



# Controllers for reachability specifications for hybrid systems<sup>1</sup>

John Lygeros\*, Claire Tomlin, Shankar Sastry

*Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720, USA*

Received 4 August 1997; revised 25 May 1998; received in final form 29 September 1998

*The problem of controller synthesis for reachability specifications is formulated for hybrid systems. A synthesis methodology based on techniques from game theory is proposed to solve it. The effectiveness of the methodology is demonstrated on two detailed examples*

---

## Abstract

The problem of systematically synthesizing hybrid controllers which satisfy multiple control objectives is considered. We present a technique, based on the principles of optimal control, for determining the class of least restrictive controllers that satisfies the most important objective (which we refer to as safety). The system performance with respect to lower priority objectives (which we refer to as efficiency) can then be optimized within this class. We motivate our approach by showing how the proposed synthesis technique simplifies to well-known results from supervisory control and pursuit evasion games when restricted to purely discrete and purely continuous systems respectively. We then illustrate the application of this technique to two examples, one hybrid (the steam boiler benchmark problem), and one primarily continuous (a flight vehicle management system with discrete flight modes). © 1999 Elsevier Science Ltd. All rights reserved.

*Keywords:* Hybrid systems; Safety properties; Controlled invariance; Multi-objective controllers

---

## 1. Introduction

Hybrid systems, or systems that involve the interaction of discrete and continuous dynamics, have attracted the attention of researchers from a number of traditionally distinct fields. Computer scientists have approached hybrid systems by extending techniques that have proved fruitful for discrete systems. The main problem addressed in this setting has been formal verification, that is proving that the hybrid system satisfies certain specifications. One approach to this problem comes from the area of *model checking* (Alur et al., 1993; Alur and Dill, 1994; Henzinger et al., 1995a; Nicollin et al., 1993), where the emphasis is on systems and properties that can be algorithmically verified. For certain classes of hybrid systems for which the model checking approach is applicable, the

verification process can be completely automated: a number of computational tools have been developed to take advantage of this property (Daws et al., 1994; Henzinger et al., 1995b; Kurshan, 1994). A different approach in the computer science literature has been to extend *deductive techniques* (Branicky et al., 1997; Heitmeyer and Lynch, 1994; Manna and Pnueli 1995). Here the emphasis has been on developing models (Lynch et al., 1996; Manna and Pnueli, 1992) that provide formal semantics for composition and abstraction, and support proof techniques such as induction on the length of the system executions, invariant assertions and simulation relations. Even though automatic theorem provers can facilitate the process, most of the responsibility for the proof with this approach falls on the designer.

Researchers in the areas of dynamical systems and control have approached hybrid systems from a “continuous state space and continuous/discrete time” point of view. Part of the research effort has been devoted to extending the standard modeling (Branicky et al., 1998; Brockett, 1993; Nerode and Kohn 1993a) and simulation techniques (Anderson et al., 1994; Deshpande et al., 1997;

---

\* Corresponding author. Tel.: + 510 643 5806; fax: + 510 642 1341; e-mail: lygeros@eecs.berkeley.edu.

<sup>1</sup> This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Guest Editors J. M. Schumacher, A. S. Morse, C. C. Pantelides and S. Sastry.

Tavernini, 1987) to capture the interaction between the continuous and discrete dynamics. Another thrust has been to develop new analysis and controller design techniques by extending existing methodologies such as Lyapunov's theorems (Branicky, 1998; Ye et al., 1998), supervisory control (Heymann et al., 1997; Lemmon et al., 1993; Maler et al., 1995; Wong-Toi, 1997) and optimal control (Branicky et al., 1998; Lygeros et al., 1996a; Nerode and Kohn, 1993b).

Our work is based on tools drawn from optimal control. In recent years, we have developed a methodology for designing controllers for large scale systems, making use of techniques from game theory and optimal control (Lygeros et al., 1996a,b). We have successfully applied these techniques to model and derive control laws for automated highway systems (Lygeros et al., 1998) and air traffic management systems (Tomlin, et al., 1998), as well as for benchmark examples such as the train gate controller (Lygeros et al., 1996b). The focus of our work so far has been on hybrid phenomena that arise due to the interaction between multiple agents (for example, the vehicles or the aircraft) in a large scale system. In this paper, we focus on the issues that arise because of the hybrid nature of the dynamics themselves (for example, a continuous system being controlled by switches). We are interested in control problems in which multiple requirements are imposed on the design, which is usually the case for most practical systems. For example, when dealing with discrete systems the requirements usually considered are those of safety (typically encoded by requirements on finite runs of the system) and liveness or fairness (typically encoded by requirements on infinite runs). For conventional control problems, on the other hand, the requirements considered are usually safety (encoded by stability or constraints on the system state) and efficiency (the requirement for small inputs or bounds on the speed of convergence).

