

Hybrid Dynamical Systems, or HDS: The Ultimate Switching Experience

Michael S. Branicky*

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

Abstract

In previous work I have concentrated on formalizing the notion of a hybrid system as switching among an indexed collection of dynamical systems. I have also studied in some depth the modeling, analysis, and control of such systems. Here, I give a quick overview of the area of hybrid systems. I also briefly review the formal definition and discuss the main approaches taken in the study of hybrid systems. Finally, I elucidate issues in each of the research areas in light of previous results.

1 Introduction

Many complicated control systems today (e.g., those for flight control, manufacturing systems, and transportation) have vast amounts of computer code at their highest level. More pervasively, programmable logic controllers are widely used in industrial process control. We also see that today's products incorporate logical decision-making into even the simplest control loops (e.g., embedded systems). Thus, virtually all control systems today issue continuous-variable controls and perform logical checks that determine the mode—and hence the control algorithms—the continuous-variable system is operating under at any given moment. As such, these “hybrid control” systems offer a challenging set of problems.

Hybrid systems involve both continuous-valued and discrete variables. Their evolution is given by equations of motion that generally depend on both. 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 Figure 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 Figure 1(b).

*Supported by Army Research Office and Center for Intelligent Control Systems, grants DAAL03-92-G-0164/-0115. MIT, 35-415, Cambridge, MA 02139. branicky@lids.mit.edu

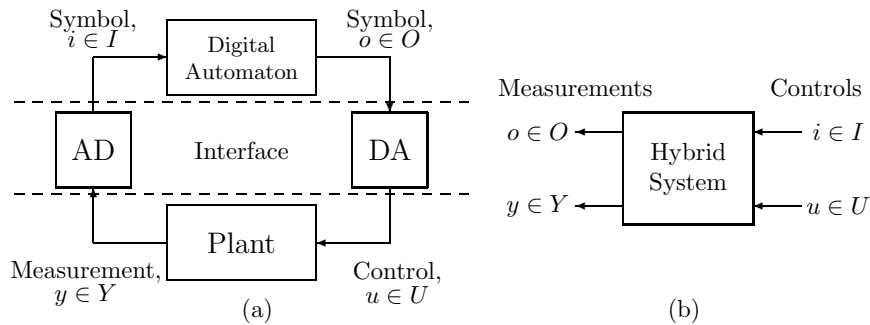


Figure 1: (a) Hybrid System. (b) Hybrid Control System.

Hybrid systems research 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 Figure 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 Figure 1(a)? What does continuity mean? What is their complexity? How do they differ from continuous dynamical systems? How 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 Figure 1(b) with a controller as in Figure 1(b)? How can we synthesize such hybrid controllers?*
- **Design:** conceiving new schemes and structures leading to easier modeling, verification, and control.

In previous work [8, 10, 11, 12, 14, 17] I have concentrated on formalizing the notion of a hybrid system sketched above and the subsequent answering of the above questions for that model. Here, I briefly review the formal definition. I then discuss the main approaches taken in the study of hybrid systems. Finally, I elucidate issues in the above research areas in light of previous results.

2 Related Work

Hybrid systems are certainly pervasive today. But they have been with us at least since the days of the relay. In control theory, there has certainly been a lot of related work in the past, including variable structure systems, jump linear systems, systems with impulse effect, impulse control, and piecewise deterministic processes. More recently, in computer science there has been a successive build-up in the large formal verification literature toward verification of systems that include both continuous and discrete variables. In control theory we have witnessed a resurgence in examining quantization effects [22, 29, 39, 40] and recent progress in analyzing switched [11, 31, 38, 45] and discretely-controlled continuous-variable systems [20, 30, 35, 34].

3 Hybrid Dynamical Systems

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.

Formally, a **controlled general hybrid dynamical system (CGHDS)** is a system $H_c = [Q, \Sigma, \mathbf{A}, \mathbf{G}, \mathbf{V}, \mathbf{C}, \mathbf{F}]$, with constituent parts as follows.

