{A}, \mathbb{V}, \mathbb{G}, \mathbb{C}, \mathbb{F}]$. Control results for this model are summarized in Section 5.

**Definition 1.** The admissible control actions available are the continuous controls $u \in U_q$, exercised in each constituent regime; the discrete controls $v \in V_q$, exercised at autonomous jump times (i.e., on hitting set $A$); and the intervention times and destinations of controlled jumps (when the state is in $C$).

Now, we place some restrictions on GHDS in order to prove some behavioral properties. We assume that $\Gamma$ is an ordered set with the least upper bound property, equipped with the order topology. Note that this implies $\Gamma$ is a lattice [30]. We also assume addition to be order-preserving in the sense that if $a > 0$, then $a + b > b$. This last assumption ensures, among other things, that $\Gamma^+ = \{a \in \Gamma \mid a \geq 0\}$ is a semigroup; likewise for $\Gamma^-$, defined symmetrically. For brevity, we call such a group (semigroup) time-like.\(^6\)

We now consider several initial value problems for GHDS. First, in the time-like case, given dynamical system $[X, \Gamma, \phi]$, we may define the positive orbit

---

\(^6\) The most widely used time-like groups are $\mathbb{R}$, $\mathbb{Z}$, and $\rho \mathbb{Z}$, $\rho \in \mathbb{R}$, each under addition and in the usual order. Example semigroups are $\mathbb{R}_+$, $\rho \mathbb{Z}_+$, and the free monoid generated by a finite, ordered alphabet (in the dictionary order).
of the point $x$ as $P(x) \equiv \phi(x, \Gamma^+)$.\footnote{The negative orbit $B(x)$ may be defined even in the non-reversible case by $y \in B(x)$ if and only if $x \in P(y)$.}

**Problem 2 — Reachability Problem.** Compute the positive orbit for GHDS $H$.

**Solution 3.** The positive orbit is the set defined as follows. We consider only initial points in $I \equiv X \setminus A$. We restrict ourselves to the case where $A_q$ is closed and $D_q \cap A_q = \emptyset$. Suppose $s_0 = (x_0, q_0) \in I$. If $P_{q_0}(x_0) = \phi_{q_0}(x_0, \Gamma^+)$ does not intersect $A_{q_0}$ we are done: the positive orbit is just $P_{q_0}(x_0)$. Else, let $g_1 = \inf \{ g \in \Gamma_{q_0}^+ \mid \phi_{q_0}(x_0, g) \in A_{q_0} \}$. Since $A_{q_0}$ closed, $\Gamma^+$ is time-like, and $\phi$ is continuous, the set in the inf equation is closed, $g_1$ exists, and $x_1^- = \phi_{q_0}(x_0, g_1) \in A_{q_0}$. Define $s_1 = G_{q_0}(x_1^-) \in I$ and continue.

When a GHDS is time-uniform with time-like group $\Gamma$, it induces a $\Gamma^+$-transition system $[I, \Gamma^+, \Phi]$. In this case, we may define its (forward) trajectory as a function from $\Gamma^+$ into $I$.

**Problem 4 — Trajectory Problem.** Compute the trajectories for GHDS $H$.

**Solution 5.** A modification of the above. See [15] for details.

The above constructions allow us to formulate stability and finite-time reachability problems. Note that trajectories may not be extendible to all of $\Gamma^+$, i.e., we have not precluded the accumulation of an infinite number of jumps in finite time; this can be removed in the case of hybrid systems by, for example, assuming uniform Lipschitz continuity of the vector fields and uniform separation of the jump and destination set $A$ and $D$ [17]. **Note:** From above, if $D_q \cap A_q = \emptyset$ and $A_q$ closed then (positive) orbits and trajectories exist (up to a possible accumulation time of finite jumps) and are unique. Similar to reachability, we have the following.

**Problem 6 — Accessibility Problem.** Compute the set of points accessible under all admissible control actions from initial set $I$ for GCHDS system $H_c$.

The solution is largely the same as above except that we must vary over all admissible control actions.