In such a multi-objective setting some of the requirements are usually assumed to be more important than others, either explicitly or implicitly. The ranking of the requirements can be ignored if the goal is to verify the performance of a given hybrid system, since the objective in this case is to ensure that *all* the requirements are met. The ranking is important from the point of view of controller synthesis however, as one would like to ensure that the high priority specifications are not violated in favor of the low priority ones. This observation implicitly restricts the possible choices for the controllers that can be used to satisfy the lower priority specifications. Ideally, one would like to be able to classify the controllers that guarantee the high priority specifications and attempt to optimize the system performance with respect to the lower priority ones within this class of controls.

Here we present a methodology for designing hybrid controllers for hybrid systems in such a multi-objective setting. For simplicity, we restrict our attention to two

performance criteria. We will use *safety* to refer to the high priority criterion and *efficiency* to refer to the low priority one. For the most part we concentrate on the high priority, safety specification, which we assume to be formulated as a question of reachability. Using optimal control tools we determine the *largest controlled invariant safe set*, the largest set of states for which there exists a control such that the safety requirement can be satisfied. In the process we also classify the *least restrictive safe controls*, or all the controls which keep the system inside the safe set. It is assumed that the efficiency requirement can then be optimized within this class.

Our analysis is based on a simplified version of the hybrid system model introduced in Lygeros (1996c), which is outlined in Section 2. In Section 3 we present a conceptual algorithm for dealing with safety specifications and show how it reduces to standard results from supervisory control and pursuit evasion games when restricted to purely discrete and purely continuous systems respectively.

In the last two sections we illustrate the application of this algorithm to two examples. The first is the steam boiler benchmark problem (Abrial, 1996). Here the plant is hybrid: a continuous process (the level of water in the boiler) is controlled using discrete controls (pumps being switched on and off). The safety specification is one of reachability in the (continuous) state space: we would like the water level to stay within certain bounds. The efficiency requirement could be to minimize the number of times pumps are switched on and off or to equalize the "on" time among pumps. We show how the proposed algorithm allows us to determine the largest set of states for which the safety specification can be satisfied and classify all the controllers that can be used to satisfy this specification. The efficiency requirement can then be optimized within this class.

The second example is a continuous time and state system and is motivated by the design of an autopilot in a flight vehicle management system for a DC-8 aircraft. We consider the speed and flight path angle dynamics of the aircraft, a two-dimensional, highly nonlinear plant influenced by two continuous inputs, the thrust (controlled by the aircraft engine) and the pitch angle (controlled through the elevators). Switching arises from the saturation of the thrust input, which imposes three modes of operation for the aircraft: one in which both its velocity and flight path angle are controlled, one in which only the velocity is controlled, and one in which only the flight path angle is controlled. Safety is again encoded by a reachability requirement: the velocity and flight path angle should stay within specified limits (imposed by the engine limitations, wing stall conditions, etc.). We classify the controllers that guarantee safety and establish the mode switching logic required to implement them. Within this class, an efficiency requirement (the magnitude of the linear and angular accelerations) is then optimized.

## 2. Problem formulation

### 2.1. Notation

Consider a finite collection  $V$  of variables. Let  $\mathbf{V}$  denote the set of valuations of these variables, the set of all possible assignments of the variables in  $V$ .  $V$  can be thought of as a collection of symbols and  $\mathbf{V}$  as a collection of functions that assigns to each one of these symbols a value from a fixed set. By abuse of notation, we use lower-case letters to denote both a variable and its valuation; the interpretation should be clear from the context. Given a set of valuations  $W \subset \mathbf{V}$  and a subset of the variables  $V' \subset V$  we denote by  $W|_{V'} \in \mathbf{V}'$  the restriction of  $W$  to  $V'$ . We use  $\wedge$  to denote the logical AND,  $\vee$  to denote the logical OR and  $\neg$  to denote logical NOT.