- Q is the set of **index states**, or **discrete states**.
- $\Sigma = \{\Sigma_q\}_{q \in Q}$ 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 system. Here, the X_q are the **continuous state spaces** and ϕ_q (or f_q) are called the **continuous dynamics**.
- $\mathbf{A} = \{A_q\}_{q \in Q}$, $A_q \subset X_q$ for each $q \in Q$, is the collection of **autonomous jump sets**.
- $\mathbf{G} = \{G_q\}_{q \in Q}$, where $G_q : A_q \times V_q \rightarrow S$ is the **autonomous jump transition map**, parameterized by the **transition control set** V_q , a subset of the collection $\mathbf{V} = \{V_q\}_{q \in Q}$; they are said to represent the **discrete dynamics**.
- $\mathbf{C} = \{C_q\}_{q \in Q}$, $C_q \subset X_q$, is the collection of **controlled jump sets**.
- $\mathbf{F} = \{F_q\}_{q \in Q}$, where $F_q : C_q \rightarrow 2^S$, is the collection of **controlled jump destination maps**.

Thus, $S = \bigcup_{q \in Q} X_q \times \{q\}$ is the **hybrid state space** of H . The case where the sets U_q and \mathbf{G} through \mathbf{F} above are empty is simply a **general hybrid dynamical system (GHDS)**: $H = [Q, \Sigma, \mathbf{A}, \mathbf{G}]$,

A CGHDS can be pictured as an automaton as in Figure 2. There, 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 (cf. [26]). The notation $![\text{condition}]$ denotes that the transition *must* be taken when enabled. The notation $?[\text{condition}]$ denotes an enabled transition that *may be taken* on command; “: \in ” means reassignment to some value in the given set.

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 $\phi_{q_0}(\cdot, \cdot, u)$ 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 *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 *may* choose to jump and, if so, we may choose the destination to be any point in $F_{q_0}(x_1^-)$. Either way, we arrive at a point $s_1 = (x_1, q_1)$ from which the process continues. See Figure 3.

The following are some important notes about CGHDS:

Dynamical Systems. GHDS with $|Q| = 1$ and \mathbf{A} empty recover all these.

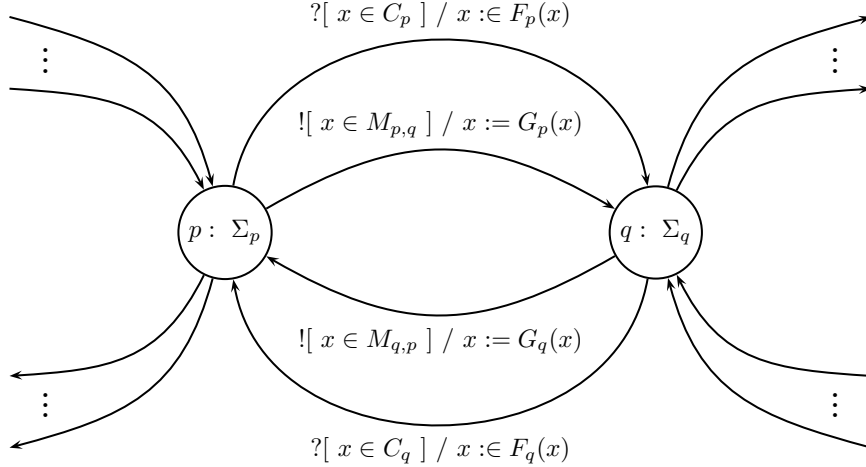


Figure 2: Automaton Associated with CGHDS.

Hybrid Systems. The case of GHDS with $|Q|$ 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 [13, 14, 25]. 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 $q \in Q$: $[Q, [\{X_q\}_{q \in Q}, \mathbb{R}_+, \{f_q\}_{q \in Q}], \mathbf{A}, \mathbf{G}]$, where f_q is a vector field on $X_q \subset \mathbb{R}^{d_q}$.¹

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 [33]. 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, say, 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 [12, 42] and **edge-output**: $H = [Q, \Sigma, \mathbf{A}, \mathbf{G}, \mathbf{O}, \eta]$, 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.²
- Marked states (including initial or final states), timing, or input and output for any constituent system.

¹Here, 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 [6].

²Think of modeling the closure time of a discretely-controlled hydraulic valve or trade mechanism imperfections in economic markets.