### 3 Classification of Hybrid Dynamical Systems

**A Taxonomy for GHDS.** The scope of hybrid dynamical systems presents a myriad of modeling choices. In this section, we classify them according to their structure and the discrete phenomena they possess. Below, the prefixes “c-,” “d-,” and “t-” are used as abbreviations for “continuous-,” “discrete-,” and “time-” respectively. If no prefix is given, either can be used.

Our **structural classification** is roughly captured by the following list.
- **Time-uniform.** The semigroups may be all be the same for each \( q \).
- **Continuous-time, discrete-time, sampled-data.** Each constituent dynamical system may be of a special type that evolves in continuous-time \((\Gamma = \mathbb{R})\), discrete-time \((\Gamma = \mathbb{Z})\), or a mixture. However, if the GHDS is time-uniform, we refer to it by the appropriate label, e.g., continuous-time-uniform.
- **C-uniform.** The ambient state space may be the same for each \( q \).
- **C-Euclidean, c-manifold.** Each ambient state space may be Euclidean\(^8\) or a smooth manifold.
- **D-compact, d-countable, d-finite.** Special cases arise when the index space is compact, finite, or countably infinite.
- **Dynamically-uniform.** The dynamics may be the same for each \( q \). Strictly, such a case would also require that the system be c-uniform and time-uniform. In these systems, the interesting dynamics arises from the transition map \( G_q [1, 26] \).
- **D-concurrent versus d-serial.** We may or may not allow more than one discrete jump to occur at a given moment of time.
- **Deterministic versus nondeterministic.**
- **Nonautonomous versus autonomous.** The continuous (or discrete)

Finally, a hybrid dynamical system may also be classified according to the discrete dynamic phenomena that it exhibits as follows (cf. [17]).

- **Autonomous-switching.** The autonomous jump map \( G \equiv \nu \) is the identity in its continuous component, i.e., \( \nu : A \rightarrow S \) has \( \nu(x, q) = (x, q') \).
- **Autonomous-impulse.** \( G \equiv J \) is the identity in its discrete component.
- **Controlled-switching.** The controlled jump map \( F \) is the identity in its continuous component, i.e., \( F(x, q) \subset \{x\} \times Q \).
- **Controlled-impulse.** \( F \) is the identity in its discrete component.

With this notation, our GHDS model admits some special cases:

- \( H \) autonomous-impulse with \( |Q| = 1 \) and \( \Gamma = \mathbb{R} \) is an autonomous system with impulse effect [5].
- \( H \) c-uniform, time-uniform, and autonomous-switching is an autonomous switched system [14].
- \( H \) continuous-time-uniform, c-Euclidean-uniform, d-countable, is what we called a hybrid system.

**Classifying Hybrid Systems.** In this section, we give explicit representations of the different classes of hybrid systems arising from the definitions above. We concentrate on the c-continuous-time, c-uniform, d-finite, time-invariant, autonomous case. Extensions to other cases above are straightforward.

\(^8\) A subset of \( \mathbb{R}^n \) in the usual topology
A (continuous-time) autonomous-switching hybrid system may be defined as follows:

\[ \dot{x}(t) = f(x(t), q(t)), \quad q^+(t) = \nu(x(t), q(t)), \]

where \( x(t) \in \mathbb{R}^n, \) \( q(t) \in Q \simeq \{1, \ldots, N\}. \) Here, \( f(\cdot, q) : \mathbb{R}^n \to \mathbb{R}^n, q \in Q, \) each globally Lipschitz continuous, is the continuous dynamics of Eq. (1); and \( \nu : \mathbb{R}^n \times Q \to Q \) is the finite dynamics of Eq. (1). An example is the Tavernini model. Adding a continuous control yields Witsenhausen’s model.

An autonomous-impulse hybrid system is a system

\[ \dot{x}(t) = f(x(t)), \quad x(t) \not\in M; \quad x^+(t) = J(x(t)), \quad x(t) \in M; \]

where \( x(t) \in \mathbb{R}^n, \) and \( J : \mathbb{R}^n \to \mathbb{R}^n. \) Examples include autonomous systems with impulse effect. Finally, a hybrid system with autonomous switching and autonomous impulses (i.e., the full power of autonomous jumps) is just a combination of those discussed above, where \( x(t) \in \mathbb{R}^n \) and \( q(t) \in Q \subset \mathbb{Z}. \) Examples include the BGM model and hence autonomous versions of of the models in [3, 4, 21, 33] (see [16, 17]).