The set of valuations of each variable in our model will be either a countable set or a subset of Euclidean space. We refer to variables whose set of valuations is countable as *discrete* and to variables whose set of valuations is a subset of Euclidean space as *continuous*. We assume that Euclidean space,  $\mathbb{R}^n$  for  $n \geq 0$ , is given the standard topology (generated by the Euclidean metric) whereas countable sets are given the discrete topology (all subsets are open). Product spaces are given the product topology (the open sets are the products of open sets). For a subset  $A$  of a topological space we assume the subset topology (the open sets are intersections of open sets with  $A$ ) and denote by  $\bar{A}$  the closure of  $A$  (smallest closed set containing  $A$ ), by  $A^\circ$  the interior of  $A$  (largest open set contained in  $A$ ), by  $\partial A = \bar{A} \setminus A^\circ$  the boundary of  $A$  and by  $A^c$  the complement of  $A$ . For any set  $K$ , we denote by  $K^\omega$  the set of infinite sequences ( $k: \mathbb{Z} \rightarrow K$ ) of elements of  $K$ , by  $\mathcal{K}$  the set of all piecewise continuous functions of the reals ( $k: \mathbb{R} \rightarrow K$ ) taking values in  $K$ , and by  $2^K$  the set of all subsets of  $K$ . For a sequence  $k \in K^\omega$  we use  $k[i]$  to denote its value at  $i \in \mathbb{Z}$ . For a function  $k \in \mathcal{K}$  we use  $k(t)$  to denote the value of the function at  $t \in \mathbb{R}$ . If  $K$  is a finite set we use  $|K|$  to denote its cardinality; if  $x$  is a real valued variable we use  $|x|$  to denote the absolute value of its valuation.

### 2.2. Hybrid automata

The basic entity of our models is the hybrid automaton. Hybrid automata are convenient abstractions of systems with phased operation and they appear extensively in the literature in various forms (Alur et al., 1993; Branicky et al., 1998; Deshpande, 1994; Heymann et al., 1997; Lynch et al., 1996; Nicollin et al., 1993; Puri, 1995). The model considered here is a simplified version of the one in Lygeros (1996) and Lygeros et al. (1998). Since we restrict our attention to controller synthesis *under full state information*, we drop the output variables and assume that the state variables also play the role of outputs (or equivalently that the output map is the identity).

Since we are working in a multi-objective setting, we would like to be able to capture successive limitations in the possible controls as successive specifications are considered. For this purpose we introduce state dependent input constraints as a primitive in our modeling formalism.

**Definition 1.** A *hybrid automaton*,  $H$ , is a collection  $(X, V, I, f, E, \phi)$ , with

- *State and input spaces:*  $X$  and  $V$  are disjoint sets of state and input variables. We assume that  $X = X_D \cup X_C$  and  $V = V_D \cup V_C$ , where  $X_C$  and  $V_C$  contain continuous and  $X_D$  and  $V_D$  discrete variables. We refer to  $x \in \mathbf{X}$  as the state of  $H$ , and to  $v \in \mathbf{V}$  as the input of  $H$ .
- *Initial states:*  $I \subset \mathbf{X}$  is a set of initial valuations of the state variables.
- *Continuous evolution:*  $f: \mathbf{X} \times \mathbf{V} \rightarrow T\mathbf{X}_C$  is a vector field.
- *Discrete transitions:*  $E \subset \mathbf{X} \times \mathbf{V} \times \mathbf{X}$  is a set of discrete transitions.
- *Admissible inputs:*  $\phi: \mathbf{X} \rightarrow 2^V$  gives the set of admissible inputs at a given state  $x \in \mathbf{X}$ .

To fix notation we let  $\mathbf{X}_C \subseteq \mathbb{R}^n$  and  $\mathbf{V}_C \subseteq \mathbb{R}^m$ . To ensure that the continuous evolution is well-posed we assume that:

**Assumption 1.**  $f$  is Lipschitz continuous in  $x|_{X_C}$  and continuous in  $v$ . For all  $x \in \mathbf{X}$ ,  $\phi(x) \neq \emptyset$ .

**Definition 2.** A *hybrid time trajectory*,  $\tau$ , is a finite or infinite sequence of intervals of the real line,  $\tau = \{I_i\}$ ,  $i \in \mathbb{N}$ , satisfying:

- $I_i$  is closed unless  $\tau$  is a finite sequence and  $I_i$  is the last interval, in which case it is left closed but can be right open.
- Let  $I_i = [\tau_i, \tau'_i]$ . Then for all  $i$ ,  $\tau_i \leq \tau'_i$ , and for  $i > 0$ ,  $\tau_i = \tau'_{i-1}$ .

