A Game Theoretic Approach to Controller Design for Hybrid Systems

CLAIRE J. TOMLIN, MEMBER, IEEE, JOHN LYGEROS, MEMBER, IEEE, AND S. SHANKAR SASTRY, FELLOW, IEEE

Invited Paper

We present a method to design controllers for safety specifications in hybrid systems. The hybrid system combines discrete event dynamics with nonlinear continuous dynamics: the discrete event dynamics model linguistic and qualitative information and naturally accommodate mode switching logic, and the continuous dynamics model the physical processes themselves, such as the continuous response of an aircraft to the forces of aileron and throttle. Input variables model both continuous and discrete control and disturbance parameters. We translate safety specifications into restrictions on the system's reachable sets of states. Then, using analysis based on optimal control and game theory for automata and continuous dynamical systems, we derive Hamilton–Jacobi equations whose solutions describe the boundaries of reachable sets. These equations are the heart of our general controller synthesis technique for hybrid systems, in which we calculate feedback control laws for the continuous and discrete variables, which guarantee that the hybrid system remains in the "safe subset" of the reachable set. We discuss issues related to computing solutions to Hamilton–Jacobi equations. Throughout, we demonstrate our techniques on examples of hybrid automata modeling aircraft conflict resolution, autopilot flight mode switching, and vehicle collision avoidance.

Keywords*—*Aircraft control, air-traffic control, automated highways, game theory, hybrid automata, optimal control.

Manuscript received November 5, 1999; revised April 17, 2000. The work of C. J. Tomlin was supported by the DARPA Software Enabled Control (SEC) Program under AFRL Contract F33615-99-C-3014 and by a Frederick E. Terman Faculty Award. The work of J. Lygeros and S. S. Sastry was supported by the DARPA SEC program under AFRL Contract F33615-98-C-3614 and by ARO under Grant DAAH04-96-1-0341.

C. J. Tomlin is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305-4035 USA (e-mail: tomlin@stanford.edu).

J. Lygeros is with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K. (e-mail: jl290@club.eng.cam.ac.uk).

S. S. Sastry is with the Information Technology Office, Defense Advanced Projects Research Agency, Arlington, VA 22203-1714 USA (e-mail: ssastry@darpa.mil).

Publisher Item Identifier S 0018-9219(00)06459-8.

I. INTRODUCTION

A. Why Study Hybrid Systems?

For about the past eight years, researchers in the traditionally distinct fields of control theory and computer science verification have proposed models and verification and controller synthesis techniques for complex, safety critical systems. The area of *hybrid systems* is loosely defined as the study of systems that involve the interaction of discrete event and continuous time dynamics, with the purpose of proving properties such as reachability and stability.

To elaborate, consider that individual feedback control scenarios are naturally modeled as interconnections of modules characterized by their input/output behavior. Modal control, by contrast, naturally suggests a state-based view, with states representing control modes. These distinct modeling techniques need to be reconciled in order to support a systematic methodology for the design, validation, and implementation of control software. The dichotomy between the input/output (feedback) view and the state-space (multimodal) view is often presented in a restricted setting, as a difference between continuous and discrete control. Continuous feedback control focuses on the analog interaction of the controller with a physical plant, through sensors and actuators. Continuous control models and design techniques have been developed, used, and validated extensively. The case for discrete multimodal control rests on the observations that discrete abstractions make it easier to manage system complexity, discrete models are easier to manipulate, and discrete representations more naturally accommodate linguistic and qualitative information in controller design. Commonly used models for hybrid systems, such as hybrid automata, combine state-transition diagrams for discrete behavior with differential equations or inclusions for continuous behavior. The discrete event systems are used to model modes of operation of the system, such as the mode of flight of an aircraft or the interaction and coordination

among several aircraft. The continuous dynamics model the physical process, such as the continuous response of an aircraft to the forces of aileron and throttle. For complex multiagent systems, in addition to the requirement of designing hierarchies of decision making at different levels of abstraction, there is the need to synchronize decision making across different agents. In the absence of a global clock, it is useful to have the mechanism of synchronization using discrete events. The resulting interplay of continuous single agent dynamics with synchronization across multiple agents once again results in a hybrid system.

B. Problems Solved Using Hybrid Systems

In work to date, a number of problems for hybrid systems have been studied.

1) Optimal Control: Roughly speaking, the optimal control problem is to drive the system to a desirable state while minimizing a cost function that depends on the path followed. It typically involves a *terminal cost* (depending on the terminal state), an *integral cost* accumulated along continuous evolution, and a series of *jump costs* associated with discrete transitions. This is a classical problem for continuous systems, extended more recently to discrete systems [1], and to classes of hybrid systems with simple continuous dynamics [2]. The approach has been extended to general hybrid systems both for the dynamic programming formulation [3] and for the variational formulation, extending the maximum principle [4].

2) Hierarchical Control: This describes the systematic decomposition of control tasks such that the resulting hierarchical controller guarantees a certain performance [5], [6].

3) Distributed, Multiagent Control: Here, optimal control problems are decomposed so that they can be solved in a distributed way by a collection of *agents* with a specified communication and information architecture [7].

4) Least Restrictive Controllers for Specifications Such as Safety and Liveness: Here it is required that all trajectories of the system satisfy certain properties. Properties include safety properties (for example, requiring that the state of the system remain in a certain safe set) and liveness properties (requiring that the state eventually enter a certain target set or visit a set infinitely often). For discrete systems, this problem has a long history in mathematics and computer science. The essence of the classical problem was posed by Church [8] and solved in different ways by a number of authors, including Büchi and Landweber [9] (for an overview, please see [10]). In the continuous domain, control problems of the safety type have been addressed in the context of pursuit evasion games [11].

In this paper, we concentrate on the solution of safety specifications for hybrid systems that have rich classes of nonlinear dynamics. We encode system safety properties into requirements that the state trajectory of the system remain within certain safe subsets of the state space. We then calculate the subset of states from which this safe subset is always reachable, and determine the control law, in both the discrete and continuous control variables, that renders this subset invariant. We present three examples to illustrate our model and control law design methodology: aircraft conflict resolution, aerodynamic envelope protection, and highway vehicle collision avoidance. These examples, introduced below, illustrate the ability of a hybrid system framework to improve the ease of analysis and control of complex safety critical systems.

C. High-Confidence Systems

We increasingly find ourselves surrounded by so-called *high-confidence systems*: transportation networks, power networks, communication networks. These are systems in which the real-time software is expected to work at a very high level of confidence: of necessity is the reliability, correctness, and graceful degradation under faulted modes of operation. These systems are safety critical since failures could result in loss of life and/or property; they are hybrid due to the multiagent hierarchical nature of the control system involved. Two key examples in the area of transportation systems have motivated our work: air-traffic management systems [12] and automated highway systems [13].

Today's crowded skies and ever-increasing demand for air travel, coupled with new technologies for navigation and surveillance, are fueling a change in the way that the Federal Aviation Administration manages air traffic. Current air-traffic control (ATC) practice manually routes aircraft along predefined paths between "fixes," using radar track and flight information from plan view displays and voice communication over radio channels. The use of global positioning systems (GPSs) and datalink communication will enable automation of some ATC functionality, such as the prediction and resolution of trajectory conflicts between aircraft. For such a safety-critical system, the integrity and acceptance of new automated control functionality depends on a *provably safe* design, which requires accurate system models, and procedures for verifying and synthesizing safe control actions. For more details, we refer the reader to [14]–[16]. A proposed new solution to the growing congestion is a program called "free" or "flexible" flight, in which each aircraft flies along optimal user-preferred routes, which can minimize flight time and fuel consumption or avoid inclement weather. Key enabling technologies for such a system are accurate methods for navigation and communication [such as inertial navigation systems (INSs), GPSs, and automatic dependence surveillance-broadcast (ADS-B) datalinks], provably safe methods for conflict detection and resolution, route generation and regeneration, and automatic flight mode switching to follow routes. In such a system, each aircraft is surrounded by a virtual cylinder called a protected zone, the radius and height of which (2.5 nautical miles by 1000 ft) depend on current International Civil Aviation Organization (ICAO) separation standards. A conflict or loss of separation between aircraft occurs when protected zones of two or more aircraft overlap. The conflict resolution algorithm must use available information to generate maneuvers that resolve conflicts as they are predicted. From a data base of flight modes, such as segments of constant heading, of constant bank angle, of constant or varying airspeed, the conflict resolution

algorithm could synthesize the parameters of the maneuver, such as the proper sequencing of these modes, the numerical values associated to each segment (heading angle, bank angle, airspeed), and the conditions for switching between flight modes. The result would be a maneuver, proven to be safe within the limits of the models used, which is a familiar sequence of commands easily executable by the flight management systems on board aircraft. The resulting maneuvers could be viewed as protocols, or "rules of the road."

Highway traffic congestion is a problem millions of commuters face every day. Even though building new highways seems like an easy solution, the price of real estate in and around urban areas makes it impractical. An alternative solution that has attracted attention in recent years uses automation to make more efficient use of the current highway system. Intelligent vehicle highway systems (IVHSs) attempt to do this by taking advantage of recent technological advances in communication, sensing, surveillance, computation, and control. The most ambitious form of IVHS is the automated highway system (AHS), in which driving is partially or even fully automated. Different AHS concepts have been proposed, ranging from longitudinal (along the lane) autonomous intelligent cruise controllers (AICCs) to fully automated driving. The platooning concept [17] is based on the empirical observation that low relative velocity collisions are safe for both the vehicles and their passengers. On an AHS that supports platooning, vehicles move in tightly spaced groups (known as *platoons*) of up to 20 vehicles, with *intraplatoon* spacings of the order of 1 to 2 m. Under normal conditions of operation, the controllers of the vehicles can be designed such that no collisions occur within a platoon. Under emergency conditions, collisions may be possible. However, because of the tight spacing, it is likely that they will be at low relative velocities. Collisions are prevented from propagating from one platoon to the next by maintaining a large *interplatoon* spacing (on the order of 50 m). Because it promises a substantial increase in highway throughput, platooning has been studied extensively in recent years. As with the air-traffic system, control design techniques that guarantee safety of the system are paramount. Controllers have been proposed for maintaining the longitudinal stability of a platoon [18], for joining and splitting platoons and maintaining the interplatoon separation [13], for regulating the lateral movement of the vehicles (lane keeping and lane changing), for coordinating the actions of different platoons [17], for stabilizing the traffic in segments of the highway, and for routing traffic along the entire highway system. The interaction among these controllers involves hybrid phenomena at different levels. For example, the discrete communication protocols that coordinate the actions of neighboring platoons [17] implement their decisions by invoking continuous controllers designed for joining platoons, splitting platoons, and other maneuvers. Moreover, even these low-level controllers may involve switching, between the different modes used for maintaining a desired speed and heading from the preceding vehicle [19], for example.

D. Game Theoretic Approach to Hybrid Systems Design

The analysis and control of hybrid systems can be based on game-theoretic methods from computer science and optimal control. A hybrid game is a multiplayer structure in which the players have both discrete and continuous moves. Each player controls a set of real-valued variables. The game proceeds in a sequence of rounds. In every round, each player either chooses to update some of its variables (a discrete move), or chooses a law according to which its variables will evolve together with an upper bound on the duration of the round (a continuous move). If some player chooses a discrete move, then the variables are updated and no time elapses. If all players choose continuous moves, then the variables evolve according to the selected laws for the minimum of the selected durations.

Hybrid games have been used both in the computer science and in the control community. In the computer science literature, they have been classified with respect to the complexity of the laws that govern the evolution of the variables and with respect to the winning conditions for the players. This has been studied in the timed games of Maler *et al.* [20], [21] (for constant differential equations of the form $\dot{x} = c$) and the rectangular games of Henzinger *et al.* [22], [23] (constant differential inclusions of the form $c \leq \dot{x} \leq d$). The classical winning conditions for infinite discrete games are safety (stay within a given set of states), Büchi (visit a given set of states infinitely often), and Boolean combinations thereof. In the control community, problems of the safety type have been addressed in the context of pursuit-evasion games and robust control [11], [24], [25].

As the reader will see in this paper, the solution of safety games for hybrid automata involves the fixed-point iteration of single-round controllability operators: the game theoretic synthesis procedure is semidecidable when certain operators called Pre_1, Pre_2, $Reach$ that we define in the paper are computable. We discuss how to approximate the solution of the exact Pre_1, Pre_2, $Reach$ operators in order to cover cases that are not decidable. It is our conviction that this theory is critical for all examples of practical importance.

E. Outline of the Paper

In Section II, we give the formal definition of the class of hybrid systems, called hybrid automata, that we will study. Section III contains three examples drawn from air-traffic management, flight systems avionics design, and automated highway systems, which we carry throughout the paper. In Section IV, we give a review of the discrete, continuous, and hybrid controller design procedure. This procedure is applied to the three model examples in Section V. Approximate solution techniques are discussed in Section VI.

II. Modeling Formalism

Our goal is to develop a mathematical model of hybrid systems rich enough to describe both the evolution of continuous dynamics and the discrete switching logic, and capable of modeling uncertainty in both the continuous and discrete input variables. In this section, we present a hybrid

system model that was developed in [16], [26], and [27] and is based on overlaying finite automata on nonlinear continuous-time control systems. To get the ideas fixed, we start with finite-state automata and continuous state, continuous time control systems.

A. Notation

Let W be a countable collection of variables and let $\mathbf{W}$ denote its set of valuations, that is, the set of all possible assignments of the variables in W. We refer to variables whose set of valuations is countable as *discrete* and to variables whose set of valuations is a subset of a Euclidean space $\mathbb{R}^n$ as *continuous*. We assume that Euclidean space is given the Euclidean metric topology, whereas countable and finite sets are given the discrete topology (all subsets are open). Subsets of a topological space are given the subset topology and products of topological spaces are given the product topology. For a subset U of a topological space we use $\overline{U}$ to denote its closure, U^o its interior, ∂U its boundary, U^c its complement, $|U|$ its cardinality, 2^U the set of all subsets of U, U^ω the set of finite or infinite sequences of elements in U, and $\mathcal{U}$ the set of piecewise continuous functions from $\mathbb{R}$ to U. We use $\wedge$ to denote conjunction, $\vee$ disjunction, $\neg$ negation, $\forall$ the universal quantifier, and $\exists$ the existential quantifier.

A finite-state automaton is represented as

$$(Q, \Sigma, Init, R) \tag{1}$$

where Q is a finite set of discrete state variables; $\Sigma = \Sigma_1 \cup \Sigma_2$ is a finite set of discrete input variables, where Σ_1 contains the controller's inputs and Σ_2 contains the environment's inputs, which cannot be controlled; $Init \subseteq \mathbf{Q}$ is a set of initial states; and R: $\mathbf{Q} \times \Sigma \to 2^{\mathbf{Q}}$ maps the state and input space to subsets of the state space and thus describes the transition logic of the finite automaton. An *execution* of (1) is defined to be a finite or infinite sequence of states and inputs $(q(\cdot), \sigma(\cdot)) \in \mathbf{Q}^\omega \times \Sigma^\omega$ where, for $i \in \mathbb{Z}$, $q(0) \in Init$ and $q(i+1) \in R(q(i), \sigma(i))$.

Continuous state, continuous time control systems, on the other hand, may be represented as differential equations evolving on a state space X

$$\dot{x} = f(x, v) \tag{2}$$

where $x \in \mathbf{X}$ is the state, usually $\mathbf{X} = \mathbb{R}^n$; $v \in \mathbf{V} = \mathbf{U} \times \mathbf{D}$ is the space of continuous input variables, where $\mathbf{U} = \mathbb{R}^u$ is the set of control inputs and $\mathbf{D} = \mathbb{R}^d$ is the set of disturbance inputs; f is a vector field, assumed to be globally Lipschitz in x and continuous in v; and the initial state $x(0) \in Init$ where $Init \subseteq \mathbf{X}$. A trajectory of (2) over an interval $[\tau, \tau'] \subseteq \mathbb{R}$ is a map: $(x(\cdot), v(\cdot)) : [\tau, \tau'] \to \mathbf{X} \times \mathbf{V}$ such that $\dot{x}(t) = f(x(t), v(t))$ for all $t \in [\tau, \tau']$.

B. Hybrid Automata

Since we are interested in hybrid phenomena that involve both continuous and discrete dynamics, we introduce the hybrid time trajectory, which will encode the set of times over which the evolution of the system is defined.