4 Hybrid Systems Analysis

**Topological Results.** In traditional feedback control systems—continuous-time, discrete-time, sampled-data—the maps from output measurements to control inputs are continuous (in the usual metric-based topologies). Continuity of state evolution and controls with respect to the states also plays a role. With these considerations in mind, we want to examine continuity for systems as in Fig. 1(a), where the set of symbols, automaton states, and outputs, are finite sets and the plant and controls belong to a continuum. Yet, in general, hybrid systems are not even continuous in the initial condition:

**Example 2.** Consider the following hybrid system on \( X_1 = X_2 = \mathbb{R}^2. \) The continuous dynamics is given by \( f_1 = (1, 0)^T \) and \( f_2 = (0, 1)^T. \) The discrete dynamics is given by \( A_1 = [0, 1]^2 \) and \( G(x, 1) = (x, 2). \) Now consider the initial conditions \( x(0) = (-\epsilon, -\epsilon)^T \) and \( y(0) = (-\epsilon, 0)^T. \) Note that \( x(1) = (1 - \epsilon, -\epsilon) \) but \( y(1) = (0, 1 - \epsilon). \) Clearly, no matter how small \( \epsilon, \) hence \( \|x(0) - y(0)\|_{\infty} \) is chosen, \( \|x(1) - y(1)\|_{\infty} = 1. \)

Further, we note that the only continuous maps from a connected set to a disconnected one are the constant ones. Hence, the usual discrete topologies on a set of symbols do not lead to nontrivial continuous AD maps.

---

9 Automata may be thought of as operating in “continuous time” by the convention that the state, input, and output symbols are piecewise right- or left- continuous functions. The notation \( t^- \) (resp. \( t^+ \)) may be used to indicate that the finite state is piecewise continuous from the right (resp. left), e.g., \( q(t) = \nu(x(t), q(t^-)) \) Here we have used Sontag’s more evocative discrete-time transition notation [37] \( q^+(t) \) to denote the “successor” of \( q(t). \)
We have examined topologies that lead to continuity of each member of topologies for which the AD maps from measurements to symbols are continuous in [10]. In dynamics terms, we examined topologies that lead to continuity of each member of the family of maps \( G \circ \phi_q \). One topology in particular, proposed by Nerode and Kohn [33], is studied in depth.

We also looked at what happens if we attempt to “complete the loop” in Fig. 1(a), by also considering the DA maps. We exhibited a topology making the whole loop continuous [10]. We further examined a different view of hybrid systems as a set of continuous controllers, with switching among them governed by the discrete state. A popular example is fuzzy control systems, consisting of a continuous plant controlled by a finite set of so-called fuzzy rules. On the surface, they are hybrid systems. Yet, we showed fuzzy control leads to continuous maps (from measurements to controls), and all such continuous maps may be implemented via fuzzy control [10].

**Complexity Results.** Computational equivalence (or simulation of computational capabilities) may be shown by comparing accepted languages [29], or simulating step-by-step behavior [9]. In order to examine the computational capabilities of hybrid and continuous systems we first must introduce notions of a continuous-time system simulating a discrete-time one [16]:

**Definition 7.** A continuous-time transition system \([X, R_+, f]\) simulates via section or S-simulates a discrete-time transition system \([Y, Z_+, F]\) if there exist a continuous surjective partial function \( \psi : X \to Y \) and \( t_0 \in R_+ \) such that for all \( x \in \psi^{-1}(Y) \) and all \( k \in Z_+ \), \( \psi(f(x, kt_0)) = F(\psi(x), k) \).