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  $t \in [\tau_j, \tau'_j] \in \tau$ ”. For a topological space  $K$  and a  $\tau \in \mathcal{T}$ , we use  $k: \tau \rightarrow K$  as a shorthand notation for a map assigning a value from  $K$  to each  $t \in \tau$ . By abuse of notation we use  $\mathcal{K}$  to denote the set of all such maps that are piecewise continuous over all intervals in  $\tau$  with nonempty interior.

For any two  $t \in [\tau_i, \tau'_i] \in \tau$  and  $t' \in [\tau_j, \tau'_j] \in \tau$  consider the relation

$$t < t' \Leftrightarrow (i < j) \vee (i = j \wedge t - \tau_i < t' - \tau_j).$$

It is easy to see that this relation induces a linear order on the elements of the intervals of  $\tau$ .  $\tau$  is called *finite* if it is a finite sequence ending with a closed interval. We say  $\tau = \{I_i\}_{i=0}^N \in \mathcal{T}$  is a *prefix* of  $\tau' = \{J_i\}_{i=0}^M \in \mathcal{T}$  and write  $\tau \leq \tau'$  if either they are identical or,  $M \geq N$ ,  $I_i = J_i$  for all

$i = 0, \dots, N - 1$  and  $I_N \subseteq J_N$ . The prefix relation defines a partial order on  $\mathcal{T}$ . For  $t \in [\tau_j, \tau'_j] \in \tau \in \mathcal{T}$  we denote by  $\tau \downarrow_t$  the finite prefix of  $\tau$  ending at  $t$ , i.e.  $\tau \downarrow_t = \{I_i\}_{i=0}^{j-1} \cup [\tau_j, t]$ . Note that  $\mathcal{T}$  is prefix closed (i.e. for all  $\tau \in \mathcal{T}$  and all  $t \in \tau$ ,  $\tau \downarrow_t \in \mathcal{T}$ ).

**Definition 3.** An execution of a hybrid automaton  $H$  is a collection  $(\tau, x, v)$  with  $\tau \in \mathcal{T}$ ,  $x: \tau \rightarrow \mathbf{X}$ , and  $v: \tau \rightarrow \mathbf{V}$  which satisfies

- *Initial condition:*  $x(\tau_0) \in I$ .
- *Discrete evolution:* for all  $i$ ,  $(x(\tau'_{i-1}), v(\tau'_{i-1}), x(\tau_i)) \in E$ .
- *Continuous evolution:* for all  $i$  with  $\tau_i < \tau'_i$ ,  $x$  is continuous and  $v$  is piecewise continuous over  $[\tau_i, \tau'_i]$  and for all  $t \in [\tau_i, \tau'_i]$ ,  $(x(t), v(t), x(t)) \in E$ . Moreover, for all  $t \in [\tau_i, \tau'_i]$  where  $v$  is continuous  $(d/dt)(x(t)|_{x_c}) = f(x(t), v(t))$ .
- *Input constraints:* for all  $t \in \tau$ ,  $v(t) \in \phi(x(t))$ .

We use  $\chi$  to denote an execution of  $H$  and  $\mathcal{H}$  to denote the set of all executions of  $H$ . We use  $x^0 = x(\tau_0)$  to denote the initial state of an execution. We say  $\chi = (\tau, x, v)$  is *finite* if  $\tau$  is finite; we use  $\mathcal{H}^*$  to denote the set of all finite executions of  $H$  and  $\mathcal{X}^*$  to denote the restriction of  $\mathcal{H}^*$  to  $X$ . We say  $\chi \in \mathcal{H}$  is a *prefix* of  $\chi' \in \mathcal{H}$  and write  $\chi \leq \chi'$  if  $\tau \leq \tau'$  and  $(x, v)(t) = (x', v')(t)$  for all  $t \in \tau$ . As for hybrid time trajectories, the prefix relation defines a partial order on  $\mathcal{H}$ . As before we use  $\chi \downarrow_t$  to denote the finite prefix of  $\chi = (\tau, x, v)$  up to time  $t \in \tau$ , and  $x \downarrow_t$  the restriction of this prefix to  $X$ . Note that  $\mathcal{H}$  is also prefix closed.

We should stress that existence and/or uniqueness of executions is not guaranteed in general for hybrid automata. For example, one can easily construct automata that deadlock at a state  $x$  under input  $v$  by allowing the set  $\{x' \in \mathbf{X} | (x, v, x') \in E\}$  to be empty. One can also construct automata that accept multiple executions for the same initial condition and input, by allowing the same set to be more than a singleton. Finally, one can construct automata that prevent time from diverging by taking infinitely many discrete transitions in finite time. Here we will not address these technical issues, as the examples we consider are simple enough to handle without developing a general theoretical framework. Nonetheless, these issues are important for a complete theory of controller synthesis; one should require for example a controller to prevent deadlock and allow only finitely many transitions in finite time.