Definition 1 (Hybrid Time Trajectory): A hybrid time trajectory $\tau = \{I_i\}_{i=0}^N$ is a finite or infinite sequence of intervals of the real line, such that:

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

The interpretation is that τ_i are the times at which discrete transitions take place. Notice that discrete transitions are assumed to be instantaneous and that multiple discrete transitions may take place at the same time, since it is possible for $\tau_i = \tau_{i+1}$. Hybrid time trajectories can extend to "infinity" if τ is an infinite sequence or if it is a finite sequence ending with an interval of the form $[\tau_N, \infty)$. Since the dynamical systems we consider are time invariant, we assume, without loss of generality, that $\tau_0 = 0$. We denote by $\mathcal{T}$ the set of all hybrid time trajectories. For $t \in \mathbb{R}$ and $\tau \in \mathcal{T}$, we use $t \in \tau$ as a shorthand notation for "there exists a j such that $t \in [\tau_j, \tau_j'] \in \tau$." We mention that in this paper, the evolution of time will be "dense" continuous time, that is, the underlying continuous state dynamics are continuous time. In applications, it is sometimes of interest to have discrete time (synchronous) evolution of the continuous state with automata like transitions (asynchronous).

Definition 2 (Hybrid Automaton): A hybrid automaton H is a collection

$$H = (Q, X, \Sigma, V, \text{Init}, f, \text{Inv}, R) \tag{3}$$

where

- $Q \cup X$ is a finite collection of state variables, with $\mathbf{Q}$ finite and $\mathbf{X} = \mathbb{R}^n$;
- $\Sigma = \Sigma_1 \cup \Sigma_2$ is a finite collection of discrete input variables, where Σ_1 is the set of discrete control inputs, and Σ_2 is the set of discrete disturbance inputs;
- $V = U \cup D$ is the set of continuous input variables, where U is the set of continuous control inputs and D is the set of continuous disturbance inputs;
- Init $\subseteq \mathbf{Q} \times \mathbf{X}$ is a set of initial states;
- f: $\mathbf{Q} \times \mathbf{X} \times \mathbf{V} \to \mathbf{X}$ is a vector field describing the evolution of x for each $q \in Q$; f is assumed to be globally Lipschitz in X (for fixed $q \in Q$) and continuous in V;
- Inv $\subseteq \mathbf{Q} \times \mathbf{X} \times \Sigma \times \mathbf{V}$ is called an invariant and defines combinations of states and inputs for which continuous evolution is allowed;
- R: $\mathbf{Q} \times \mathbf{X} \times \Sigma \times \mathbf{V} \to 2^{\mathbf{Q} \times \mathbf{X}}$ is a reset relation, which encodes the discrete transitions of the hybrid automaton.

We refer to $(q, x) \in \mathbf{Q} \times \mathbf{X}$ as the *state* of H and to $(\sigma, v) \in \Sigma \times \mathbf{V}$ as the *input* of H. We make the following assumption to ensure that the hybrid automaton does not *block* trajectories, causing the system to deadlock: assume that Inv is an open set, and that if $(q, x, \sigma, v) \notin Inv$ then $R(q, x, \sigma, v) \neq \emptyset$.

The main differences between the model presented here and that of timed and linear hybrid automata are in the continuous dynamics: we incorporate full nonlinear models of the

continuous state dynamics and include continuous input variables to model both parameters that the designer may control and disturbance parameters that the designer must control against. This allows an accurate representation of the continuous physical processes that we would like to model and control.

Definition 3 (Execution of a Hybrid Automaton): An *execution* of a hybrid automaton H is a hybrid trajectory $\chi = (\tau, q, x, \sigma, v)$ with:

- *Initial Condition*: $(q(\tau_0), x(\tau_0)) \in$ Init;
- *Continuous Evolution*: for all i with $\tau_i < \tau_i'$, $q(\cdot)$, $\sigma(\cdot)$ are constant, $v(\cdot)$ is piecewise continuous, $x(\cdot)$ is a solution to the differential equation $\dot{x} = f(q, x, v)$ over $[\tau_i, \tau_i']$, and for all $t \in [\tau_i, \tau_i')$, $(q(t), x(t), \sigma(t), v(t)) \in$ Inv;
- *Discrete Evolution*: for all i, $(q(\tau_{i+1}), x(\tau_{i+1})) \in R(q(\tau_i'), x(\tau_i'), \sigma(\tau_i'), v(\tau_i'))$.

A hybrid automaton is interpreted as accepting, rather than generating, an execution. Hybrid automata may accept no executions for some initial states or some inputs, may accept multiple executions for the same initial state and inputs, or may not accept executions over arbitrarily long time horizons. More formally, an execution $\chi = (\tau, q, x, \sigma, v)$ is called *finite* if τ is a finite sequence ending with a closed interval, *infinite* if τ is an infinite sequence or if $\sum_{i=0}^{\infty}(\tau_i' - \tau_i) = \infty$, and *Zeno*[1] if it is infinite but $\sum_{i=0}^{\infty}(\tau_i' - \tau_i) < \infty$. In [28] and [29], conditions are given that allow one to ensure that a hybrid automaton accepts a unique infinite, non-Zeno execution.

Associated to the hybrid automaton H is a *trajectory acceptance condition*, which describes the specification that one would like executions of the system to satisfy. We define a *property* as a map from the set of executions to {True, False}. Our work has been motivated by verification and synthesis for safety critical applications, and as such we have been primarily interested in safety specifications. These specifications are encoded as subsets of the state space of the hybrid system: the *safe set* $F \subseteq \mathbf{Q} \times \mathbf{X}$ is that subset in which the system is defined to be safe. We assume that safe sets are closed and unsafe sets are open; we use F to denote safe sets and $G^o = F^c$ to denote unsafe sets. We define a *safety property*, denoted by $\Box F$, by

$$\Box F(\chi) = \begin{cases} \text{True}, & \text{if } \forall t \in \tau,\ (q(t), x(t)) \in F \\ \text{False}, & \text{otherwise.} \end{cases}$$

Safety properties are more general than they may initially appear. Consider, for example, another property $\Diamond G$ defined by

$$\Diamond G(\chi) = \begin{cases} \text{True}, & \text{if } \exists t \in \tau,\ (q(t), x(t)) \in G \\ \text{False}, & \text{otherwise.} \end{cases}$$

It is easy to see that $\Diamond G(\chi) = \neg\Box(G^c)(\chi)$. Examples of more complex specifications not covered by safety properties are so-called *liveness properties*, for example, the "always eventually" property $\Box\Diamond F(\chi)$. The full Borel hierarchy of specifications built up from $\Box$, $\Diamond$ constitutes an important set of temporal properties (see [30]).

In what follows, we will restrict ourselves to static-state feedback controllers. We define a *static-state feedback controller* for a hybrid automaton H to be a map from the state space to subsets of the controller's input space

$$g\colon \mathbf{Q} \times \mathbf{X} \to 2^{\Sigma_1 \times \mathbf{U}}. \tag{4}$$

Thus, the controller may affect the behavior of H though its discrete and continuous control inputs $\sigma_1 \in \Sigma_1$ and $u \in \mathbf{U}$.

III. Motivating Examples

We now present three examples of hybrid systems: resolution of trajectory conflicts between aircraft, single aircraft aerodynamic envelope protection, and collision avoidance for automated vehicles in an AHS. In the conflict resolution and collision avoidance problems, the system is safe if the aircraft or vehicles always maintain *minimum separation* with each other. In the aerodynamic envelope protection problem (representative of autopilot design problems), system safety means that the state of the aircraft remains within minimum and maximum bounds imposed on its velocities and orientation variables.

A. Aircraft Conflict Resolution

We present as motivating example a model for the kinematic motions of two aircraft, labeled 1 and 2, at a fixed altitude. Let $(x_r, y_r, \psi_r) \in \mathbb{R}^2 \times [-\pi, \pi)$ represent the relative position and orientation of aircraft 2 with respect to aircraft 1. In terms of the absolute positions and orientations of the two aircraft x_i, y_i, ψ_i for $i = 1, 2$, it may be verified that $x_r = \cos\psi_1(x_2 - x_1) + \sin\psi_1(y_2 - y_1)$, $y_r = -\sin\psi_1(x_2 - x_1) + \cos\psi_1(y_2 - y_1)$, $\psi_r = \psi_2 - \psi_1$ and it is easy to derive that

$$\begin{aligned} \dot{x}_r &= -v_1 + v_2\cos\psi_r + \omega_1 y_r \\ \dot{y}_r &= v_2\sin\psi_r - \omega_1 x_r \\ \dot{\psi}_r &= \omega_2 - \omega_1 \end{aligned} \tag{5}$$

where v_i is the linear velocity of aircraft i and ω_i is its angular velocity. The protected zone of aircraft 2 may be translated to the origin of this relative frame, and thus the relative position (x_r, y_r) must remain outside of the disk $\{(x_r, y_r, \psi_r): x_r^2 + y_r^2 < 5^2\}$. The flight modes for this system of two aircraft are based on the linear and angular velocities of the aircraft. We consider two possibilities: $\omega_i = 0$, meaning that aircraft i follows a straight line, and $\omega_i = 1$, meaning that aircraft i follows an arc of a circle if v_i is kept constant. These maneuvers approximate closely the behavior of pilots flying aircraft: straight line segments (constant heading) and arcs of circles (constant bank angle) are easy to fly both manually and on autopilot. Consider a maneuver in which there are three modes in sequence: a *cruise* mode in which both aircraft follow a straight path; an *avoid* mode in which both aircraft follow a circular arc path; and a second *cruise* mode in which the aircraft return to the straight path. The protocol

[1]The name "Zeno" comes from the ancient Greek philosopher Zeno who lived in Elea, a Greek colony in southern Italy, in the fifth century B.C. Zeno spent his time posing paradoxes about time.

of the maneuver is that as soon as the aircraft are within a certain distance of each other, each aircraft turns 90° to its right and follows a half-circle. Once the half-circle is complete, each aircraft returns to its original heading and continues on its straight path (Fig. 1). We assume that both aircraft switch modes simultaneously, so that the relative orientation ψ_r is constant, and we assume that both aircraft fly an arc with the same radius at the same velocity. These assumptions simply allow us to display the evolution of the continuous state in two dimensions, making the results easier to present: in a true conflict resolution scenario, these assumptions would be removed. This maneuver generalizes to n-aircraft as a "roundabout" maneuver, discussed in [12].

The dynamics of the maneuver can be encoded by the hybrid automaton of Fig. 2, where q_1 corresponds to cruising before the avoid maneuver, q_2 corresponds to the avoid mode, and q_3 corresponds to cruising after the avoid maneuver has been completed. There is one discrete control input σ_1, such that the switch from $\sigma_1 = 0$ to $\sigma_1 = 1$ triggers the transition from q_1 to q_2. The transition from q_2 to q_3 is required to take place after the aircraft have completed a half-circle: note that with $\omega_i = 1$, for $i = 1, 2$, it takes π time units to complete a half circle. The continuous state space is augmented with a timer $z \in \mathbb{R}^+$ to force this transition. Let $x = (x_r, y_r, \psi_r, z)^T$. At each transition, both aircraft change heading instantaneously by $\pi/2$ radians; we represent this with the standard rotation matrix $Rot(\pi/2)$. Assuming computation in the flight management system of aircraft 1, we assume that v_1 is controllable, and v_2 is known to within some uncertainty. Safety is defined in terms of the relative distance between the two aircraft

$$G = \{q_1, q_2, q_3\} \times \{x \in \mathbf{X}: x_r^2 + y_r^2 \leq 5^2\}. \qquad (6)$$

Thus the state space of this two-aircraft system is $\mathbf{Q} \times \mathbf{X} = \{q_1, q_2, q_3\} \times (\mathbb{R}^2 \times [-\pi, \pi) \times \mathbb{R}^+)$. The discrete input space is $\Sigma = \Sigma_1 = \{0, 1\}$ ($\Sigma_2 = \emptyset$), and the continuous input space is $\mathbf{V} = \mathbf{U} \times \mathbf{D}$, where $\mathbf{U} = \{v_1\}$ and $\mathbf{D} = \{v_2\}$ (we assume in this example that v_1 and v_2 are fixed, the more general case is presented in [12], [16]). We assume $Init = q_1 \times (G^\circ)^c$, that f is described by the relative aircraft dynamics (5) augmented with a timer, as shown in Fig. 2, and that Inv is given as follows:

$$\text{Inv} = (q_1, \mathbf{X}, 0, \mathbf{V}) \cup (q_2, \{x \in \mathbf{X} | 0 \leq z \leq \pi\}, \Sigma, \mathbf{V}) \cup (q_3, \mathbf{X}, \Sigma, \mathbf{V}).$$

The map R that resets x_r, y_r in transitions from q_1 to q_2 and q_2 to q_3 is described in Fig. 2. The controller synthesis problem is therefore to generate the relative distance between aircraft at which the aircraft may switch safely from mode 1 to mode 2, and the minimum turning radius R in mode 2, to ensure that the five-nautical-mile separation is maintained.

B. Aerodynamic Envelope Protection

The example is inspired by the work of [31], in which the flight modes for the airspeed and flight path angle dynamics of an aircraft are derived. We consider a nonlinear model

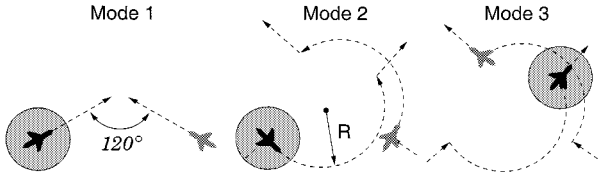

Fig. 1. Two aircraft in three modes of operation: in modes 1 and 3 the aircraft follow a straight course and in mode 2 the aircraft follow a half-circle. The initial relative heading (120°) is preserved throughout.

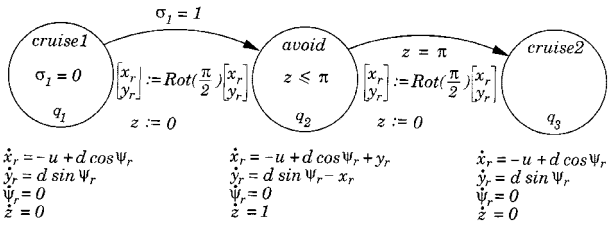

Fig. 2. In q_1 both aircraft follow a straight course, in q_2 a half-circle, and in q_3 both aircraft return to a straight course.

of the longitudinal axis dynamics of a conventional takeoff and landing (CTOL) aircraft in normal aerodynamic flight in still air [32], [33], shown in Fig. 3. The horizontal and vertical axes are, respectively, the $(x_{\text{inertial}}, h_{\text{inertial}})$ (denoted x, h) axes, and the *pitch angle* θ is the angle made by the aircraft body axis, x_{body} with the x axis. The *flight path angle* γ and the *angle of attack* α are defined as: $\gamma = \tan^{-1}(\dot{h}/\dot{x})$, $\alpha = \theta - \gamma$. Expressions for the lift (L) and drag (D) forces are given by $L = a_L(\dot{x}^2 + \dot{h}^2)(1 + c\alpha)$, $D = a_D(\dot{x}^2 + \dot{h}^2)(1 + b(1 + c\alpha)^2)$, where a_L, a_D are dimensionless *lift* and *drag coefficients* and b and c are positive constants. We assume that the autopilot has direct control over both the forward thrust T (throttle) and the aircraft pitch θ (elevators); thus there are two continuous control inputs $(u_1, u_2) = (T, \theta)$. Physical considerations impose constraints on the inputs: $u \in [T_{\min}, T_{\max}] \times [\theta_{\min}, \theta_{\max}]$. The longitudinal dynamics may be modeled by the Newton–Euler equations

$$M \begin{bmatrix} \ddot{x} \\ \ddot{h} \end{bmatrix} = \text{Rot}(\theta) \left[\text{Rot}^T(\alpha) \begin{bmatrix} -D \\ L \end{bmatrix} + \begin{bmatrix} T \\ 0 \end{bmatrix} \right] + \begin{bmatrix} 0 \\ -Mg \end{bmatrix} \qquad (7)$$

where Rot(α) and Rot(θ) are standard 2×2 rotation matrices, M is the mass of the aircraft, and g is gravitational acceleration. The simplified flight management system (FMS) studied in this paper uses control inputs T and θ to control combinations of the speed $V = \sqrt{\dot{x}^2 + \dot{h}^2}$, flight path angle γ, and altitude h. The linear and angular accelerations $(\dot{V}, V\dot{\gamma})$ may be derived directly from (7)

$$\dot{V} = -\frac{D}{M} - g \sin\gamma + \frac{T}{M} \cos\alpha \qquad (8)$$

$$V\dot{\gamma} = \frac{L}{M} - g \cos\gamma + \frac{T}{M} \sin\alpha. \qquad (9)$$

Note that these dynamics are expressed solely in terms of (V, γ) and inputs (T, θ), where $\alpha = \theta - \gamma$; thus (8) and (9) are a convenient way to represent the dynamics for modes in which h is not a controlled variable. Safety regulations for the

aircraft dictate that V, γ, and h must remain within specified limits

$$V_{\min} \le V \le V_{\max} \quad \gamma_{\min} \le \gamma \le \gamma_{\max} \quad h_{\min} \le h \le h_{\max} \tag{10}$$

where $V_{\min}, V_{\max}, \gamma_{\min}, \gamma_{\max}, h_{\min}, h_{\max}$ are functions of such factors as airspace regulations, type of aircraft, and weather. For aircraft flying at cruise altitude, we assume that these limits are constants, and thus the aerodynamic flight envelope F is as illustrated in Fig. 4 as projections in the (V, γ)-space and $(h, V, \dot{h})$-space, where $\dot{h} = V \sin\gamma$. The state trajectory must remain within F at all times within cruise mode (this is called aerodynamic envelope protection). The system may be discretized into five flight modes, depending on the state variables being controlled:

- **Mode 1**: (Speed, Flight Path), in which the thrust T is between its specified operating limits ($T_{\min} < T < T_{\max}$), the control inputs are T and θ, and the controlled states are the speed and the flight path angle of the aircraft $(V, \gamma)^T$;
- **Mode 2**: (Speed), in which the thrust saturates ($T = T_{\min} \vee T = T_{\max}$) and thus is no longer available as a control input; the only input is θ, and the only controlled state is V;
- **Mode 3**: (Flight Path), in which the thrust saturates ($T = T_{\min} \vee T = T_{\max}$); the input is again θ, and the controlled state is γ;
- **Mode 4**: (Speed, Altitude), in which the thrust T is between its specified operating limits ($T_{\min} < T < T_{\max}$), the control inputs are T and θ, and the controlled states are the speed and the vertical position of the aircraft $(V, h)^T$;
- **Mode 5**: (Altitude), in which the thrust saturates ($T = T_{\min} \vee T = T_{\max}$); the input is θ, and the controlled state is h.