**Definition 8.** A continuous-time transition system \([X, R_+, f]\) simulates via intervals or I-simulates a discrete-time transition system \([Y, Z_+, F]\) if there exist a continuous surjective partial function \( \psi : X \to Y \) and \( \epsilon > 0 \) such that \( V \equiv \psi^{-1}(Y) \) is open and for all \( x \in V \) the set \( T = \{ t \in R_+ : f(x, t) \in V \} \) is a union of intervals \( (\tau_k, \tau'_k) \), \( 0 = \tau_0 < \tau'_0 < \tau_1 < \tau'_1 < \cdots, |\tau'_k - \tau_k| \geq \epsilon, \) with \( \psi(f(x, \tau_k)) = F(\psi(x), k) \), for all \( k \in (\tau_k, \tau'_k) \).

When the continuous-time transition system is a HDS, the maps \( \psi \) above can be viewed as an edge-output and state-output map, respectively. In this case, S-simulation can be viewed as equivalent behavior at equally-spaced edges and I-simulation as equivalent behavior in designated nodes. S- and I-simulation are distinct; SI-simulation denotes both holding. In [16] we show

- Every dynamical system \([R^n, Z, F]\) can be S-simulated by an autonomous-switching, two-discrete-state hybrid system on \( R^{2n} \).
- Every dynamical system \([R^n, Z_+, F]\) can be S-simulated by an autonomous-jump, two-discrete-state hybrid system on \( R^n \).
- Every dynamical system \([Y, Z, F]\), \( Y \subset Z^n \), can be SI-simulated by a (continuous) dynamical system of the form \([R^{2n+1}, R_+, f]\). Furthermore, if \( Y \) is bounded \( f \) can be taken Lipschitz continuous.

As corollaries to the last, we have (via demonstrated isomorphisms with dynamical systems on \( Z \))
Every Turing machine, pushdown automata, and finite automaton can be SI-simulated by a (continuous) dynamical system of the form $[\mathbb{R}^3, \mathbb{R}_+, f]$.

Using SI-simulation, there is continuous ODE in $\mathbb{R}^3$ with the power of universal computation.

Noting that even ordinary dynamical systems are so computationally powerful, we have used the famous asynchronous arbiter problem [11, 31, 40] to distinguish between dynamical and hybrid systems [16]. In particular, we settled the problem in an ODE framework by showing that no system of the form $[\mathbb{R}^n, \mathbb{R}_+, B, f, W, h]$, with $f$ Lipschitz and $h$ continuous, can implement an arbiter [11, 16]. On the other hand, we exhibited a hybrid system of the form $[\{1, 2\}, [\mathbb{R}^n, \mathbb{R}_+, B, \{f_1, f_2\}, W, h], A, G]$, with each $f_i$ Lipschitz, $h$ continuous, and $G$ autonomous-switching, that satisfies the arbiter specifications.

**Analysis Tools.** We have developed general tools for analyzing continuous switched systems (where the vector fields agree at switching times) [12]. For instance, we prove an extension of Bendixon’s Theorem to the case of Lipschitz continuous vector fields. This gives us a tool for analyzing the existence of limit cycles of continuous switched systems. We also proved a lemma dealing with the continuity of differential equations with respect to perturbations that preserve a linear part. Colloquially, this lemma demonstrates the robustness of ODEs with a linear part (Linear Robustness Lemma; [13]). This lemma is useful in easily deriving some of the common robustness results of nonlinear ODE theory [6]. It also becomes useful in studying singular perturbations if the fast dynamics are such that they maintain the corresponding algebraic equation to within a small deviation. Some simple propositions that allow this appear in [12].

In [14], we examined stability of switched and hybrid systems. We introduced multiple Lyapunov functions as a tool for analyzing Lyapunov stability of such systems. The idea here is to impose conditions on switching that guarantee stability when we have Lyapunov functions for each system $f_i$ individually. Iterative function systems were presented as a tool for proving Lagrange stability and positive invariance. We also addressed the case where the finite switching set is replaced by an arbitrary compact set.

**Analyzing Examples.** We have analyzed example systems arising from a realistic aircraft controller problem (called the max system) which logically switches between two controllers—one for tracking and one for regulation about a fixed angle of attack—in order to achieve reasonable performance and safety [12]. While stability of such hybrid systems has previously only been examined using simulation [38], we were able to prove global asymptotic stability for a meaningful class of cases [12]. Using our linear robustness lemma to compare ODE solutions, we extended the result to a class of continuations of these systems, in which an ODE dynamically smooths the logical nonlinearity.