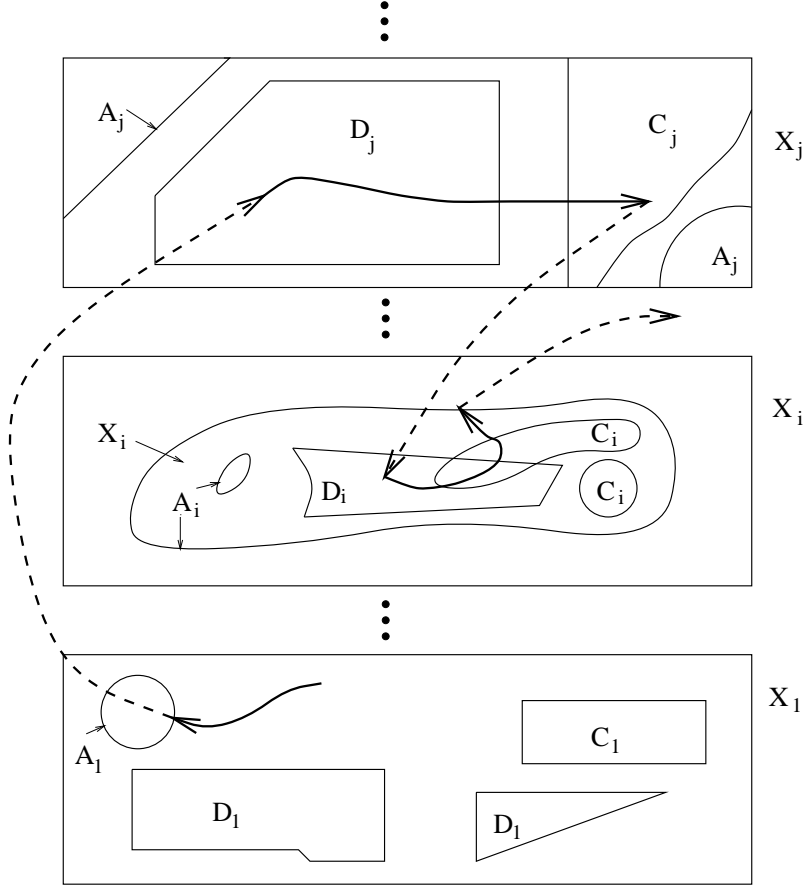


Figure 3: Example dynamics of CGHDS.

Other Notes. (1) **Nondeterminism** in transitions may be taken care of by partitioning C into those which are controlled and uncontrolled (cf. DEDS). **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 (Δ_a and Δ_c , respectively) may be added as above. (2) The model includes the “unified” model posed by Branicky, Borkar, and Mitter (**BBM**; [14]) and thus several other previously posed hybrid systems models [3, 4, 19, 36, 43, 46]. It also includes systems with impulse effect [5] and hybrid automata [23]. (3) In particular, our unified BBM model is, briefly, a **controlled hybrid system**, with the form $[\mathbb{Z}_+, [\{\mathbb{R}^{d_i}\}_{i=0}^\infty, \mathbb{R}_+, \{f_i\}_{i=0}^\infty, \mathbf{U}], \mathbf{A}, \mathbf{V}, \mathbf{G}, \mathbf{C}, \mathbf{F}]$. The **admissible control actions** available in this model are the **continuous controls** $u \in U_q$, exercised in each constituent regime; the **discrete controls** $v \in V_q$, exercised at the autonomous jump times (i.e., on hitting set A); and the **intervention times** and **destinations** of controlled jumps (when the state is in C). Control results for this model are given in [14, 17].

4 Paradigms for Hybrid Systems

I see four basic paradigms for the study of hybrid systems: aggregation, continuation, automatization, and systemization. The first two approaches deal with the different sides—analogue and digital—of hybrid systems. They attempt to suppress the hybrid nature of the system by converting it into a purely discrete or purely continuous one, respectively. The last two approaches are more general and potentially more powerful. Under them, a hybrid system is seen directly as an interacting set of automata or dynamical systems; they complement the input-output and state-space paradigms, respectively, of both control theory and computer science. More specifically, the approaches are:

Aggregation. That is, suppress the continuous dynamics so that the hybrid system is a finite automaton or discrete-event dynamical system (DEDS). This is the approach most often taken in the literature, e.g., [3]. The drawback of this approach is three-fold.