Modeling this system as a hybrid automaton, the discrete state may take on one of five possible values, $\mathbf{Q} = \{q_1, \cdots, q_5\}$, corresponding to the five flight modes. The continuous state of the system is $x = (x, \dot{x}, h, \dot{h})^T \in \mathbf{X} = \mathbb{R}^4$, with continuous dynamics specified by (7). The control inputs are the throttle T and pitch θ with input constraint set $\mathbf{U} = [T_{\min}, T_{\max}] \times [\theta_{\min}, \theta_{\max}]$, and we assume for simplicity that there are no continuous disturbance inputs ($D = \emptyset$) (a possible extension to this problem would be to consider wind as a continuous disturbance). The controllable discrete inputs label transitions from each mode to every other mode: let σ_1^{ij}, for $i \in \{1, \cdots, 5\}$ and $j \in \{1, \cdots, 5\}$, be the action labeling the transition from q_i to q_j. We assume that there are no disturbance actions ($\Sigma_2 = \emptyset$) (although it will be a very nice extension to introduce disturbance actions representing pilot error in manually switching modes). The safe set F is illustrated in Fig. 4. In our calculations, we use parameter values corresponding to a DC-8 at cruising speed; the details are described in [16] and [34]. The controller synthesis problem is therefore to generate the continuous control inputs (T, θ) to use in each flight mode, as well as the allowable mode transitions, so that flight envelope protection is guaranteed.

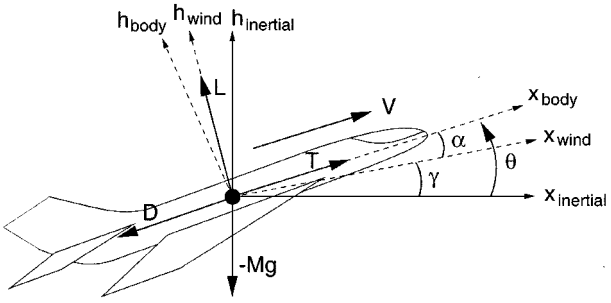

Fig. 3. The longitudinal dynamics of a conventional take-off and landing (CTOL) aircraft in flight with attached axes about its center of mass.

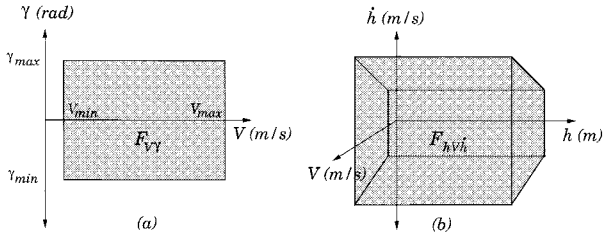

Fig. 4. (a) Simplified aerodynamic flight envelope in (V, γ)-space: axes are airspeed V, flight path angle γ. (b) Simplified aerodynamic flight envelope in $(h, V, \dot{h})$-space: axes are altitude h, airspeed V, vertical speed $\dot{h}$.

C. Vehicle Collision Avoidance

The need to ensure the safety of the vehicles on an AHS dictates that formal methods have to be used to design and analyze the hybrid interactions. In [13], the design methodology presented in this paper was used to derive safety conditions for the longitudinal movement of the vehicles in a multilane AHS. Here we highlight a simple example from that study. Consider two platoons, labeled A and B, moving on an AHS (Fig. 5) with A following B. Let L_i denote the length of platoon $i = A, B$ and x_i its position from a fixed road-side reference frame. Since neither the dynamics nor the safety requirements depend on the absolute position of the platoons, we introduce a variable $D_{AB} = x_B - x_A - L_B$ to keep track of the spacing between platoons A and B. We assume that (after feedback linearization) the controller of vehicle A can directly affect the acceleration of A, $u = \ddot{x}_Z$, through brake and throttle actuators. We also assume that vehicle A is equipped with sensors to measure its own velocity and the spacing and relative velocity with respect to vehicle B. The acceleration of vehicle B, $\ddot{x}_B$, is assumed to be unknown to vehicle A and is treated as a disturbance. The continuous dynamics can now be described by a state vector $x = [x_1\ x_2\ x_3]^T = [\dot{x}_A\ D_{AB}\ \dot{D}_{AB}]^T \in \mathbb{R}^3$ with

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ddot{x}_B \\ &= F_x + G_1 u + G_2 \ddot{x}_B. \end{aligned} \tag{11}$$

Physical considerations impose constraints on u and $\ddot{x}_B$: vehicles are not allowed to move in reverse and are required to keep their speed below a certain speed limit $v^{\max}$. To enforce these requirements, we assume that u and $\ddot{x}_B$ satisfy

$$u \in \begin{cases} [0,\, a_A^{\max}], & \text{if } x_1 \le 0 \\ [a_A^{\min},\, a_A^{\max}], & \text{if } 0 < x_1 < v^{\max} \\ [a_A^{\min},\, 0], & \text{if } x_1 \ge v^{\max} \end{cases}$$

and

$$\ddot{x}_B = \ddot{x}_B \in \begin{cases} [0,\, a_B^{\max}], & \text{if } x_1 + x_3 \le 0 \\ [a_B^{\min},\, a_B^{\max}], & \text{if } 0 < x_1 + x_3 < v^{\max} \\ [a_B^{\min},\, 0], & \text{if } x_1 + x_3 \ge v^{\max}. \end{cases} \quad (12)$$

To ensure that the dynamics of the system are physically meaningful, we assume that the set of initial states is such that $\dot{x}_A(0) = x_1(0) \in [0,\, v^{\max}]$, $\dot{x}_B(0) = x_1(0) + x_3(0) \in [0,\, v^{\max}]$, and that the constants satisfy $a_A^{\min} < 0 < a_A^{\max}$ and $a_B^{\min} < 0 < a_B^{\max}$. In this case, $\dot{x}_A(t),\, \dot{x}_B(t) \in [0,\, v^{\max}]$ for all $t \ge 0$.

Even though the model of this two platoon system seems continuous, there are a number of sources of discrete behavior. The first is the mode switching necessary to enforce the constraints on the velocities. Three discrete states are introduced for each platoon to account for this, one for $\dot{x}_i = 0$, one for $\dot{x}_i \in (0,\, v^{\max})$, and one for $\dot{x}_i = v^{\max}$, $i = A,\, B$. This gives rise to a total of nine discrete states. Additional discrete phenomena are introduced by the intraplatoon collisions that A and B may experience in case of emergency. From the point of view of platoon A, these collisions can be treated as a source of disturbance and can be modeled as discrete events that instantaneously reset the velocities of certain vehicles. For simplicity, we assume that the first vehicle of platoon A (*leader* of A) can experience at most one collision with the vehicle immediately behind it (first *follower* in A), and parameterize the disturbance by the time at which the collision occurs (T_A) and the resulting increase in the velocity of the leader (δv_A). Likewise, we assume that the last vehicle of platoon B can experience at most one collision, and use the time at which the collision occurs (T_B) and the decrease of the velocity of the last vehicle (δv_B) to parameterize the disturbance. Since the vehicles are not allowed to move in reverse, we assume that collisions with a platoon will not result in negative velocities, or, in other words, that $\delta v_B \le x_1(T_B) + x_3(T_B)$. Likewise, since vehicles are not allowed to move faster than the speed limit, it is natural to assume that collisions within a platoon will not result in velocities greater than $v^{\max}$, or, in other words, $\delta v_A \le v^{\max} - x_1(T_A)$. Finally, if the intraplatoon controllers are designed properly, we can assume that all intraplatoon collisions will be at low relative velocities, below a certain "safe" value, $v^S \approx 3$ m/s. Under these assumptions, which are reasonable if the vehicles have roughly equal masses and coefficients of restitution, the discrete disturbance caused by intraplatoon collisions can be parameterized by

$$T_A \ge 0, \quad T_B \ge 0, \quad \delta v_A \in [0,\, \min\{v^S,\, v^{\max} - x_1(T_A)\}],$$
$$\delta v_B \in [0,\, \min\{v^S,\, x_1(T_B) + x_3(T_B)\}]. \quad (13)$$

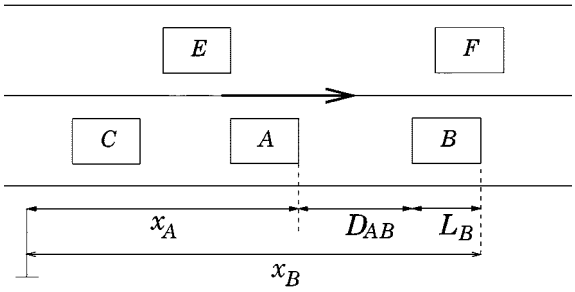

Fig. 5. AHS model with five platoons and distances as marked.

The hybrid automaton used to capture the intraplatoon collisions in platoons A and B is shown in Fig. 6. The discrete states q_0 to q_3 represent: no collisions in either platoon; collision in platoon B; collision in platoon A; and two simultaneous collisions, one in platoon A and one in platoon B. Two discrete disturbance inputs ($Coll_A$ and $Coll_B$) are introduced to trigger the collisions, and four continuous disturbance inputs (T_A, T_B, δv_A, δv_B) are introduced to capture their effect. The discrete states introduced to model the velocity constraints have been suppressed to simplify the figure; with these states, the total number of discrete states is 36. To simplify the notation, we use $d = (\ddot{x}_B,\, T_A,\, T_B,\, \delta v_A,\, \delta v_B)$ to denote the continuous disturbance inputs, even though, strictly speaking, T_A and T_B encode the times at which the discrete disturbance inputs $Coll_A$ and $Coll_B$ change values and δv_A and δv_B are only relevant at those times. It is easy to show that for each initial condition $(q^0,\, x^0)$ and each control u and disturbance d, there exists a unique state trajectory. Moreover, this state trajectory satisfies $\dot{x}_A(t),\, \dot{x}_B(t) \in [0,\, v^{\max}]$ for all $t \ge 0$. Finally, an additional source of discrete dynamics is the communication protocols proposed in [17] to coordinate the actions of neighboring platoons. The methods discussed in this paper can also be used to establish conditions that ensure the safety of the interaction between the discrete communication protocols and the low-level, continuous controllers. This issue will not be addressed here because of the complicated notation needed. The interested reader is referred to [13] for details.

Recall that even though intraplatoon collisions with A and B are acceptable in case of emergency, interplatoon collisions should be avoided at all costs. Thus, for safety, we would like to prevent collisions between platoons A and B. In other words, we would like to ensure that $x_2(t) \ge 0$ for all $t \ge 0$. Notice that the limiting case $x_2(t) = 0$ is considered acceptable, since the vehicles just touch at zero relative velocity. Summarizing, the controller synthesis problem we would like to solve involves selecting the continuous control variable u such that for all actions of the disturbance d the two platoons are guaranteed not to collide.

IV. Reachability Analysis and Controller Design

A state of a dynamical system is defined to be *reachable* if there is an execution of the system that touches it. A subset of the state space is said to be *controlled invariant* if there exists a controller that guarantees that if the execution starts

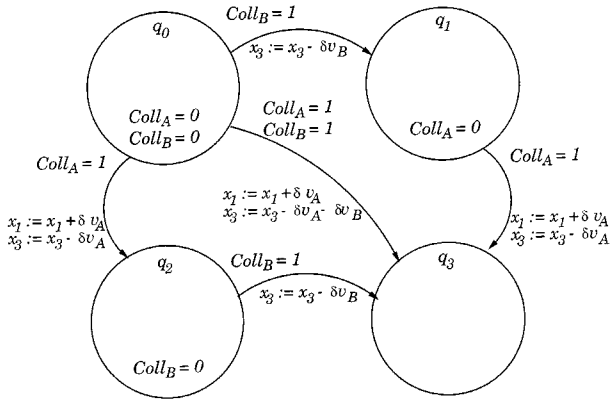

Fig. 6. Hybrid automaton modeling intraplatoon collisions in platoons A and B. The discrete states are q_0 for no collisions, q_1 for collision inside platoon B, q_2 for collision inside platoon A, and q_3 for simultaneous intraplatoon collisions in A and B. The continuous dynamics within each discrete mode are given by (11).

in the subset, the execution stays in the subset for all future time. For a hybrid system H, we seek to design a controller that prunes away executions that reach unsafe states. For such a problem, the initial state or set of initial states is usually left unspecified. We therefore pose the controller synthesis problem as: Given a safe set F, determine i) the maximal controlled invariant set contained in F and ii) the controller which renders this set invariant. In this paper, we restrict ourselves to safety specifications and consequently safety games alone. For more general specifications such as $\Box\Diamond$, $\Diamond\Box$, or others, the games to be considered are referred to as Büchi games and include nested versions of the games that we will discuss here. In this section, we solve the controller synthesis problem for safety specifications in hybrid automata. Our method is based on computing the *backward-reachable* set from F. As before, we present the algorithm first on the finite-state automata and continuous-state control systems.