### 2.3. Specifications

A *property*,  $P$ , of a hybrid automaton  $H$  is a map

$$P: \mathcal{H} \rightarrow \{\text{True}, \text{False}\}. \quad (1)$$

We say a run  $\chi \in \mathcal{H}$  satisfies property  $P$  if  $P(\chi) = \text{True}$ ; we say a hybrid automaton satisfies a property  $P$  if  $P(\chi) = \text{True}$  for all  $\chi \in \mathcal{H}$ . Given a set  $F \subset \mathbf{X}$  we define

an *invariance property*, denoted by  $\square F$ , by

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

and an *eventuality property*, denoted by  $\diamond F$ , by

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

An invariance property is a special case of what is known as a *safety property* in the theoretical computer science literature (Manna and Pnueli, 1995). In the purely continuous setting invariance properties have mostly been studied in the context of viability theory (Aubin, 1991). An eventuality property, on the other hand, is a special case of what is known as a *guarantee property* in the computer science literature. The two classes of properties are dual in the sense that for all  $F \subset \mathbf{X}$  and all  $\chi \in \mathcal{H}$ :

$$\diamond F(\chi) \Leftrightarrow \neg \square F^c(\chi).$$

Here we restrict our attention to invariance properties. Modifying the standard terminology slightly we sometimes refer to them as *safety properties*. Because of the above duality, the algorithms we derive for solving the controller synthesis problems for  $\square F$  specifications can, in principle, be reinterpreted as algorithms for solving synthesis problems for  $\diamond F$  specifications. Some subtle technical issues still need to be resolved in this transition; for example, the controllers we synthesize may no longer be least restrictive.

### 2.4. Controllers

Assume that we are given a hybrid automaton  $H$  (which we refer to as the *plant*) and we are required to control it using its input variables, so that its trajectories satisfy certain properties. For the purpose of control the input variables of the plant are partitioned into two classes: *controls* and *disturbances*. We write  $V = U \cup D$  for the set of input variables of  $H$ , where  $U$  and  $D$  are the sets of control and disturbance inputs, respectively. The interpretation is that the controls can be influenced using a controller, in an attempt to guide the system, whereas the disturbances are determined by the “environment” and may potentially disrupt the controller’s plans.

More formally, we consider a *controller* for a plant  $H = (X, U \cup D, I, f, E, \phi)$  to be a map

$$C: \mathcal{X}^* \rightarrow 2^U.$$

For the hybrid automaton  $H$  and a controller  $C$  we define the set of *closed loop causal executions* by

$$\mathcal{H}_C = \{(\tau, x, (u, d)) \in \mathcal{H} | u(t) \in C(x \downarrow_t)\}.$$

Clearly,  $\mathcal{H}_C \subseteq \mathcal{H}$ . A controller is said to be *non-blocking* if for all  $(\{[\tau_i, \tau'_i]\}_{i=0}^N, x, v) \in \mathcal{H}$ ,  $C(x) \neq \emptyset$  and

$C(x) \subseteq \phi(x(\tau'_N))|_U$ . In subsequent discussion we restrict our attention to non-blocking controllers.<sup>2</sup>

We call a controller a *feedback controller* if for any two finite executions  $\chi_1 = (\tau_1, x_1, (u_1, d_1))$  and  $\chi_2 = (\tau_2, x_2, (u_2, d_2))$  ending in the same state,  $C(x_1) = C(x_2)$ . A feedback controller can be characterized by a feedback map:

$$g: \mathbf{X} \rightarrow 2^U.$$

Clearly, a feedback controller is non-blocking if and only if for all  $x \in \mathbf{X}$ ,  $g(x) \neq \emptyset$  and  $g(x) \subseteq \phi(x)$ . For a hybrid automaton  $H$  and a feedback controller with feedback map  $g$  we define the closed-loop hybrid automaton  $H_g = (X, U \cup D, I, f, E, \phi_g)$  with  $\phi_g(x)|_U = \phi(x)|_U \cap g(x)$  and  $\phi_g(x)|_D = \phi(x)|_D$  for all  $x \in \mathbf{X}$ . It is immediate that:

**Proposition 1.** *If  $C$  is feedback controller with feedback map  $g$  then  $H_g$  is a hybrid automaton with  $\mathcal{H}_g = \mathcal{H}_C$ .*

While we do not have a proof yet, we conjecture that even for a general controller  $C$ ,  $\mathcal{H}_C$  is in fact the set of executions of some hybrid automaton.