- *Nondeterminism*: one usually obtains nondeterministic automata [3, 27].
- *Nonexistence*, i.e., even if clever constructions are used, no finite automaton may exist that captures the combined behavior [24].
- *Partition Problem*. It appears a conceptually deep problem to determine when there exist partitions of just a continuous system such that its dynamics is captured by a meaningful finite automaton. “Meaningful,” since we note that every system is homomorphic to one with a single equilibrium point [41]. The answer thus depends on the dynamical behavior one is interested in capturing and the questions one is asking.

The aggregation program has been fully carried out so far only under strong assumptions on the hybrid system [1, 24].

Continuation. The complement of aggregation, that is, suppress the discrete dynamics so that the hybrid system becomes a differential equation. This original idea of Prof. Sanjoy Mitter and myself is to convert hybrid models into purely continuous ones—modeled by differential equations—using differential equations that simulate finite automata. In this familiar, unified realm one could answer questions of stability, controllability, and observability, converting them back to the original model by taking a “singular limit.” For instance, one would like tools that allow one to conclude the following: if a “sufficiently close” continuation of a system is stable, then the original system is stable. Such a program is possible in light of the existence of simple continuations of finite automata [13, 18] and pushdown automata and Turing machines [13]. The drawback of this approach is three-fold.

- *Arbitrariness*, i.e., how one accomplishes the continuation is largely arbitrary. For example, to interpolate or “simulate” the step-by-step behavior of a finite automaton Brockett used his double-bracket equations [19] and the author used stable linear equations [7, 13]. In certain cases this freedom is an advantage. However, care must be taken to insure that the dynamics used does not introduce spurious behavior (like unwanted equilibria) or that it itself is not hard to analyze or predict.
- *Hiding Complexity*. One cannot generally get rid of the underlying discrete dynamics, i.e., the complexity is merely hidden in the “right-hand side” of the continuation differential equations [9].

- *Artificiality.* It can lead to a possibly unnatural analytical loop of going from discrete to continuous and back to discrete. Cf. Chen’s recent results in stochastic approximation vis-à-vis Kushner’s [21].

The combination of these points has been borne out by some experience: it can be easier to examine the mixed discrete-continuous system. Cf. my switched aircraft controller analysis [10] and Megretsky’s relay system analysis [32].

Automatization. Treat the constituent systems as a network of interacting automata [36, p. 325]. The focus is on the input-output or language behavior. This language view has been largely taken in the computer science literature in extending the dynamical behavior of finite automata incrementally toward full hybrid systems (see [1, 25] for background).

Automatization was pioneered in full generality by Nerode and Kohn [36]. The viewpoint is that systems, whether analog or digital, are automata. As long as there is compatibility between output and input alphabets, links between automata can be established. However, there is still the notion of “reconciling different time scales” [36, p. 325]. For instance, a finite automaton receives symbols in abstract time, whereas a differential equation receives inputs in “real time.” This reconciliation can take place by either of the following: forcing synchronization at regular sampling instants [36, p. 333], or synchronizing the *digital* automaton to advance at event times when its input symbols change [36]. For hybrid systems of interest, the latter mechanism appears more useful. It has been used in many hybrid systems models, e.g., [3, 19]. For a fruitful example of this approach see [23].

Systemization. Treat the constituent systems as interacting dynamical systems [41]. The focus is on the state-space. This state-space view has been taken most profitably in the work of Witsenhausen [46] and Tavernini [43].

Systemization is developed in full generality in the thesis [12]. The viewpoint is that systems, whether analog or digital, are dynamical systems. As long as there is compatibility at switching times when the behavior of a system changes in response to a logical decision or event occurrence, links between these dynamical systems can be established. Again, there is still the notion of reconciling dynamical systems with different time scales (i.e., transition semigroups). For instance, a finite automaton abstractly evolves on the positive integers (or on the free monoid generated by its input alphabet), whereas a differential equation evolves on the reals. This reconciliation can take place by either or both of the following: sequentially synchronizing the dynamical systems at event times when their states enter prescribed sets, or forcing uniform semigroup structure via “timing maps.” Both approaches are introduced here, but the concentration is on the former.