A. Finite Automata

The problem of synthesizing control laws g: $\mathbf{Q} \to 2^{\Sigma_1}$ in the presence of uncertain actions $\sigma_2(\cdot) \in \Sigma_2^\omega$ for the finite automaton described by (1) was first posed by Church in 1962 [8], who was studying problems in digital circuit design, and was solved by Büchi and Landweber [9] and Rabin [35] in the late 1960s and early 1970s using a version of the von Neumann–Morgenstern discrete game [36]. More recently, Ramadge and Wonham [37] added new insight into the structure of the control law. A temporal logic for modeling such games is introduced in [38]. We define the *winning states* W^* for the controller as the subset of F from which the system has a sequence of control actions $\sigma_1(\cdot)$, which can force the system to remain in F despite the actions of the environment $\sigma_2(\cdot)$. The set W^* can be calculated as the fixed point of the following iteration (where a negative index $i \in \mathbb{Z}_-$ is used to indicate that each step is a predecessor operation).

```
Algorithm 1 (Maximal Controlled
Invariant Set for Finite State Automata)
  initialization: W^0 = F, W^{-1} = ∅, i = 0.
  while W^i ≠ W^{i-1} do
    W^{i-1} = W^i ∩ {q ∈ Q | ∃σ1 ∈ Σ1 ∀σ2 ∈ Σ2
    R(q, σ1, σ2) ⊆ W^i}
    i = i - 1
  end while
```

The iteration terminates when $W^i = W^{i-1} \triangleq W^*$. At each step of the iteration, $W^{i-1} \subseteq W^i$. Since $|\mathbf{Q}|$ is finite the iteration terminates in a finite number of steps. The set W^i contains those states for which the controller has a sequence of actions that will ensure that the system remains in F for at least i steps, for all possible sequences $\sigma_2(\cdot) \in \Sigma_2$. In order to characterize this iteration mathematically, we associate a *value function* $J(q, i)$ to each state at each iteration, representing the future reward or cost to be incurred by the system given that its current state is q and iteration i

$$J(q, i)\colon \mathbf{Q} \times \mathbb{Z}_- \to \{0, 1\} \quad \text{such that}$$
$$J(q, i) = \begin{cases} 1, & q \in W^i \\ 0, & q \in (W^i)^c. \end{cases} \tag{14}$$

Therefore, $W^i = \{q \in \mathbf{Q} | J(q, i) = 1\}$. Since the most logical action of the controller is to keep the system inside F in the face of unknown and therefore possibly hostile actions of the environment

$$\max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} \min_{q' \in R(q, \sigma_1, \sigma_2)} J(q', i)$$
$$= \begin{cases} 1, & \text{if } \exists \sigma_1 \in \Sigma_1 \; \forall \sigma_2 \in \Sigma_2, \; R(q, \sigma_1, \sigma_2) \subseteq W^i \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The "$\min_{q' \in R(q, \sigma_1, \sigma_2)}$" in the above compensates for the nondeterminism in R; the order of operations $\max_{\sigma_1} \min_{\sigma_2}$ means that the controller *plays first*, trying to maximize the minimum value of $J(\cdot)$. This representation gives the environment the advantage, since it has "prior" knowledge of the controller's action when making its own choice. Therefore, in general

$$\max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} \min_{q' \in R(q, \sigma_1, \sigma_2)} J(\cdot)$$
$$\leq \min_{\sigma_2 \in \Sigma_2} \max_{\sigma_1 \in \Sigma_1} \min_{q' \in R(q, \sigma_1, \sigma_2)} J(\cdot) \tag{16}$$

with equality occurring when the action (σ_1, σ_2) is a *saddle solution*, or a *no regret* solution for each player. Here, we do not need to assume the existence of a saddle solution; rather, we always give advantage to the environment, the player doing its worst to drive the system out of F, in order to ensure a conservative solution. Strictly speaking, this is a *Stackelberg solution* of the game with the controller as leader.

The iteration process in Algorithm 1 may be summarized by the difference equation

$$J(q, i-1) - J(q, i)$$
$$= \min\left\{0, \max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} \left[\min_{q' \in R(q, \sigma_1, \sigma_2)} J(q', i) - J(q, i)\right]\right\}. \tag{17}$$

We refer to (17) as a "discrete Hamilton–Jacobi equation." The first "min" in the equation ensures that states outside

W^i that can be forced by the controller to transition into W^i are prevented from appearing in W^{i-1}. This means that once a state has associated to it a value of zero, the value stays at zero for all subsequent iterations: enforcing the requirement that "once a state becomes unsafe, it remains unsafe."

Proposition 1 (Winning States W^*): For finite sets $\mathbf{Q}$ and Σ, a fixed point $J^*(q)$ of (17) is reached in a finite number of steps. The set of winning states for the controller is $W^* = \{q \in \mathbf{Q} | J^*(q) = 1\}$. W^* is the largest controlled invariant subset of F.

B. Continuous-Time Dynamics

For the continuous nonlinear dynamics, described by (2), the solution of an optimal control law $u(\cdot)$ in the presence of environmental uncertainties $d(\cdot)$ was solved as a zero-sum dynamic game by Isaacs in the early 1950s [39].[2] Solutions for linear differential games were presented by Pontrjagin in [40]. An excellent modern reference is [11]. To conform with convention in the dynamical games literature, we represent the specification in terms of the *unsafe* set G: the controller wins if it can keep the system from entering the interior of the set G, denoted $G^\circ = \{x \in \mathbf{X} | l(x) < 0\}$ for a differentiable function l: $\mathbf{X} \to \mathbb{R}$, with boundary ∂G. Conversely, the environment wins if it can drive the system into G°. The winning states for the controller are those states $W^* \subseteq X$ from which there exists a control law $u(\cdot) \in \mathcal{U}$ that can keep the system outside G° despite the disturbance $d(\cdot) \in \mathcal{D}$.

Consider the system over the time interval $[t, 0]$, where $t < 0$. The value function of the game is defined by

$$J: \mathbf{X} \times \mathcal{U} \times \mathcal{D} \times \mathbb{R}_- \to \mathbb{R} \quad \text{with}$$
$$J(x, u(\cdot), d(\cdot), t) = l(x(0)) \tag{18}$$

and is interpreted as the cost of a trajectory $x(\cdot)$ that starts at x at initial time $t \leq 0$, evolves according to (2) with input $(u(\cdot), d(\cdot))$ and ends at the final state $x(0)$, with cost $l(x(0))$. Note that the value function depends only on the final state: there is no running cost, or *Lagrangian*. This is because, for proving safety of the system, we are only interested in whether or not the system trajectory enters G°, and we wish to compute the control law that maximizes the set of initial states from which the system trajectory is guaranteed to remain outside of G°. Thus, we do not restrict the trajectories further with a running cost. The game is won by the environment if the terminal state $x(0)$ is in G° [i.e. $J(x, u(\cdot), d(\cdot), t) < 0$], and is won by the controller otherwise.

The optimal action of the controller is one that tries to maximize the minimum cost, to try to counteract the optimal disturbance action of pushing the system toward G. As in the discrete game, the disturbance is given the advantage: the control $u(\cdot)$ plays first and disturbance $d(\cdot)$ plays second with the knowledge of the controller's play. This kind of solution is referred to as a *Stackelberg solution*; in the event that the $(\max \min)$ solution is equal to the $(\min \max)$ solution, then the solution is also the *saddle solution* of the game. The Stackelberg solution corresponds to $\max_{u(\cdot)\in\mathcal{U}} \min_{d(\cdot)\in\mathcal{D}} J(x, u(\cdot), d(\cdot), t)$, and we define $J^*(x, t)$, the optimal cost, as

$$J^*(x, t) = \max_{u(\cdot)\in\mathcal{U}} \min_{d(\cdot)\in\mathcal{D}} J(x, u(\cdot), d(\cdot), t) \tag{19}$$

and the corresponding optimal input and disturbance as

$$u^*(\cdot) = \arg \max_{u(\cdot)\in\mathcal{U}} \min_{d(\cdot)\in\mathcal{D}} J(x, u(\cdot), d(\cdot), t)$$
$$d^*(\cdot) = \arg \min_{d(\cdot)\in\mathcal{D}} J(x, u^*(\cdot), d(\cdot), t).$$

What is not explicit in this formulation is the "information patterns" used by the input and disturbance. In the event that the input and disturbance choices are causal (i.e., based only on past values of the input, disturbance, and state), the solution to the game can be characterized using Hamilton–Jacobi (Isaacs) theory. More precisely, the Hamiltonian of the system is $H(x, p, u, d) \triangleq p^T f(x, u, d)$, where $p \in \mathbb{R}^n$ is the costate vector [41], [11]. Standard results in optimal control theory [41]–[43], may be extended [11] and [16] to yield the optimal solution

$$u^* = \arg \max_{u\in\mathbf{U}} \min_{d\in\mathbf{D}} H(x, p, u, d) \tag{20}$$
$$d^* = \arg \min_{d\in\mathbf{D}} H(x, p, u^*, d). \tag{21}$$

The *optimal Hamiltonian* is therefore given by

$$H^*(x, p) = \max_{u\in\mathbf{U}} \min_{d\in\mathbf{D}} p^T f(x, u, d). \tag{22}$$

The *Hamilton–Jacobi partial differential equation* for the evolution of the value function J in backward time is derived [41], [16] to be

$$-\frac{\partial J^*(x, t)}{t} = H^*\left(x, \frac{\partial J^*(x, t)}{x}\right) \tag{23}$$

with boundary condition $J^*(x, 0) = l(x)$. However, the solution $J^*(x, t)$ to (23) includes as safe states those states for which optimal trajectories pass through G° and end up outside G° at time 0. To prevent this from happening, we modify (23) to guarantee that, if for some $x \in X$ there exists an $s \in [t, 0]$ such that $J^*(x, s) < 0$, then $J^*(x, t)$ is nonincreasing for time less than s. We do this by requiring that

$$-\frac{\partial J^*(x, t)}{t} = \begin{cases} H^*\left(x, \dfrac{\partial J^*(x, t)}{x}\right) \\ \qquad \text{for } \{x \in \mathbf{X} | J^*(x, t) > 0\} \\ \min\left\{0, H^*\left(x, \dfrac{\partial J^*(x, t)}{x}\right)\right\} \\ \qquad \text{for } \{x \in \mathbf{X} | J^*(x, t) \leq 0\} \end{cases} \tag{24}$$

with boundary condition $J^*(x, 0) = l(x)$. It is easy to show [16] that if $J^*(x, t)$ is a smooth solution to (24), then the subset of the state space enclosed by the zero level set of $J^*(x, t)$ cannot decrease as time marches backward, that is, for all $t_2 \leq t_1 \leq 0$, $\{x \in \mathbf{X} | J^*(x, t_1) \leq 0\} \subseteq \{x \in \mathbf{X} | J^*(x, t_2) \leq 0\}$. Equation (24) is the continuous analog

[2] Isaacs was then a researcher at Rand Corp. and was motivated by tactical issues for U.S. Air Force pilots (dog fights, missile evasion).

to (17) of the preceding discrete game and describes the relationship between the time and state evolution of $J^*(x, t)$. We claim that $\{x \in \mathbf{X} | J^*(x, t) < 0\}$, where $J^*(x, t)$ is the solution to (24), is the set of states from which the environment can force the system into G° in at most $|t|$ seconds. The pictorial explanation of this is given in Fig. 7. The part of the boundary of $J^*(x, 0)$ where $(\partial J^*(x, 0)/\partial t) \leq 0$ does not grow with negative time, as shown by point 1 in Fig. 7(a). Part (b) of the same figure shows the existence of a stationary solution as $t \to -\infty$. Questions about the smoothness of solutions to the Hamilton–Jacobi equation, shocks, and what to make of the solutions in this instance are subtle ones, and the interested reader is referred to Section VI and [16] for details on these points.

Proposition 2 (Winning States W^*): Assume that $J^*(x, t)$ satisfies the Hamilton–Jacobi equation (24) for all t, and that it converges uniformly in x as $t \to -\infty$ to a function $J^*(x)$. Then the set of winning states for the controller is

$$W^* = \{x \in \mathbf{X} | J^*(x) \geq 0\}. \tag{25}$$

W^* is the largest controlled invariant set contained in $F = (G^\circ)^c$.

The *least restrictive* feedback controller for u that renders W^* invariant can now be constructed. The controller $g(x)$ is defined to be

$$g(x) = \begin{cases} \left\{ u \in \mathbf{U}: \min\limits_{d \in \mathbf{D}} \dfrac{\partial J^*(x)}{\partial x} f(x, u, d) \geq 0 \right\}, & \text{if } x \in \partial W^* \\ \mathbf{U}, & \text{if } x \in (W^*)^\circ. \end{cases}$$

Thus, in the interior of W^*, u is free to take on any value in $\mathbf{U}$. Existence of such u for $x \in W^*$ is guaranteed by construction.

C. Hybrid Systems

Consider the nonlinear hybrid automaton (3) with safety property $\Box F$, where $F \subseteq \mathbf{Q} \times \mathbf{X}$. We seek to construct the largest set of states for which the control $(\sigma_1(t), u(t))$ can guarantee that the safety property is met despite the action of the disturbance $(\sigma_2(t), d(t))$. For a given set $K \subseteq Q \times X$, we define the *controllable predecessor* $Pre_1(K)$ and the *uncontrollable predecessor* $Pre_2(K^c)$ by

$$\begin{aligned} Pre_1(K) = \{&(q, x) \in K: \exists (\sigma_1, u) \in \Sigma_1 \times \mathbf{U} \\ &\forall (\sigma_2, d) \in \Sigma_2 \times \mathbf{D} (q, x, \sigma_1, \sigma_2, u, d) \\ &\notin Inv \wedge R(q, x, \sigma_1, \sigma_2, u, d) \subseteq K\} \\ Pre_2(K^c) = \{&(q, x) \in K: \forall (\sigma_1, u) \in \Sigma_1 \times \mathbf{U} \exists (\sigma_2, d) \in \\ &\Sigma_2 \times \mathbf{D} R(q, x, \sigma_1, \sigma_2, u, d) \cap K^c \neq \emptyset\} \\ &\cup K^c. \end{aligned} \tag{26}$$

Therefore $Pre_1(K)$ contains all states in K for which controllable actions (σ_1, u) can force the state to remain in K for at least one step in the discrete evolution. $Pre_2(K^c)$, on the other hand, contains all states in K^c, the complement of

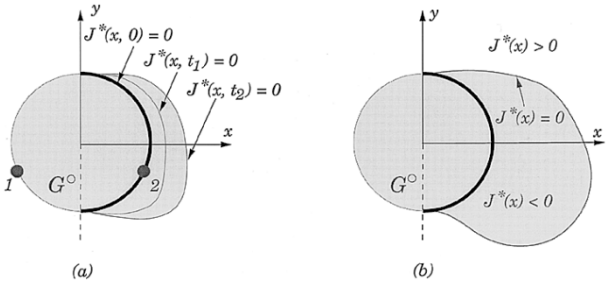

Fig. 7. (a) The sets $\{x \in \mathbf{X} | J^*(x, 0) = 0\}$, $\{x \in \mathbf{X} | J^*(x, t_1) = 0\}$, $\{x \in \mathbf{X} | J^*(x, t_2) = 0\}$ for $0 > t_1 > t_2$. (b) The fixed point $\{x \in \mathbf{X} | J^*(x) < 0\}$, $\{x \in \mathbf{X} | J^*(x) = 0\}$, and $\{x \in \mathbf{X} | J^*(x) > 0\}$.

K, as well as all states from which uncontrollable actions (σ_2, d) may be able to force the state outside of K. In the definition of Pre_1, the controllable actions are required to be able to *force* a transition (hence the Inv in the formula). In contrast, for Pre_2, we simply require that a transition be possible, giving the advantage to the uncontrollable actions. The controllable and uncontrollable predecessors will form the discrete part of the algorithm for computing controlled invariant sets. For the continuous part of the algorithm, we need the $Reach$ operator.

Definition 4 (Reach): Consider two subsets $G \subseteq \mathbf{Q} \times \mathbf{X}$ and $E \subseteq \mathbf{Q} \times \mathbf{X}$ such that $G \cap E = \emptyset$. The Reach operator is defined as

$$\begin{aligned} \text{Reach}(G, E) = \{&(q, x) \in \mathbf{Q} \times \mathbf{X} | \forall u \in \mathcal{U} \exists d \in \mathcal{D} \\ &\text{and } t \geq 0 \quad \text{such that } (q(t), x(t)) \in G \\ &\text{and } (q(s), x(s)) \in \Pi(Inv) \setminus E \\ &\text{for } s \in [0, t]\} \end{aligned} \tag{27}$$

where $(q(s), x(s))$ is the continuous state trajectory of $\dot{x} = f(q(s), x(s), u(s), d(s))$ starting at (q, x) and $\Pi(Inv)$ represents the state space components of Inv. The set Reach(G, E) describes those states from which, for all $u(\cdot) \in \mathcal{U}$, there exists a $d(\cdot) \in \mathcal{D}$, such that the state trajectory $(q(s), x(s))$ can be driven to G while avoiding an "escape" set E.

The following algorithm describes the construction of the maximal controlled invariant set for hybrid systems.

```
Algorithm 2 (Maximal Controlled
Invariant Set for Hybrid Systems)
  initialization: W^0 = F, W^{-1} = ∅, i = 0.
  while W^i ≠ W^{i-1} do
    W^{i-1} = W^i \ Reach(Pre_2((W^i)^c), Pre_1(W^i))
    i = i - 1
  end while
```

In the first step of this algorithm, we remove from F all states from which there is a disturbance $d(\cdot) \in \mathcal{D}$ forcing the system either outside F or to states from which an environment action $\sigma_2 \in \Sigma_2$ may cause transitions outside F, without first touching the set of states from which there is a control action $\sigma_1 \in \Sigma_1$ keeping the system inside F. Since

at each step, $W^{i-1} \subseteq W^i$, the set W^i decreases monotonically as i decreases. If the algorithm terminates, we denote the fixed point as W^*.

Proposition 3 (Winning States W^*): If the algorithm terminates after a finite number of steps, the fixed point W^* is the maximal controlled invariant subset of F.