---

10 An arbiter is a device that can be used to decide the winner of two-person races (within some tolerances). It has two input buttons, $B_1$ and $B_2$, and two output lines, $W_1$ and $W_2$, that can each be either 0 or 1.
5 Hybrid Control

Theoretical Results. We consider the following optimal control problem for controlled hybrid systems. Let $a > 0$ be a discount factor. We add to our model the following known maps:

- **Running cost**, $k : S \times U \to \mathbb{R}_+$.
- **Autonomous jump cost and delay**, $c_a : A \times V \to \mathbb{R}_+$ and $\Delta_a : A \times V \to \mathbb{R}_+$.
- **Controlled jump (or impulse) cost and delay**, $c_c : C \times D_c \to \mathbb{R}_+$ and $\Delta_c : C \times D_c \to \mathbb{R}_+$.

The total discounted cost is defined as

$$\int_T e^{-at} k(x(t), u(t)) \, dt + \sum_i e^{-a\sigma_i} c_a(x(\sigma_i), u_i) + \sum_i e^{-a\zeta'_i} c_c(x(\zeta_i), x(\zeta'_i))$$

(3)

where $T = \mathbb{R}_+ \backslash (\bigcup_i [\tau_i, I_i])$, $\{\sigma_i\}$ (resp. $\{\zeta_i\}$) are the successive pre-jump times for autonomous (resp. impulsive) jumps and $\zeta'_i$ is the post-jump time (after the delay) for the $j$th impulsive jump. The decision or control variables over which Eq. (3) is to be minimized are the admissible controls of our controlled hybrid system (see Def. 1). Under some assumptions (the necessity of which are shown via examples) we have the following results [17]:

- A finite optimal cost exists for any initial condition. Furthermore, there are only finitely many autonomous jumps in finite time.
- Using the relaxed control framework, an optimal trajectory exists for any initial condition.
- For every $\epsilon > 0$ an $\epsilon$-optimal control policy exists wherein $u(\cdot)$ is precise, i.e., a Dirac measure.
- The value function, $V$, associated with the optimal control problem is continuous on $S \backslash (\partial A \cup \partial C)$ and satisfies the following generalized quasi-variational inequalities (GQVI's).

1. $x \in S \backslash A$: $\min_u (\nabla_x V(x), f(x, u)) - aV(x) + k(x, u) \leq 0$.
2. On $C$: $V(x) \leq \min_{z \in D_c} \{c_c(x, z) + e^{-a\Delta_a(x, z)} V(z)\}$
3. On $A$: $V(x) \leq \min_u \{c_a(x, u) + e^{-a\Delta_a(x, u)} V(G(x, u))\}$
4. On $C$: $(1) \cdot (2) = 0$

Algorithms and Examples. We have outlined four approaches to solving the generalized quasi-variational inequalities (GQVI's) associated with optimal hybrid control problems [15]. Our algorithmic basis for solving these GQVI's is the generalized Bellman Equation: $V^*(x) = \min_{p \in \Pi} \{g(x, p) + V^*(x'(x, p))\}$, where $\Pi$ is a generalized set of actions. The three classes of actions available in our hybrid systems framework at each $x$ are the admissible control actions from Def. 1. From this viewpoint, generalized policy and value iteration become solution tools [15].

The key to efficient algorithms for solving optimal control problems for hybrid systems lies in noticing their strong connection to the models of impulse control
and piecewise-deterministic processes. Making this explicit, we have developed algorithms similar to those for impulse control [24] and one based on linear programming [23, 35] (see [15]).

Three illustrative examples are solved in [15]. First, we consider a hysteresis system that exhibits autonomous switching and has a continuous control. Then, we discuss a satellite station-keeping problem, where the on-off nature of the satellite's reaction jets creates a system involving controlled switching. We end with a transmission control problem, where the goal is to find the hybrid strategy of continuous accelerator input and discrete gear-shift position to achieve maximum acceleration. In each example, the synthesized optimal controllers verify engineering intuition.

References