Systemization is established in my formulation of hybrid dynamical systems above. It has been used in examining complexity and simulation capabilities of hybrid systems [13], analyzing their stability [10, 11], and in establishing the first comprehensive state-space paradigm for the control of hybrid systems [14].

5 Thoughts on Hybrid Systems

Modeling. There are a myriad of hybrid systems models (see [12] and the collections [25, 2] for references). Of less prevalence are results proven about these models. I mention a few here. Tavernini [43], for instance, has given a model plus assumptions and proven properties of trajectories and simulations of trajectories of hybrid systems. In early work, Witsenhausen [46] gave a model

and derived some optimal control results. Deshpande [23] took an automata approach to modeling and derived results for viable control of their hybrid systems. Above, and in [14, 12] I have given a model which (1) captures the phenomena associated with hybrid systems, (2) subsumes many of the models presented in the literature, (3) is amenable to the posing and solution of optimal control problems (see the discussion below).

One needs to explore the plethora of modeling choices available in hybrid systems. Since hybrid systems include dynamical systems as a subset, subclasses which permit efficient simulation, analysis, and verification should be explored. I believe that such a program is indeed being carried out by the computer scientists. Control theorists should do the same in their field in examining the hybrid control of hybrid systems.

Analysis. In general, hybrid systems analysis is theoretically intractable. The reason is that even simple hybrid systems possess the power of universal computation [1, 12]. Thus, any purported general solution of reachability or stability would lead to the impossible situation of a solution to the halting problem. There are, however, at least two ways to proceed. The first is to limit the hybrid systems one wants to consider. The second is to use general conditions for, say, stability of hybrid systems as *sufficient* conditions for design, i.e., design controls so that the system is easy to analyze.

The first approach has been taken by the computer scientists (as mentioned above). The second approach has had success in control theory. For example, even though it is in general, notoriously hard to find Lyapunov functions to prove stability. Yet, most stability proofs in adaptive control treat Lyapunov stability as a sufficient condition and design stable adaptive controllers by first choosing a Lyapunov function and then controls that enforce its decay.

In the case of hybrid systems, the analysis tools I have developed [10, 11] in general deal not only with the behavior of the constituent systems, but also impose conditions on the behavior of the system at switching times. Thus, control design arising from the constraints of [11] is a topic of further research.

There are also theoretical issues to be explored. Some examples include the stability of systems with multiple equilibrium points, the stability of switched systems, relations between fixed-point theory and Lyapunov stability, and the stability and dynamics of ordinary differential equations driven by Markov chains whose transition probabilities are a function of the continuous state. The latter may provide a link to the large literature on jump systems.

Another important topic of further research is to incorporate developed analysis tools into software engineering tools. This will allow application of these tools to complicated examples in a timely manner.

Control. Specific open theoretical issues were discussed in [14]. Another has to do with the robustness of our hybrid controls with respect to state. Here, the transversality assumptions made in [14] should combine with Tavernini's result on continuity with respect initial condition to yield continuity of control laws on an open dense set.

Algorithms for synthesizing optimal hybrid controllers have also been given [12, 17], along with some academic examples. An important area of current research is to develop good computational schemes to compute near-optimal controls in realistic cases. Analysis of rates of convergence of discretized algorithms should be explored. Later, the development of software tools to design such controllers automatically will become an important area of research.

Design. Finally, from modeling, through analysis and control, we come to design of complex, hybrid systems. Here, some of the interaction between levels is under our jurisdiction. What would we do with such freedom, coupled with

our new-found analysis and control techniques? For example, we might design a flexible manufacturing system that not only allows quick changes between different product lines, but allows manufacture of new ones with relative ease.

Consider the so-called reflex controllers of [37, 44], which constitute a dynamically consistent interface between low-level servo control and higher-level planning algorithms that ensures obstacle avoidance. Thus as a step in the direction of hierarchical, hybrid design, the reflex control concept is an example of how to incorporate a new control module to allow rapid, dynamically transparent design of higher-level programs. Further, there are some structures for the control programs used in research aircraft [28] that may lend themselves to such an approach. In each case, these designs incorporate structures which allow engineers to separate the continuous and logical worlds.