In order to implement this algorithm, we need to calculate Pre_1, Pre_2, and $Reach$. The computation of Pre_1 and Pre_2 requires inversion of the transition relation R subject to the quantifiers $\exists$ and $\forall$; existence of this inverse can be guaranteed subject to well-understood conditions on the map R. The computation of $Reach$ requires the development of a new algorithm for determining the set of initial conditions from which trajectories can reach one set, avoiding a second set along the way. In the following analysis, we describe this calculation for a single discrete state q.

Recall that along continuous evolution the value of the discrete state remains constant. Therefore, since the computation of the Reach operator involves only continuous evolution, it can be carried out for each discrete state separately. Fix the value of $q \in \mathbf{Q}$ and let l_G: $\mathbf{X} \to \mathbb{R}$ and l_E: $\mathbf{X} \to \mathbb{R}$ be differentiable functions such that $G \triangleq \{x \in \mathbf{X}: l_G(x) \leq 0\}$ and $E \triangleq \{x \in \mathbf{X}: l_E(x) \leq 0\}$. Consider the following system of interconnected Hamilton–Jacobi equations:

$$-\frac{\partial J_G^*(x,t)}{t} = \begin{cases} H_G^*\left(x, \dfrac{\partial J_G^*(x,t)}{x}\right), \\ \quad \text{for } \{x \in \mathbf{X} | J_G^*(x,t) > 0\} \\ \min\left\{0, H_G^*\left(x, \dfrac{\partial J_G^*(x,t)}{x}\right)\right\}, \\ \quad \text{for } \{x \in \mathbf{X} | J_G^*(x,t) \leq 0\} \end{cases} \tag{28}$$

and

$$-\frac{\partial J_E^*(x,t)}{t} = \begin{cases} H_E^*\left(x, \dfrac{\partial J_E^*(x,t)}{\partial x}\right), \\ \quad \text{for } \{x \in \mathbf{X} | J_E^*(x,t) > 0\} \\ \min\left\{0, H_E^*\left(x, \dfrac{\partial J_E^*(x,t)}{\partial x}\right)\right\}, \\ \quad \text{for } \{x \in \mathbf{X} | J_E^*(x,t) \leq 0\} \end{cases} \tag{29}$$

where $J_G(x, u(\cdot), d(\cdot), 0) = l_G(x)$ and $J_E(x, u(\cdot), d(\cdot), 0) = l_E(x)$, and

$$H_G^*\left(x, \frac{\partial J_G^*}{\partial x}\right) = \begin{cases} 0, \\ \quad \text{for } \{x \in \mathbf{X} \quad | J_E^*(x,t) \leq 0\} \\ \max\limits_{u \in \mathbf{U}} \min\limits_{d \in \mathbf{D}} \dfrac{\partial J_G^*}{\partial x} f(x,u,d), \\ \quad \text{otherwise} \end{cases} \tag{30}$$

$$H_E^*\left(x, \frac{\partial J_E^*}{\partial x}\right) = \begin{cases} 0, \\ \quad \text{for } \{x \in \mathbf{X} | J_G^*(x,t) \leq 0\} \\ \min\limits_{u \in \mathbf{U}} \max\limits_{d \in \mathbf{D}} \dfrac{\partial J_E^*}{\partial x} f(x,u,d), \\ \quad \text{otherwise.} \end{cases} \tag{31}$$

Equation (28) describes the evolution of the set G under the Hamiltonian H_G^* (30). This is the "$\max_u \min_d$" game of the previous section, with the modification that $H_G^* = 0$ in $\{x \in \mathbf{X} | J_E^*(x,t) \leq 0\}$, which ensures that the evolution of $J_G^*(x,t)$ is frozen in this set. Similarly, (29) describes the evolution of the set E under the Hamiltonian H_E^*. Here a "$\min_u \max_d$" is used, since it is assumed that the control tries to push the system into E, to escape from G. $H_E^* = 0$ in $\{x \in \mathbf{X} | J_G^*(x,t) \leq 0\}$ to ensure that the evolution of $J_E^*(x,t)$ is frozen in this set. Note that in both games, the disturbance is given the advantage by assuming that the control plays first. Fig. 8 illustrates a sample evolution.

It is proven in [16] that the resulting set $\{x \in \mathbf{X} | J_G^*(x,t) < 0\}$ contains neither E nor states for which there is a control $u(\cdot) \in \mathcal{U}$ that drives the system into E; and the set $\{x \in \mathbf{X} | J_E^*(x,t) < 0\}$ contains neither G nor states for which there is a disturbance input $d(\cdot) \in \mathcal{D}$ that drives the system into G. Our theorem states that $\{x \in \mathbf{X} | J_G^*(x,t) < 0\}$ is the set Reach(G, E) [16].

Theorem 1 (Characterization of Reach): Assume that $J_G^*(x,t)$ [$J_E^*(x,t)$, respectively] is a smooth function of x and t, that it satisfies the Hamilton–Jacobi equation (28) [(29), respectively], and that it converges uniformly in x as $t \to -\infty$ to a function $J_G^*(x)$ [$J_E^*(x)$, respectively]. Then

$$\text{Reach}(G, E) = \{x \in \mathbf{X} | J_G^*(x) < 0\}. \tag{32}$$

The *least restrictive* controller that renders W^* invariant is:

$$g(q,x) = \begin{cases} \{(\sigma_1, u) \in \Sigma_1 \times \mathbf{U}: \\ \quad \forall (\sigma_2, d) \in \Sigma_2 \times \mathbf{D}\, R(q,x,\sigma,v) \subseteq W^*\}, \\ \quad \text{if } (q,x) \in (W^*)^\circ \\ \{(\sigma_1, u) \in \Sigma_1 \times \mathbf{U}: \\ \quad \forall (\sigma_2, d) \in \Sigma_2 \times \mathbf{D} \dfrac{\partial J_G^*(x)}{\partial x} f(q,x,u,d) \\ \quad \geq 0 \wedge (q,x,\sigma,v) \in Inv \\ \quad \vee R(q,x,\sigma,v) \subseteq W^* \\ \quad \wedge (q,x,\sigma,v) \notin Inv\}, \\ \quad \text{if } x \in \partial W^* \\ \Sigma_1 \times \mathbf{U}, \\ \quad \text{if } x \in (W^*)^c. \end{cases} \tag{33}$$

D. Remarks

In general, one cannot expect to solve for W^* using a finite computation. The class of hybrid systems for which algorithms like the one presented here are guaranteed to terminate is known to be restricted [44]. In general, Algorithm 2 is semidecidable when the operators Pre_1, Pre_2, $Reach$ are computable. For example, when the continuous state dynamics are constant and the guards and resets are polyhedra, then the operators Pre_1, Pre_2, $Reach$ map polyhedral sets back into polyhedral sets. These hybrid systems are referred to as *linear hybrid automata*. When the hybrid system is

a timed automaton, the synthesis procedure is actually decidable [45]. The main reason for the somewhat pessimistic news about the decidability of the controller synthesis algorithm has to do with the fact that at heart these algorithms involve quantifier elimination for entry into "bad" sets or steering around "good" sets. However, thanks to some recent activity in mathematical logic in what are known as O-minimal systems, one can extend the class of systems for which the synthesis algorithm is semidecidable, to "O-minimal hybrid systems" (see [46]).

However, our main focus in the rest of this paper is to show how one can make progress in getting approximate solutions even when the given application does not belong to a general class of hybrid systems for which the algorithm is semidecidable. In practice, we are helped by the fact that we are usually interested in finite time computations, rather than computing for $t \to -\infty$ or until a fixed point is reached. Numerical techniques are discussed in Section VI.

Another problem is the requirement that the controller resulting from our algorithm be *non-Zeno* (does not enforce the safety requirement by preventing time from diverging). The algorithm proposed here has no way of preventing such behavior, as will be illustrated in the third example, which we solve in the next chapter. There are several ways of removing Zeno behavior. One that we discuss in the next section is a practical method of resolving the Zeno effect, by adding a requirement that the system must remain in each discrete state for a nonzero amount of time. For a further discussion of how to regularize hybrid systems that have Zeno behavior, and to classify Zeno behaviors, see [47] and [48].

V. Solutions to the Examples

In this section, we apply our techniques to the three examples previously introduced. For each example, we first derive and solve the Hamilton–Jacobi equation, and then apply the controller synthesis algorithm to compute the maximal controlled invariant set and corresponding control law so that each system satisfies its specified safety requirement. For these examples, the Hamilton–Jacobi equations are simple enough, and the dimensions of the discrete and continuous state spaces small enough, to permit solutions using the method of characteristics. We discuss computational issues for larger systems in Section VI.

A. Aircraft Conflict Resolution

Consider the three-mode conflict resolution example pictured in Fig. 1 and modeled in Section III-A. We assume that for this example, the speeds (v_1, v_2) of both aircraft are constant even in the straight modes, so that the input and disturbance sets are singletons $(\mathbf{U} = v_1, \mathbf{D} = v_2)$ and $u^* = v_1, d^* = v_2$. The general case, in which $\mathbf{U}$ and $\mathbf{D}$ are ranges of possible speeds, is considered in the examples in [12] and [16]. Recall that our goal is to calculate the relative distance at which the system may safely switch from mode 1 to mode 2, and the minimum turning radius R in mode 2, to ensure that separation between aircraft is maintained. The evolution of the protected zone in each mode, assuming no switches, is computed using the continuous-time Hamilton–Jacobi method. The unsafe set G is defined as $G = \{q_1, q_2, q_3\} \times \{x \in \mathbf{X} |\ l(x) \leq 0\}$, where $l(x) = x_r^2 + y_r^2 - 5^2$. Let $G_i = (q_i, \{x \in \mathbf{X} |\ l(x) \leq 0\})$ represent the unsafe set in mode i. Thus the set $\{x \in \mathbf{X} |\ J^*_{G_i}(x) \leq 0\}$, where $J^*_{G_i}$ is the optimal cost, is the backward evolution of the protected zone in mode i, assuming no switches between modes. These sets are shown in Fig. 9. In both cases, the relative heading between aircraft is assumed fixed at $\psi_r = 2\pi/3$ (because of our assumption that aircraft switch modes instantaneously). We implement Algorithm 2 for this example, at each step computing the sets Pre_1, Pre_2, and $Reach(Pre_2, Pre_1)$. In the first step, $W^0 = F \triangleq G^c$, the complement of G

$$\begin{aligned} W^0 = &((q_1, \{x \in \mathbf{X} |\ l(x) \leq 0\}^c \cap \{x \in \mathbf{X} |\ z = 0\}) \\ &\cup (q_2, \{x \in \mathbf{X} |\ l(x) \leq 0\}^c) \\ &\cup (q_3, \{x \in \mathbf{X} \mid l(x) \leq 0\}^c \cap \{x \in \mathbf{X} |\ z = 0\})) \end{aligned} \tag{34}$$

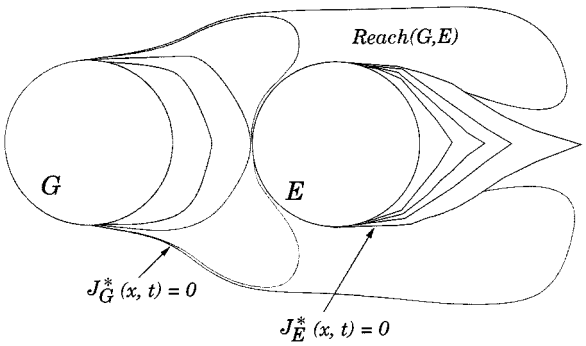

Fig. 8. Computation of $Reach(G, E)$ in a single discrete state q.

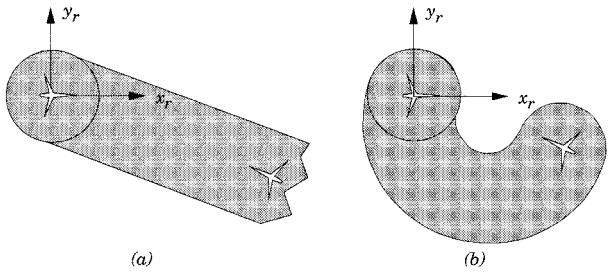

Fig. 9. $J^*_{G_i}(x) \leq 0$ for (a) Modes 1 and 3 ($i = 1, 3$), $\omega_1 = \omega_2 = 0$ (the jagged edge means the set extends infinitely) and (b) Mode 2 ($i = 2$), $\omega_1 = \omega_2 = 1$. In both cases, $\psi_r = 2\pi/3$, and $v_1 = v_2 = 5$.

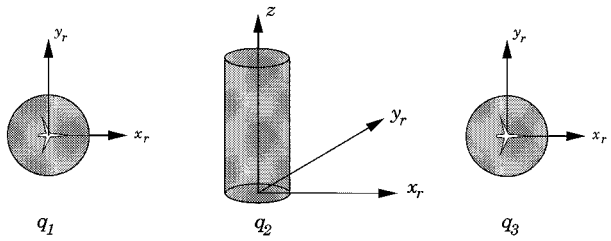

Fig. 10. $(W^0)^c$.

as shown in Fig. 10 (the complement is shown in the figure)

$$Pre_1(W^0) = (q_1, \{x \in \mathbf{X} |\ l(x) \le 0\}^c \cap \{x \in \mathbf{X} |\ z = 0\}) \quad (35)$$

$$Pre_2((W^0)^c) = G. \quad (36)$$

Note that $Pre_1(W^i) \subseteq \{(q_1, \mathbf{X})\}$ for all i, since σ_1 labels transitions from q_1. The set W^{-1} (Fig. 11) is

$$W^{-1} = W^0 \backslash Reach(Pre_2((W^0)^c), Pre_1(W^0)). \quad (37)$$

The set W^{-2} involves computing $Reach(Pre_2((W^{-1})^c), Pre_1(W^{-1}))$; this computation is illustrated in Fig. 12(a), and the set is shown in Fig. 12(b) as the shaded region. Continuing, a fixed point is reached after three iterations: Fig. 13 illustrates this fixed point $W^* = W^{-3}$ in q_1. Since we assumed in this example that the continuous control input $u = v_1$ is fixed, we need only design the discrete part of the controller σ_1 and the radius of the maneuver R. The design is as illustrated in Fig. 13(a): the enabling and forcing of σ_1 occurs at the boundary of W^* as shown, as explained below. The transition from q_1 to q_2, governed by σ_1, must be disabled until the relative position of the two aircraft reach the dashed line as shown; otherwise, the aircraft will lose separation with each other either during the maneuver or after the maneuver is complete. At the dashed line, σ_1 is enabled, meaning the transition from q_1 to q_2 may occur at any time. σ_1 remains enabled until the dynamics reach the solid line (boundary of W^*), at which point it must be both enabled and forced: otherwise the aircraft lose separation immediately. Note that there are states (x_r, y_r) that are not rendered safe by the maneuver. Indeed, if the initial state is in the darker shaded region shown in Fig. 13(a), then the aircraft are doomed to collide. Fig. 13(b) displays the result of increasing the radius of the turn in q_2. Notice that the set W^* (the complement of the shaded region) increases as the turning radius increases. This implies that the maneuver renders a larger subset of the state space safe. Fig. 13(b) shows the critical value of the turning radius, for which the maneuver is guaranteed to be safe, provided the conflict is detected early enough. Thus, the controller synthesis procedure presented in Section IV, applied to this example, generates conditions for the enabling and forcing of σ_1, and also the turning radius R.

B. Aerodynamic Envelope Protection

Consider the longitudinal dynamics of the CTOL aircraft (7) in which the state $\mathbf{x} = (x, \dot{x}, h, \dot{h})^T$ is required to stay in the envelope F, shown in Fig. 4(a) in (V, γ)-space and Fig. 4(b) in $hV\dot{h}$-space. In contrast to the previous example, this example has a range of possible continuous input variables: $\mathbf{U} = [T_{\min}, T_{\max}] \times [\theta_{\min}, \theta_{\max}]$, and thus we will exemplify the continuous Hamilton–Jacobi calculation of Section IV in some detail below.