These controllers provide the inspiration, our analysis and control results the foundation, and our steps toward efficient algorithms the impetus, for setting a course toward design of hybrid systems. Ultimately, it is hoped they will lead to truly intelligent engineering systems.

References

- [1] R. Alur *et al.*. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In Grossman *et al.* [25], pp. 209–229.
- [2] P. Anstaklis *et al.*, eds. *Hybrid Systems II*. Springer, New York, 1995.
- [3] P. J. Antsaklis, J. A. Stiver, and M. D. Lemmon. Hybrid system modeling and autonomous control systems. In Grossman *et al.* [25], pp. 366–392.
- [4] A. Back, J. Guckenheimer, and M. Myers. A dynamical simulation facility for hybrid systems. In Grossman *et al.* [25], pp. 255–267.
- [5] D. Bainov and P. Simeonov. *Systems with Impulse Effect*. Ellis Horwood, 1989.
- [6] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers. *Bulletin American Math. Society*, 21(1):1–46, July 1989.
- [7] M. S. Branicky. Equivalence of analog and digital computation. In *Workshop on Continuous Algorithms and Complexity*, Barcelona, October 1993. Abstract.
- [8] M. S. Branicky. Topology of hybrid systems. In *Proc. IEEE Conf. Decision and Control*, pp. 2309–2314, San Antonio, December 1993.
- [9] M. S. Branicky. Analog computation with continuous ODEs. In *Proc. IEEE Workshop Physics and Computation*, pp. 265–274, Dallas, November 1994.
- [10] M. S. Branicky. Analyzing continuous switching systems: Theory and examples. In *Proc. American Control Conf.*, pp. 3110–3114, Baltimore, June 1994.
- [11] M. S. Branicky. Stability of switched and hybrid systems. In *Proc. IEEE Conf. Decision and Control*, pp. 3498–3503, Lake Buena Vista, December 1994.
- [12] M. S. Branicky. *Studies in Hybrid Systems: Modeling, Analysis, and Control*. PhD thesis, Massachusetts Institute of Technology, EECS Dept., June 1995.
- [13] M. S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Comp. Science*, 138(1):67–100, 1995.
- [14] M. S. Branicky, V. Borkar, and S. K. Mitter. A unified framework for hybrid control. Technical Report LIDS-P-2239, Lab. for Information and Decision Systems, Massachusetts Institute of Technology, April 1994. Also see: [15, 16].
- [15] M. S. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control. In G. Cohen and J.-P. Quadrat, eds., *Proc. 11th INRIA Intl. Conf. Analysis and Optimization of Systems*, vol. 199 of *Lecture Notes Control and Info. Sciences*, pp. 352–358, New York, 1994. Springer. Extended Abstract.
- [16] M. S. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control. In *Proc. IEEE Conf. Decision and Control*, pp. 4228–4234, Lake Buena Vista, December 1994.
- [17] M. S. Branicky and S. K. Mitter. Algorithms for optimal hybrid control. In *Proc. IEEE Conf. Decision and Control*, pp. 2661–2666, New Orleans, December 1995.

- [18] R. W. Brockett. Smooth dynamical systems which realize arithmetical and logical operations. In H. Nijmeijer and J. M. Schumacher, eds., *Three Decades of Mathematical Systems Theory*, pp. 19–30. Springer, Berlin, 1989.
- [19] R. W. Brockett. Hybrid models for motion control systems. In H. L. Trentelman and J. C. Willems, eds., *Essays in Control*, pp. 29–53. Birkhäuser, Boston, 1993.
- [20] C. Chase, J. Serrano, and P. J. Ramadge. Periodicity and chaos from switched flow systems. *IEEE Trans. Automatic Control*, 38(1):70–83, 1993.
- [21] H.-F. Chen. Stochastic approximation and its new applications. In *Proc. Hong Kong Intl. Workshop New Directions of Control and Manuf.*, pp. 2–12, 1994.
- [22] D. F. Delchamps. Stabilizing a linear system with quantized state feedback. *IEEE Trans. Automatic Control*, 35(8):916–924, August 1990.
- [23] A. Deshpande. *Control of Hybrid Systems*. PhD thesis, U. Cal. Berkeley, 1994.
- [24] S. Gennaro, C. Horn, S. Kulkarni, P. Ramadge. Reduction of timed hybrid systems. In *Proc. IEEE Conf. Decision and Cont.*, pp. 4215–4220, December 1994.
- [25] R. L. Grossman et al., eds. *Hybrid Systems*. Springer, New York, 1993.
- [26] D. Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231–274, 1987.
- [27] C. S. Hsu. *Cell-to-Cell Mapping*. Springer, New York, 1987.
- [28] C. S. Hynes. Flight control systems that can be formally verified. Technical report, NASA Ames Research Center, 1993. Extended abstract.
- [29] R. Koplon and E. Sontag. Sign-linear systems as cascades of automata and continuous variable systems. In *Proc. IEEE Conf. Decision and Control*, pp. 2290–2291, San Antonio, December 1993.
- [30] S. R. Kulkarni and P. J. Ramadge. On the existence and memory requirements of convergent on-line decision rules. In *Proc. IEEE Conf. Decision and Control*, pp. 3022–3027, New Orleans, December 1995.
- [31] G. A. Lafferriere. Discontinuous stabilizing feedback using partially defined Lyapunov functions. In *Proc. IEEE Conf. Decision and Control*, pp. 3487–3491, Lake Buena Vista, December 1994.
- [32] A. Megretsky. Robustness analysis of a relay system. Lab. for Information and Decision Systems Seminar, Massachusetts Institute of Technology, April 1995.
- [33] G. Meyer. Design of flight vehicle management systems. In *Proc. IEEE Conf. Decision and Control*, Lake Buena Vista, December 1994. Plenary Lecture.
- [34] A. S. Morse. Supervisory control of families of linear set-point controllers—Part 1: Exact matching, Part 2: Robustness. Preprints, March 1993, November 1994.
- [35] A. S. Morse. Control using logic-based switching. In A. Isidori, ed., *Trends in Control*, pp. 69–113. Springer, Berlin, 1995.
- [36] A. Nerode and W. Kohn. Models for hybrid systems: Automata, topologies, stability. In Grossman et al. [25], pp. 317–356.
- [37] W. S. Newman and M. S. Branicky. Experiments in reflex control for industrial manipulators. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, May 1990.
- [38] P. Peleties and R. DeCarlo. Asymptotic stability of m -switched systems using Lyapunov-like fncs. *Proc. Amer. Cont. Conf.*, pp. 1679–1684, Boston, June 1991.
- [39] P. J. Ramadge. On the periodicity of symbolic observations of piecewise smooth discrete-time systems. *IEEE Trans. Automatic Control*, 35(7):807–813, 1990.
- [40] L. J. Serrano. *The Effects of Time Sampling and Quantization in the Discrete Control of Continuous Systems*. PhD thesis, Princeton, EE Dept., October 1990.
- [41] K. S. Sibirsky. *Intro. to Topological Dynamics*. Noordhoff, Netherlands, 1975.
- [42] E. D. Sontag. *Mathematical Control Theory*. Springer, New York, 1990.
- [43] L. Tavernini. Differential automata and their discrete simulators. *Nonlinear Analysis, Theory, Methods, and Applications*, 11(6):665–683, 1987.
- [44] T. S. Wickman, M. S. Branicky, and W. S. Newman. Reflex control for robot system preservation, reliability, and autonomy. *Computers and Electrical Engineering*, 20(5):391–407, 1994. Special Issue on Fault Tolerance and Robotics.
- [45] M. A. Wicks, P. Peleties, and R. A. DeCarlo. Construction of piecewise Lyapunov functions for stabilizing switched systems. In *Proc. IEEE Conf. Decision and Control*, pp. 3492–3497, Lake Buena Vista, December 1994.
- [46] H. S. Witsenhausen. A class of hybrid-state continuous-time dynamic systems. *IEEE Trans. Automatic Control*, 11(2):161–167, 1966.