The specification may be decoupled according to $F_{V\gamma}$ and $F_{hV\dot{h}}$: the airspeed V and flight path angle γ must remain in the envelope $F_{V\gamma}$ at all times; and the airspeed, altitude h, and vertical speed $\dot{h}$ must remain in the envelope $F_{hV\dot{h}}$ at all times. In the speed and flight path modes (modes 1, 2, 3), V and γ are the only controlled variables. Therefore, we may derive the maximal controlled invariant set contained in $F_{V\gamma}$, using the (V, γ)-dynamics (8), (9). Let

$$F_{V\gamma} = \{(V, \gamma) \mid \forall i \in \{1, 2, 3, 4\},\ l_i(V, \gamma) \ge 0\} \quad (38)$$

where

$$l_1(V, \gamma) = V - V_{\min}$$
$$l_2(V, \gamma) = -\gamma + \gamma_{\max}$$
$$l_3(V, \gamma) = -V + V_{\max}$$
$$l_4(V, \gamma) = \gamma - \gamma_{\min}.$$

Thus, $\partial F_{V\gamma}$ is only piecewise smooth, yet for this example we can prove that the calculation can be performed one edge of the boundary at a time: we can derive a Hamilton–Jacobi equation for each l_i, and prove that the intersection of the resulting sets is the maximal controlled invariant subset of $F_{V\gamma}$. The subscript i in each J_i, H_i will indicate that the calculation is for boundary l_i. In the following, we describe how the computation is performed by looking at one edge of the boundary l_1. The details of the proofs of controlled invariance are presented in [16].

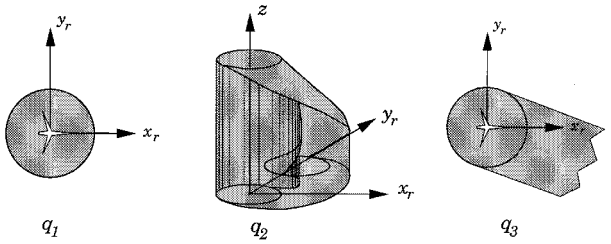

Fig. 11. $(W^{-1})^c$. The jagged edge in q_3 means that the set extends infinitely.

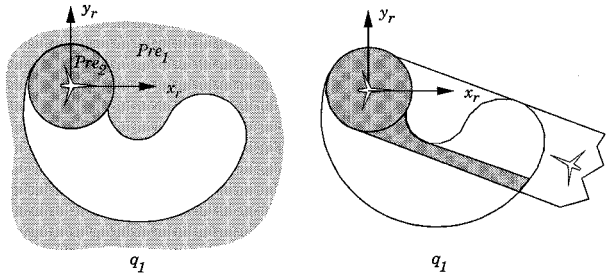

Fig. 12. (a) $Pre_1(W^{-1})$ and $Pre_2(W^{-1})$ in q_1; (b) $Reach(Pre_2(W^{-1}), Pre_1(W^{-1}))$ in q_1.

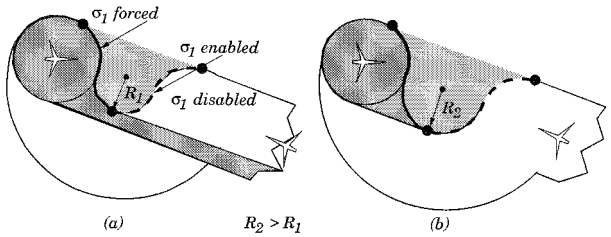

Fig. 13. Showing the enabling and forcing boundaries for σ_1 in state q_1; and the result of increasing the radius of the turn in the avoid maneuver to increase W^*.

The optimal Hamiltonian in this case (there is no disturbance, hence this is not a game but an optimal control problem) is given by the following, where we have substituted into the dynamics the expressions for the lift L and drag D forces (neglecting the quadratic term in D):

$$\begin{aligned} H_1^*((V,\gamma),p) &= \max_{u\in\mathbf{U}}\left[p_1\left(-\frac{a_D V^2}{M} - g\sin\gamma + \frac{1}{M}T\right)\right. \\ &\quad \left. + p_2\left(\frac{a_L V(1-c\gamma)}{M} - \frac{g\cos\gamma}{V} + \frac{a_L c V}{M}\theta\right)\right] \end{aligned} \tag{39}$$

where $p = (p_1, p_2) \in \mathbb{R}^2$. The Hamilton–Jacobi equation describing the evolution of $J_1^*((V,\gamma),t)$ is obtained from (24)

$$-\frac{\partial J_1^*(x,t)}{\partial t} = \begin{cases} H_1^*\left((V,\gamma), \dfrac{\partial J_1^*((V,\gamma),t)}{\partial (V,\gamma)}\right), \\ \qquad \text{for } \{(V,\gamma)\in\mathbf{X} |\ J_1^*((V,\gamma),t) > 0\} \\ \min\left\{0, H_1^*((V,\gamma), \dfrac{\partial J_1^*((V,\gamma),t)}{\partial (V,\gamma)})\right\}, \\ \qquad \text{for } \{(V,\gamma)\in\mathbf{X} |\ J_1^*((V,\gamma),t) \le 0\} \end{cases} \tag{40}$$

with boundary condition $J_1^*((V,\gamma),0) = l_1((V,\gamma))$.

The optimal control at $t = 0$ is computed from (39). The optimal throttle input T may be calculated directly from this equation: $u_1^*(0) = T_{\max}$ (since $p_1 > 0$ for the inward pointing normal). The optimal pitch input $u_2^* = \theta_{\min}$ is calculated indirectly [16], since $H_1^*((V,\gamma),p)$ loses dependence on u_2 on the set $\{(V,\gamma) |\ l_1(V,\gamma) = 0\}$. Define $(V_{\min},\gamma_a) = \{(V,\gamma) |\ l_1(V,\gamma) = 0 \cap H_1^*(V,\gamma) = 0\}$. Then

$$\gamma_a = \sin^{-1}\left(\frac{T_{\max}}{Mg} - \frac{a_D V_{\min}^2}{Mg}\right). \tag{41}$$

Integrate the system dynamics (8), (9) with $(V(0),\gamma(0)) = (V_{\min},\gamma_a)$, $u = (u_1^*, u_2^*)$, backward from $t = 0$ to $t = -T$, where T is chosen to be large enough so that the solution intersects $\{(V,\gamma) |\ l_2(V,\gamma) = 0\}$. Now denote this point of intersection as $(V_a,\gamma_{\max})$, and the solution to (8), (9) between $(V_{\min},\gamma_a)$ and $(V_a,\gamma_{\max})$ as ∂J^a, as shown in Fig. 14. Repeating this calculation for the remaining three boundaries, only $\{(V,\gamma) |\ l_3(V,\gamma) = 0\}$ contains a point at which the associated optimal Hamiltonian, $H_3^*((V,\gamma),p)$, becomes zero. We denote this point as $(V_{\max},\gamma_b)$, where

$$\gamma_b = \sin^{-1}\left(\frac{T_{\min}}{Mg} - \frac{a_D V_{\max}^2}{Mg}\right) \tag{42}$$

and similarly calculate ∂J^b and V_b, as shown in Fig. 14. In summary, for the aircraft dynamics (8), (9) with flight envelope $F_{V\gamma}$ given by (38), and input constraints, the maximal controlled invariant subset of $F_{V\gamma}$, denoted $W_{V\gamma}^*$, is the set enclosed by

$$\begin{aligned} \partial W_{V\gamma}^* = \{(V,\gamma) |\ & (V = V_{\min}) \wedge (\gamma_{\min} \le \gamma \le \gamma_a) \\ & \vee (V,\gamma) \in \partial J^a \vee \\ & (\gamma = \gamma_{\max}) \wedge (V_a \le V \le V_{\max}) \\ & \vee (V = V_{\max}) \wedge (\gamma_b \le \gamma \le \gamma_{\max}) \\ & \vee (V,\gamma) \in \partial J^b \vee (\gamma = \gamma_{\min}) \\ & \wedge (V_{\min} \le V \le V_b)\}. \end{aligned} \tag{43}$$

The least restrictive controller that renders $W_{V\gamma}^*$ controlled invariant is $g(V,\gamma) = \mathbf{U} \cap \hat{g}(V,\gamma)$, where

$$\hat{g}(V,\gamma) = \begin{cases} \emptyset, & \text{if } (V,\gamma) \in (W_{V\gamma}^*)^c \\ T \ge T_a(\gamma), & \text{if } (V = V_{\min}) \\ & \quad \wedge (\gamma_{\min} \le \gamma \le \gamma_a) \\ \theta = \theta_{\min} \wedge T = T_{\max}, & \text{if } (V,\gamma) \in \partial J^a \\ \theta \le \theta_c(V), & \text{if } (\gamma = \gamma_{\max}) \\ & \quad \wedge (V_a \le V \le V_{\max}) \\ T \le T_b(\gamma), & \text{if } (V = V_{\max}) \\ & \quad \wedge (\gamma_b \le \gamma \le \gamma_{\max}) \\ \theta = \theta_{\max} \wedge T = T_{\min}, & \text{if } (V,\gamma) \in \partial J^b \\ \theta \ge \theta_d(V), & \text{if } (\gamma = \gamma_{\min}) \\ & \quad \wedge (V_{\min} \le V \le V_b) \end{cases} \tag{44}$$

with

$$\begin{aligned} T_a(\gamma) &= a_D V_{\min}^2 + Mg\sin\gamma \\ T_b(\gamma) &= a_D V_{\max}^2 + Mg\sin\gamma \\ \theta_c(V) &= \frac{M}{a_L V c}\left(\frac{g\cos\gamma_{\max}}{V} - \frac{a_L V(1-c\gamma_{\max})}{M}\right) \\ \theta_d(V) &= \frac{M}{a_L V c}\left(\frac{g\cos\gamma_{\min}}{V} - \frac{a_L V(1-c\gamma_{\min})}{M}\right). \end{aligned} \tag{45}$$

In Fig. 14, the portions of $W_{V\gamma}^*$ for which all control inputs are safe ($g(V,\gamma) = \mathbf{U}$) are indicated with solid lines; those for which only a subset are safe ($g(V,\gamma) \subset \mathbf{U}$) are indicated with dashed lines. The map defines the *least restrictive safe control scheme* and determines the mode switching logic. On ∂J^a and ∂J^b, the system must be in Mode 2 or Mode 3. Anywhere else in $W_{V\gamma}^*$, any of the three modes is valid as long as the input constraints of (44) are satisfied. In the regions $F_{V\gamma}\backslash W_{V\gamma}^*$ (the upper left and lower right corners of $F_{V\gamma}$), no control inputs will keep the system inside of $F_{V\gamma}$. Repeating these calculations for the speed and altitude modes (modes 4, 5), using the dynamics (7) and envelope illustrated in Fig. 4(b), the controlled invariant subset W_{hVh}^* is computed and shown in Fig. 15, and the least restrictive control scheme is as indicated. This calculation incorporates the limits on the altitude h into the previous calculation: at $h = h_{\max}$, the control must be chosen so that $\ddot{h} \le 0$, whereas at $h = h_{\min}$, the control is restricted to force $\ddot{h} \ge 0$.

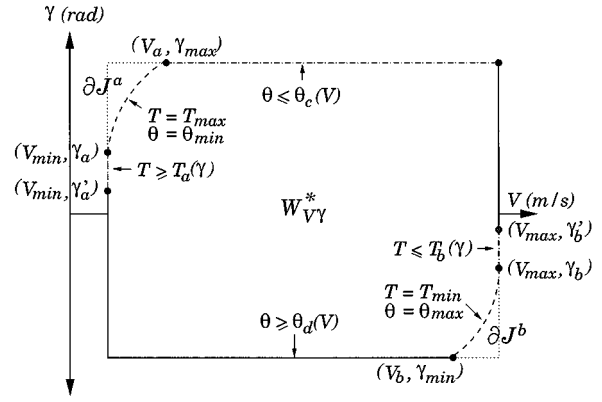

Fig. 14. The set $W^*_{V\gamma}$ in (V, γ)-space, with control law as indicated. Values used are for a DC-8: $\gamma_{\min} = -\pi/8$ rad, $\gamma_{\max} = \pi/8$ rad, $V_{\min} = 180$ m/s, $V_{\max} = 240$ m/s, $\theta_{\min} = -\pi/8$ rad, $\theta_{\max} = \pi/8$ rad, $T_{\min} = 40$ kN, $T_{\max} = 80$ kN.

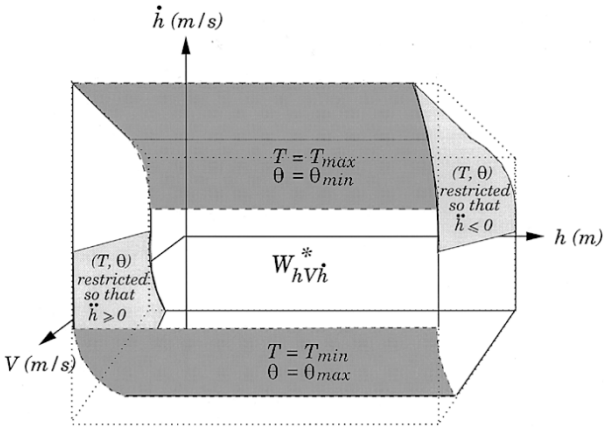

Fig. 15. The set $W^*_{hV\dot{h}}$ in $(h, V, \dot{h})$-space, with control law as indicated. Altitudes are $h_{\min} = 10$ kft, $h_{\max} = 51$ kft.

We would now like to apply Algorithm 2 to generate the controllable actions σ_1^{ij}, which force transitions between discrete states to ensure safety. However, we quickly run into a problem. At the first step of the algorithm, $W^0 = F$, and since there are no uncontrollable actions, $Pre_2(F^c) = F^c$. However, since the controllable actions are always enabled, $Pre_1(F) = F$. Thus $Reach(Pre_2(F^c), Pre_1(F)) = F^c$ so that $W^{-1} = F \backslash F^c = F$. Similarly, $W^{-2} = F$, $W^{-3} = F$, and the fixed point is $W^* = W^0$, meaning that the maximal controlled invariant set contained in F is F itself! This is clearly incorrect for the real system: the calculations to produce Figs. 14 and 15 showed that certain "corners" of F are not controlled invariant. The error lies in the fact that this system is Zeno: if forced into one of these corners, the system could avoid flowing out of F by switching infinitely often in zero time between discrete states. Unlike the previous examples, there is no specified minimum time for the system to stay in each discrete state. A possible remedy is to enforce that the system remain in each discrete state for some minimum time $T > 0$. If this is the case, then the algorithm calculates W^* as the union of $W^*_{hV\dot{h}}$ and $W^*_{V\gamma}$ for their applicable discrete modes. The mode switching logic is implicit in these calculations: as the aircraft approaches maximum or minimum altitude, the FMS must force the autopilot to switch to modes 4 or 5 and choose a control scheme which satisfies the limits on $\ddot{h}$. As the aircraft approaches its maximum or minimum speed and flight path angle, the FMS must force the system into modes 1, 2, or 3 and select those control inputs which either drive the aircraft back inside the envelope, or keep it on the boundary of the envelope.

In summary, this example uses the Hamilton–Jacobi formulation of Section IV to calculate the maximal controlled invariant set within the specified aerodynamic flight envelope, as well as the least restrictive control scheme that renders this set invariant.

C. Vehicle Collision Avoidance

The design of safe controllers for AHS platoon leaders can be cast as a game between the control (representing the acceleration of platoon A) and the disturbance (representing the acceleration of platoon B and the effect of intraplatoon collisions within platoons A and B) over a cost function

$$J(q^0, x^0, u, d) = \inf_{t \geq 0} x_2(t)$$

that encodes the requirement that the two platoons should not collide. Fortunately, the system is simple enough that physical intuition allows us to guess the optimal strategy for both the control and the disturbance. The worst that can happen from the point of view of platoon A is that both collisions take place immediately and at the maximum possible relative velocity, and then platoon B decelerates as hard as possible until it comes to a stop. The best that platoon A can do in response is also to decelerate as hard as possible until it comes to a stop.[3] In other words, $d^* = (\ddot{x}_B^*, T_A^*, T_B^*, \delta v_A^*, \delta v_B^*)$ with $T_A^* = T_B^* = 0$, $\delta v_A^* = \min\{v^S, v^{\max} - x_1(0)\}$, $\delta v_B = \min\{v^S, x_1(0) + x_3(0))\}$, and

$$u^*(q, x) = \begin{cases} a_A^{\min}, & \text{if } x_1 > 0 \\ 0, & \text{if } x_1 = 0 \end{cases}$$

and

$$\ddot{x}_B^*(q, x) = \begin{cases} a_B^{\min}, & \text{if } x_1 + x_3 > 0 \\ 0, & \text{if } x_1 + x_3 = 0. \end{cases}$$

Notice that the inputs are in feedback form and can naturally be encoded by a trivial hybrid controller. By direct computation, one can show that u^* and d^* not only satisfy the conditions of Section IV, but are in addition a saddle equilibrium for the two-player game, that is, for all (q^0, x^0), u, and d

$$J(q^0, x^0, u, d^*) \leq J(q^0, x^0, u^*, d^*) \leq J(q^0, x^0, u^*, d).$$

In other words, a player can never improve his/her situation by changing unilaterally away from the saddle equilibrium. Let $J^*(q^0, x^0) = J(q^0, x^0, u^*, d^*)$.

[3]It should be noted that physical intuition may lead to erroneous conclusions even for small changes in the specification. For example, one can show [13] that if the safety specification is relaxed from requiring no intercollisions between A and B to allowing collisions at low relative velocity, maximum deceleration may no longer be the optimum strategy for the control or disturbance.

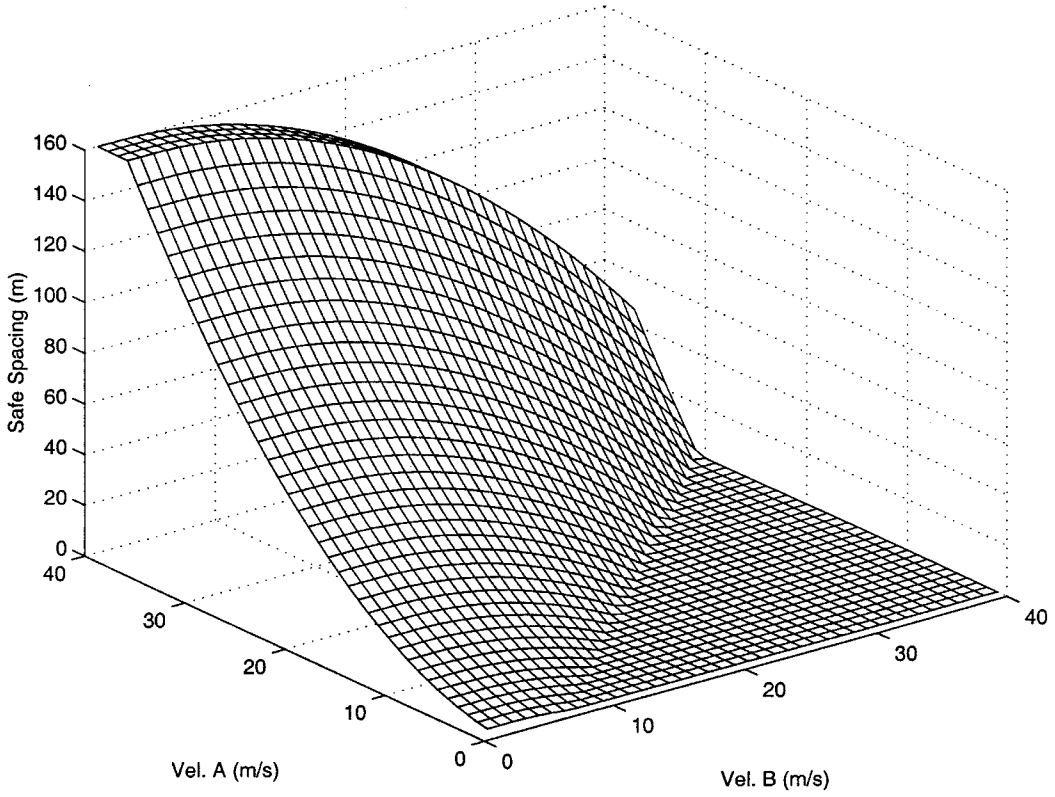

Fig. 16. The boundary of the maximal controlled invariant set for vehicle collision avoidance in $q = q_0$.

The computation used to show that (u^*, d^*) is a saddle equilibrium also allows us to analytically characterize the maximal controlled invariant set

$$W^* = \{(q, x) \in X | J^*(q, c) \geq 0\}. \tag{46}$$

The boundary of the set $a_A^{\min} = -5$ m/s^2, $a_B^{\min} = -6$ m/s^2, $v^S = 3$ m/s, and $v^{\max} = 40$ m/s is shown pictorially in Fig. 16 for $q = q_0$ (the safe set is everything "above" the boundary shown in the figure). The least restrictive controller that renders W^* invariant is

$$q^*(q, x) = \begin{cases} u^*, & \text{if } J^*(q, c) = 0 \\ [0, a_A^{\max}], & \text{if } (J^*(q, x) > 0) \wedge (x_1 = 0) \\ [a_A^{\min}, a_A^{\max}], & \text{if } (J^*(q, x) > 0) \wedge (0 < x_1 < v^{\max}) \\ [a_A^{\min}, 0], & \text{if } (J^*(q, x) > 0) \wedge (x_1 \geq v^{\max}). \end{cases} \tag{47}$$

If additional requirements, such as passenger comfort, fuel efficiency, and emission reduction, are imposed, controllers that optimize the system performance with respect to these objectives can be sought among the class of controllers defined by g^*.

Similar computations lead to maximal controlled invariant sets and least restrictive controllers to implement the remaining functions that an automated vehicle may be called upon to perform: join a platoon, split from a platoon, change lanes, etc. In addition to characterizing the safe inputs for each maneuver, the controlled invariant sets also provide guidelines for the switching among the different controllers that is carried out by the communication protocols that coordinate the actions of neighboring platoons. A controller should not be switched on unless the state is in the corresponding controlled invariant set. For technical details on how this can be accomplished, the reader is referred to [13].

VI. Computational Methods

The algorithm for reachability analysis and controller synthesis for hybrid systems presented here provides the complete necessary and sufficient conditions for design of the controller provided that the operators Pre_1, Pre_2, $Reach$ can be computed. As we pointed out, the set of systems for which these can be computed is very restrictive: timed or linear hybrid automata. Here we discuss computational techniques for approximating the optimal control and disturbance inputs $(u^*(\cdot), d^*(\cdot))$, as well as solutions to the Hamilton–Jacobi partial differential equation.

Numerical solutions are potentially complicated by the facts that the right-hand side of the Hamilton–Jacobi equation is nonsmooth and that the initial data may have a nonsmooth boundary, that $(u^*(\cdot), d^*(\cdot))$ may be discontinuous, and that $J^*(x, t)$ may not remain a continuous function of x and t even if the boundary condition $J^*(x, 0) = l(x)$ is differentiable (this is known as a *shock*). The discontinuity on the right-hand side of (24) further complicates the solution, as does the discontinuous switching of the optimal control and disturbance u^* and d^*. In addition, we are often interested in cases in which G

has nonsmooth boundary, so that the boundary conditions of the Hamilton–Jacobi equation are not differentiable. In order to admit discontinuous solutions, a "weak" derivative and "weak" solution to the Hamilton–Jacobi equation was developed by Crandall *et al.* in the early 1980s [49], [50]. A *viscosity solution* to (24) is defined as the limit as ϵ goes to zero of solutions $J^*_\epsilon(x, t)$ to the Hamilton–Jacobi equation regularized by adding $\epsilon\Delta J^*_\epsilon$ to the right-hand side; here ΔJ^* refers to the Laplacian of J^*. For $\epsilon > 0$ and for smooth Hamiltonians, it may be shown [49], [50] that there exists a unique continuous solution to the Hamilton–Jacobi equation: the second derivative term $\Delta J_\epsilon(x, t)$ acts like a smoothing term and is called a "viscosity" term for that reason. As $\epsilon \to 0$, the solution $J_\epsilon(x, t)$ approaches the viscosity solution to the Hamilton–Jacobi equation. Thus, even when classical smooth solutions do not exist, solutions in this "weak sense" exist.

A. Level Set Methods for Computing Solutions to Hybrid Systems

We discuss a numerical technique developed by Osher and Sethian [51], which computes the viscosity solution to the Hamilton–Jacobi equation, ensuring that discontinuities are preserved. We conclude with a discussion of its application to the reachability analysis of hybrid systems. The level set methods of Osher and Sethian compute the solution of the Hamilton–Jacobi equation to be the one obtained from the regularized system as the viscosity coefficient $\epsilon \to 0$.

In order for the numerical scheme to closely approximate the gradient $\partial J^*(x, t)/x$, especially at points of discontinuity, the numerical approximation of the spatial derivative must be chosen carefully. Consider an example in two dimensions, with X discretized into a grid with spacing Δx_1 and Δx_2. The *forward difference operator* D^{+x_i} at $x = (x_1, x_2)$ is defined as (for x_1, similarly for x_2)

$$D^{+x_1}J^*(x, t) = \frac{J^*((x_1 + \Delta x_1, x_2), t) - J^*(x, t)}{\Delta x_1}. \quad (48)$$

The *backward difference operator* D^{-x_i} is defined as (for x_1, similarly for x_2)

$$D^{-x_1}J^*(x, t) = \frac{J^*(x, t) - J^*((x_1 - \Delta x_1, x_2), t)}{\Delta x_1}. \quad (49)$$

Similarly, the *central difference operator* D^{0x_i} is defined as (for x_1, similarly for x_2)

$$D^{0x_1}J^*(x, t) = \frac{D^{+x_1}J^*(x, t) + D^{-x_1}J^*(x, t)}{2}. \quad (50)$$

At each grid point $x = (x_1, x_2)$, the partial derivatives $(\partial J^*(x, t)/x_1)$ and $(\partial J^*(x, t)/x_2)$ may be approximated to first order using the forward, backward, or central difference operators. The correct choice of operator depends on the direction of $f(x, u^*, d^*)$ [in our case, it depends on $-f(x, u^*, d^*)$ since we compute backward in time]. If $-f(x, u^*, d^*)$ flows from left to right (from smaller to larger values of x_1), then D^{-x_1} should be used to approximate $(\partial J^*(x, t)/x_1)$ (and vice versa); and if $-f(x, u^*, d^*)$ flows from bottom to top (from smaller to larger values of x_2), then D^{-x_2} should be used to approximate $(\partial J^*(x, t)/x_2)$ (and vice versa). Such an approximation is called an *upwind* scheme, since it uses information upwind of the direction that information propagates.

The algorithm for the two-dimensional example proceeds as follows. Choose a domain of interest in X and discretize the domain with a grid of spacing Δx_1, Δx_2. Let x_{ij} represent the grid point $(i\Delta x_1, j\Delta x_2)$ and let $\tilde{J}^*(x_{ij}, t)$ represent the numerical approximation of $J^*(x_{ij}, t)$. Using the boundary condition $J^*(x, 0) = l(x)$, compute $\tilde{J}^*(x_{ij}, 0)$ for each x_{ij}.

Let $t = 0$. While $\tilde{J}^*(x_{ij}, t) \neq \tilde{J}^*(x_{ij}, t - \Delta t)$, perform the following steps.

1) Compute

$$u^*\left(x_{ij}, \frac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x_1}, \frac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x_2}\right)$$
$$d^*\left(x_{ij}, \frac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x_1}, \frac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x_2}\right)$$

using the initial approximations to the derivatives

$$\frac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x_1} = D^{0x_1}, \quad \frac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x_2} = D^{0x_2}. \quad (51)$$

2) Calculate $f(x_{ij}, u^*, d^*)$.
3) If $-f(x_{ij}, u^*, d^*)$ flows from larger to smaller values of x_1, let

$$\frac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x_1} = D^{+x_1} \quad (52)$$

else use D^{-x_1}.
4) If $-f(x_{ij}, u^*, d^*)$ flows from larger to smaller values of x_2, let

$$\frac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x_2} = D^{+x_2} \quad (53)$$

otherwise use D^{-x_2}.
5) Compute $\tilde{J}^*(x_{ij}, t - \Delta t)$. For x_{ij} such that $J^*(\tilde{x}_{ij}, t) > 0$

$$\tilde{J}^*(x_{ij}, t - \Delta t) = \tilde{J}^*(x_{ij}, t) + \Delta t \frac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x} f(x_{ij}, u^*, d^*).$$

For x_{ij} such that $J^*(\tilde{x}_{ij}, t) \leq 0$

$$\tilde{J}^*(x_{ij}, t - \Delta t) = \begin{cases} \tilde{J}^*(x_{ij}, t) + \Delta t \dfrac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x} f(x_{ij}, u^*, d^*), \\ \quad \text{if } \dfrac{\partial \tilde{J}^*(x_{ij}, t)}{\partial x} f(x_{ij}, u^*, d^*) < 0 \\ \tilde{J}^*(x_{ij}, t), \qquad \text{otherwise.} \end{cases}$$

We have recently designed a tool for computing reachable sets for hybrid systems based on this level set technique [52], have implemented it in Matlab 5.3, and have used it to compute reachable sets for several examples, including the first example in this paper. Using a grid spacing of $\Delta x = 0.1$ (or about 90 000 grid points) each iteration of this

example required about 1400 timesteps on a Sun UltraSparc 10 (a 300-MHz UltraSparc processor with 512-KB cache and 128-MB main memory). This translated to about 75 min. Computation time will decrease significantly with our new version in C, which exploits opportunities for parallelism in the algorithm. In addition, our current version used the very basic idea in level set methods presented above; for special forms of the Hamilton–Jacobi equation, many extremely efficient variants of this method exist [53]. In particular, the *narrow-band* and *fast marching* methods speed up the algorithm by confining the computation to a narrow band around the evolving front.

B. Other Computational Methods Involving Approximations

Other methods have been presented for approximating the reach set calculation. One idea has been to use rectangular hybrid automata to approximate conservatively the reach set of general hybrid automata. This procedure consists of subdividing the state space into regions where one can find upper and lower bounds for each component of the right-hand side of the continuous dynamics and using the reach set analysis for the resulting rectangular hybrid system. The package HyTech does precisely this computation provided that the guards and invariants are polyhedra [54]. A synthesis procedure based on this appears in the paper by Wong-Toi [23]. The main advantage of this approximation procedure is that it deals with a class of systems for which the synthesis algorithm is semidecidable. The main drawback is that there is an exponential growth in the number of discrete states in approximating the continuous dynamics. The successor to HyTech is a package called HyperTech [55], which reduces the conservativeness of HyTech by using interval arithmetic with some systematic checks to reduce the divergence of interval arithmetic estimates to approximate reach sets. A controller design procedure using HyperTech has yet to be completed.

1) Approximating Dynamics with Differential Inclusions: Suppose the continuous dynamics in the nonlinear hybrid automaton (3) were approximated with the differential inclusion

$$\dot{x} \in g(q, x) \tag{54}$$

where $g(q, x) = \{f(q, x, u, d) | \forall u \in \mathbf{U}, d \in \mathbf{D}\}$. A computationally efficient method for approximating the reach set of $g(q, x)$ is to conservatively approximate $g(q, x)$ by a set of constant inclusions, each of the form

$$\dot{x} \in [g_{\min}, g_{\max}] \tag{55}$$

and then to compute the reach set of the constant inclusions. This method is presented in [56] and [57], where it is proved that the approximation error can be made arbitrarily small by approximating the differential inclusion arbitrarily closely (ϵ-approximation). An advantage of this method is that the class of constant inclusions used to approximate the differential inclusion is known to be decidable, thus one can guarantee that the reachable set as $t \to -\infty$ can be computed in a finite number of steps. The amount of preprocessing required to initially approximate the dynamics may be quite formidable, however, especially to achieve a close approximation of the true reach set.

2) Approximating Nonsmooth Sets with Smooth Sets: We have shown that the reach set at any time $t \in (-\infty, 0]$ may have a nonsmooth boundary due to switches in (u^*, d^*), nonsmooth initial data, or the formation of shocks. The level set scheme propagates these discontinuities, yet its implementation may require a very small time step to do this accurately. In [58], we present a method for overapproximating such nonsmooth sets with sets for which the boundary is continuously differentiable by using smoothing functions to derive smooth inner and outer approximations. By applying Algorithm 2 to smooth inner and outer approximations of the sets G and E, we calculate smooth inner and outer approximations to the true reach set.

3) Ellipsoidal Methods: A similar idea is to use ellipsoids as inner and outer approximations to the reach set [59], [60]. To preserve the propagation of ellipsoids, the continuous dynamics in each of the discrete locations needs to be approximated by linear dynamics. Bounds on the conservativeness of this approximation and their validity have not yet been worked out. However, [60] presents efficient algorithms for calculating both the minimum volume ellipsoid containing given points, and the maximum volume ellipsoid in a polyhedron, using a matrix determinant maximization procedure subject to linear matrix inequality constraints.

4) Quantifier Elimination and Linear Hybrid Systems: While the decidability results for the controller synthesis algorithm gave sharp results about the class of hybrid systems for which the design procedure is (semi)decidable, there has been a reawakening of interest in mathematical logic, which enables us to extend these results using so-called order-minimal or O-minimal systems. These are examples of systems that may not admit quantifier elimination but do nonetheless allow for semidecidable algorithms [61]. Using these results, we are able to perform controller synthesis for classes of hybrid systems for which the dynamics in each discrete location is linear (in the sense that $\dot{x} = A(q)x + B(q)u + E(q)d$) and the guards, invariants, and resets are subanalytic sets. This has been used in a symbolic package using QEPCAD in [46]. Finally, for hybrid systems in which the continuous state dynamics are linear and in discrete time, techniques from quantifier elimination and linear programming can be used to develop semidecidable procedures for controller design [62].

VII. Conclusion

Hybrid control design techniques are an important design tool for rapid prototyping of controller designs for real-time and embedded systems, by which one may achieve better performance, handle larger systems, and have greater confidence in the functioning of the system according to specification.

This paper is a survey of a new method of controller design for hybrid systems, along with its application to three interesting and topical examples from air-traffic management, au-

tomated highway systems, and flight management systems. We have had success in applying these methods to examples in other arenas: such as the control of unmanned aerial vehicles and communication networks. Our method represents a rapprochement between the game theoretic synthesis techniques of computer science and the robust control techniques of control theory. Current work focuses on computational methods for mechanizing the algorithm or its approximation. This is especially challenging given the limits on decidability results that we have quoted in the paper. Especially promising are level set methods, quantifier elimination methods, and ellipsoidal methods.

REFERENCES

[1] R. Sengupta and S. Lafortune, "An optimal control theory for discrete event systems," *SIAM J. Contr. Optim.*, vol. 36, no. 2, pp. 488–541, 1998.

[2] E. Asarin and O. Maler, "As soon as possible: Time optimal control for timed automata," in *Hybrid Systems: Computation and Control*, F. Vaandrager and J. van Schuppen, Eds. Berlin, Germany: Springer-Verlag, 1999, vol. 1569, Lecture Notes in Computer Science, pp. 19–30.

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

[4] G. Grammel, "Maximum principle for a hybrid system via singular pertrubations," *SIAM J. Contr. Optim.*, vol. 37, no. 4, pp. 1162–1175, 1999.

[5] P. E. Caines and Y. J. Wei, "Hierarchical hybrid control systems: A lattice theoretic formulation," *IEEE Trans. Automat. Contr.*, vol. 43, pp. 501–508, Apr. 1998.

[6] G. J. Pappas, G. Lafferriere, and S. Sastry, "Hierarchically consistent control systems," in *Proc. IEEE Conf. Decision and Control*, Tampa, FL, Dec. 1998, pp. 4336–4341.

[7] A. Nerode and W. Kohn, "Multiple agent hybrid control architecture," in *Hybrid Systems*, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. New York: Springer-Verlag, 1993, pp. 297–316.

[8] A. Church, "Logic, arithmetic, and automata," in *Proc. Int. Congr. Mathematicians*, 1962, pp. 23–35.

[9] J. R. Büchi and L. H. Landweber, "Solving sequential conditions by finite-state operators," in *Proc. Amer. Mathematical Soc.*, 1969, pp. 295–311.

[10] W. Thomas, "On the synthesis of strategies in infinite games," in *Proceedings of STACS 95*, E. W. Mayr and C. Puech, Eds. Munich, Germany: Springer-Verlag, 1995, vol. 900, Lecture Notes in Computer Science, pp. 1–13.

[11] T. Başar and G. J. Olsder, *Dynamic Non-Cooperative Game Theory*, 2nd ed. New York: Academic, 1995.

[12] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A case study in multi-agent hybrid systems," *IEEE Trans. Automat. Contr.*, vol. 43, pp. 509–521, Apr. 1998.

[13] J. Lygeros, D. N. Godbole, and S. Sastry, "Verified hybrid controllers for automated vehicles," *IEEE Trans. Automat. Contr.*, vol. 43, pp. 522–539, Apr. 1998.

[14] T. S. Perry, "In search of the future of air traffic control," *IEEE Spectrum*, vol. 34, pp. 18–35, 1997.

[15] J. W. Jackson and S. M. Green, "Control applications and challenges in air traffic management," in *Proc. Amer. Control Conf.*, Philadelphia, PA, 1998.

[16] C. J. Tomlin, "Hybrid control of air traffic management systems," Ph.D. dissertation, Dept. Elect. Eng., Univ. California, Berkeley, 1998.

[17] P. Varaiya, "Smart cars on smart roads: Problems of control," *IEEE Trans. Automat. Contr.*, vol. 38, no. 2, pp. 195–207, 1993.

[18] D. Swaroop, "String stability of interconnected systems: An application to platooning in automated highway systems," Ph.D. dissertation, Dept. Mech. Eng., Univ. California, Berkeley, 1994.

[19] D. N. Godbole and J. Lygeros, "Longitudinal control of the lead car of a platoon," *IEEE Trans. Veh. Technol.*, vol. 43, no. 4, pp. 1125–1135, 1994.

[20] O. Maler, A. Pnueli, and J. Sifakis, "On the synthesis of discrete controllers for timed systems," in *STACS 95: Theoretical Aspects of Computer Science*, E. W. Mayr and C. Puech, Eds. Munich, Germany: Springer-Verlag, 1995, vol. 900, Lecture Notes in Computer Science, pp. 229–242.

[21] E. Asarin, O. Maler, and A. Pnueli, "Symbolic controller synthesis for discrete and timed systems," in *Proceedings of Hybrid Systems II*, P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, Eds. Berlin, Germany: Springer-Verlag, 1995, vol. 999, Lecture Notes in Computer Science.

[22] M. Heymann, F. Lin, and G. Meyer, "Control synthesis for a class of hybrid systems subject to configuration-based safety constraints," in *Hybrid and Real Time Systems*, O. Maler, Ed. Berlin, Germany: Springer-Verlag, 1997, vol. 1201, Lecture Notes in Computer Science, pp. 376–391.

[23] H. Wong-Toi, "The synthesis of controllers for linear hybrid automata," in *Proc. IEEE Conf. Decision and Control*, San Diego, CA, 1997.

[24] J. Lewin, *Differential Games*. Berlin, Germany: Springer-Verlag, 1994.

[25] T. Başar and P. Bernhard, *H-Infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Boston, MA: Birkhauser, 1991.

[26] J. Lygeros, "Hierarchical, hybrid control of large scale systems," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, 1996.

[27] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, pp. 349–370, Mar. 1999.

[28] J. Lygeros, K. H. Johansson, S. Sastry, and M. Egerstedt, "On the existence of executions of hybrid automata," in *Proc. IEEE Conf. Decision and Control*, Phoenix, AZ, Dec. 7–10, 1999.

[29] R. Alur and T. A. Henzinger, "Reactive modules," in *Proc. 11th Annu. Symp. Logic in Computer Science*, 1996, pp. 207–218.

[30] Z. Manna and A. Pnueli, *Temporal Verification of Reactive Systems: Safety*. New York: Springer-Verlag, 1995.

[31] C. Hynes and L. Sherry, "Synthesis from design requirements of a hybrid system for transport aircraft longitudinal control," NASA Ames Research Center, Honeywell Air Transport Division, Preprint, 1996.

[32] C. Tomlin, J. Lygeros, L. Benvenuti, and S. Sastry, "Output tracking for a nonminimum phase dynamic CTOL aircraft model," in *Proc. IEEE Conf. Decision and Control*, New Orleans, LA, 1995, pp. 1867–1872.

[33] C. Tomlin and S. Sastry, "Bounded tracking for nonminimum phase nonlinear systems with fast zero dynamics," *Int. J. Control*, vol. 68, no. 4, pp. 819–847, 1997.

[34] C. Tomlin, J. Lygeros, and S. Sastry, "Aerodynamic envelope protection using hybrid control," in *Proc. Amer. Control Conf.*, Phildelphia, PA, 1998, pp. 1793–1796.

[35] M. O. Rabin, "Automata on infinite objects and Church's problem," in *Regional Conference Series in Mathematics*, 1972.

[36] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton, NJ: Princeton Univ. Press, 1947.

[37] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event dynamical systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[38] R. Alur, T. A. Henzinger, and O. Kupferman, "Alternating-time temporal logic," in *Proc. 38th Annu. Symp. Foundations of Computer Science*, 1997, pp. 100–109.

[39] R. Isaacs, *Differential Games*. New York: Wiley, 1967.

[40] L. Pontrjagin, "Linear differential games," *Soviet Math. Doklady*, pp. 769–771–and 910–912, 1967.

[41] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control*. Waltham, MA: Blaisdell, 1969.

[42] L. C. Young, *Optimal Control Theory*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1980.

[43] W. Fleming and R. Rishel, *Deterministic and Stochastic Optimal Control*. Berlin, Germany: Springer-Verlag, 1975.

[44] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata," in *Proc. 27th Annu. ACM Symp. Theory of Computing*, 1995.

[45] R. Alur and D. Dill, "A theory of timed automata," *Theoretical Comput. Sci.*, vol. 126, pp. 183–235, 1994.
[46] O. Shakernia, G. Pappas, and S. Sastry, "Decidable controllers for a class of hybrid systems," in *Hybrid Systems: Computation and Control*, B. Krogh and N. Lynch, Eds. Berlin, Germany: Springer-Verlag, 2000, vol. 1790, Lecture Notes in Computer Science, pp. 407–420.
[47] K. J. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry, "On the regularization of zeno hybrid automata," Syst. Contr. Lett., 1999, to be published.
[48] S. Simic, K. H. Johansson, J. Lygeros, and S. Sastry, "Toward a theory of hybrid dynamical systems," in *Hybrid Systems: Computation and Control*, B. Krogh and N. Lynch, Eds. Berlin, Germany: Springer-Verlag, 2000, vol. 1790, Lecture Notes in Computer Science, pp. 421–436.
[49] M. G. Crandall and P.-L. Lions, "Viscosity solutions of Hamilton–Jacobi equations," *Trans. Amer. Math. Soc.*, vol. 277, no. 1, pp. 1–42, 1983.
[50] M. G. Crandall, L. C. Evans, and P.-L. Lions, "Some properties of viscosity solutions of Hamilton–Jacobi equations," *Trans. Amer. Math. Soc.*, vol. 282, no. 2, pp. 487–502, 1984.
[51] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations," *J. Computat. Phys.*, vol. 79, pp. 12–49, 1988.
[52] I. Mitchell and C. Tomlin, "Level set methods for computation in hybrid systems," in *Hybrid Systems: Computation and Control*, B. Krogh and N. Lynch, Eds. Berlin, Germany: Springer-Verlag, 2000, vol. 1790, Lecture Notes in Computer Science, pp. 310–323.
[53] J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. New York: Cambridge Univ. Press, 1996.
[54] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "A user's guide to HyTech," in *Proc. 1st Int. Workshop Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*: Springer-Verlag, 1995, vol. 1019, Lecture Notes in Computer Science, pp. 41–71.
[55] T. A. Henzinger, B. Horowitz, and R. Majumdar, "Rectangular hybrid games," in *Proc. 10th Int. Conf. Concurrence Theory (CONCUR)*, 1999.
[56] A. Puri, "Theory of hybrid systems and discrete event systems," Ph.D. dissertation, Dept. Elect. Eng., Univ. California, Berkeley, 1995.
[57] A. Puri, P. Varaiya, and V. Borkar, "ϵ-approximation of differential inclusions," in *Proc. IEEE Conf. Decision and Control*, New Orleans, LA, 1995, pp. 2892–2897.
[58] J. Lygeros, C. Tomlin, and S. Sastry, "On controller synthesis for nonlinear hybrid systems," in *Proc. IEEE Conf. Decision and Control*, Tampa, FL, 1998, pp. 2101–2106.
[59] A. B. Kurzhanski and I. Valyi, *Ellipsoidal Calculus for Estimation and Control*. Boston, MA: Birkhauser, 1997.
[60] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM J. Matrix Anal. Applicat.*, vol. 19, no. 2, pp. 499–533, 1998.
[61] J. Lygeros, G. Pappas, and S. Sastry, "An approach to the verification of the Center-TRACON automation system," in *Hybrid Systems: Computation and Control*, S. Sastry and T. A. Henzinger, Eds. Berlin, Germany: Springer-Verlag, 1998, vol. 1386, Lecture Notes in Computer Science, pp. 289–305.
[62] R. Vidal, S. Schaffert, J. Lygeros, and S. Sastry, "Controlled invariance of discrete time systems," in *Hybrid Systems: Computation and Control*, B. Krogh and N. Lynch, Eds. Berlin, Germany: Springer-Verlag, 2000, vol. 1790, Lecture Notes in Computer Science, pp. 437–450.

Claire J. Tomlin (Member, IEEE) received the B.A.Sc. degree in electrical engineering from the University of Waterloo, Canada, in 1992, the M.Sc. degree in electrical engineering from Imperial College, University of London, U.K., in 1993, and the Ph.D. degree in electrical engineering from the University of California, Berkeley, in 1998.

She was a graduate fellow in the Division of Applied Sciences at Harvard University, Cambridge, MA, in 1994, and a Visiting Researcher at NASA Ames Research Center from 1994 to 1998, at Honeywell Technology Center in 1997, and at the University of British Columbia in 1994. In addition, she has spent industrial work-terms at Sofresid Engineering, Paris, and at Bell-Northern Research and Gandalf Data Limited, both in Ottawa. Since September 1998, she has been Assistant Professor in the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, with a courtesy appointment in Electrical Engineering. Her research interests are in hybrid and nonlinear control systems, air traffic management, and formation flying of UAVs.

Dr. Tomlin is a recipient of the Terman Fellowship, Stanford (1998), the Bernard Friedman Memorial Prize in applied mathematics, Berkeley (1998), the Zonta Amelia Earhart Awards for aeronautics research (1996–1998), the Natural Sciences and Engineering Research Council of Canada 1967 Scholarship (1992), the Athlone–Vanier and British Council Engineering Fellowships (1992), and the Bell Canada Engineering and Computer Science Award (1991).

John Lygeros (Member, IEEE) received the B.Eng. degree in electrical and electronic engineering in 1990 and the M.Sc. degree in systems and control in 1991, both from Imperial College of Science, Technology, and Medicine, London, U.K. He received the Ph.D. degree from the Electrical Engineering and Computer Sciences Department, University of California, Berkeley (UCB), in 1996.

From June to October 1996, he was a Visiting Postdoctoral Researcher in the California PATH project, Institute of Transportation Studies, UCB. Until September 1997, he was a Postdoctoral Research Associate at the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA. From October 1997 to December 1999, he was a Postdoctoral Researcher at the Electrical Engineering and Computer Sciences Department, UCB. From May 1998 to December 1999, he was also employed as a part-time Research Engineer at SRI International. He is now a Lecturer in information engineering at the Engineering Department, University of Cambridge, U.K. His research interests include hierarchical systems, hybrid systems, nonlinear control, and their application to automated highway systems, air traffic management, uninhabited aerial vehicles, and power networks.

Dr. Lygeros is the co-recipient of the 1997 Eliahu Jury award, awarded by the Electrical Engineering and Computer Sciences Department, UCB, for "Excellence in Systems Research."

S. Shankar Sastry (Fellow, IEEE) received the Ph.D. degree from the University of California, Berkeley (UCB), in 1981.

He was on the faculty of the Massachusetts Institute of Technology, Cambridge, from 1980 to 1982, and at Harvard University as a Gordon McKay Professor in 1994. He is currently Director of the Information Technology Office at DARPA, on leave from UCB, where he is Professor of electrical engineering and computer sciences and bioengineering. He was Director of the Electronics Research Laboratory at UCB from 1996 to 1999. He has held visiting appointments at the Australian National University, Canberra, the University of Rome, Scuola Normale, the University of Pisa, the CNRS Laboratory LAAS in Toulouse, and VERIMAG, Grenoble. His areas of research are hybrid, nonlinear, and adaptive control, robotic telesurgery, embedded and autonomous software, learning and adaptation in natural and biological systems. He is a co-author, with M. Bodson, of *Adaptive Control: Stability, Convergence and Robustness* (Englewood Cliffs, NJ: Prentice Hall, 1989), and, with R. Murray and Z. Li, of *A Mathematical Introduction to Robotic Manipulation* (Boca Raton, FL: CRC, 1994). His newest book is *Nonlinear Systems: Analysis, Stability and Control* (New York: Springer-Verlag, 1999). He has co-edited five proceedings volumes in the Springer Lecture Notes in Computer Science Series in hybrid control and robotics.

Dr. Sastry received the President of India Gold Medal in 1977, the IBM Faculty Development award for 1983–1985, the NSF Presidential Young Investigator Award in 1985, the Eckman Award of the American Automatic Control Council in 1990, and a distinguished alumnus award of the IIT Bombay in 1999.