An instance of the *controller synthesis problem* consists of a pair,  $(H, P)$ , of a plant hybrid automaton and a property of that automaton. We sometimes refer to the property  $P$  as a *specification*. We say a controller,  $C$ , solves the controller synthesis problem if  $P(\chi) = \text{True}$  for all  $\chi \in \mathcal{H}_C$ . Note that a feedback controller solves the controller synthesis problem if  $H_g$  satisfies property  $P$ . A solution to the controller synthesis problem consists of either a controller that solves the problem or the answer “None” if no such controller exists. Our goal in this paper is to provide solutions to a special class of controller synthesis problems, where the desired property  $P$  is of the form  $\square F$  for some  $F \subseteq \mathbf{X}$ .

**Proposition 2.** *A controller synthesis problem  $(H, \square F)$  can be solved by some controller if and only if it can be solved by a feedback controller.*

**Proof.** The “if” part is immediate. The proof of the “only if” part is given in Appendix A.  $\square$

Motivated by Proposition 2, we restrict our attention to feedback controllers for  $(H, \square F)$  synthesis problems. For a controller synthesis problem  $(H, \square F)$  we would like to establish the largest set of initial conditions,  $I$ , for which there exists a controller that solves the synthesis problem. A subset  $W \subseteq \mathbf{X}$  is called *controlled invariant* if there exists a controller that solves the controller synthesis problem  $(H, \square W)$ , when  $I = W$ . A controlled in-

variant set  $W$  is *maximal* if it is not a proper subset of another controlled invariant set, or, equivalently, if for all  $W' \subseteq \mathbf{X}$  with  $W \subset W'$  the synthesis problem  $(H, \square W')$  with  $I = W'$  has the solution “None”. We say a controller *renders  $W$  invariant* if it solves the controller synthesis problem  $(H, \square W)$  when  $I = W$ .

**Proposition 3.** *A controller that solves the synthesis problem  $(H, \square F)$  exists for some  $I$  if and only if there exists a unique maximal controlled invariant  $W \subseteq F$ .*

**Proof.** The “if” is again immediate. Refer to Appendix A for the “only if” proof.  $\square$

Safety properties can sometimes be satisfied using trivial controllers. For example, a controller may be able to satisfy the property by creating a deadlock or by forcing the system to be *Zeno*, that is take infinitely many discrete transitions in finite time. For a controller synthesis problem  $(H, \square F)$  we would like to derive a feedback controller that solves the problem while imposing minimal restrictions on the controls it allows. This will prevent the pathological designs listed above whenever possible. It will also allow us greater flexibility when seeking optimal controllers for lower priority requirements, in the multi-objective setting. Feedback controllers that solve the synthesis problem  $(H, \square F)$  can be partially ordered by the relation

$$g_1 \subseteq g_2 \Leftrightarrow g_1(x) \subseteq g_2(x) \quad \text{for all } x \in \mathbf{X}.$$

We say that a controller that solves  $(H, \square F)$  is *least restrictive* if it is a maximal element in this partial order. While we do not have a proof yet, we conjecture that for every controller synthesis problem  $(H, \square W)$  with  $I = W$  either the solution is “None” or there exists a unique least restrictive controller that solves the problem. Note that the least restrictive controller solving  $(H, \square W)$ , with  $I = W$ , should allow *any* control input if the state is outside  $W$ .

### 2.5. Implementation issues

The notion of a feedback controller introduced above is conceptually clear but may be inadequate when it comes to implementation. For one thing, the set-valued map  $g$  allows non-deterministic choices of control inputs. Since in practice only one input can be applied to the system at any time, this nondeterminism has to somehow be resolved when the time comes to implement such a controller. The set-valued map can in this sense be thought of as a family of single-valued controllers; implementation involves choosing one controller from this family.

Formally, one would “implement” a controller by another hybrid automaton, which, when composed with the plant automaton yields the desired behavior (see, for

<sup>2</sup>Note that a non-blocking controller can still cause a deadlock if for all  $u \in \phi(x(\tau'_N))|_U$  and for all  $d \in \phi(x(\tau'_N))|_D$ ,  $\{x' \in \mathbf{X} | (x(\tau'_N), (u, d), x') \in E\} = \emptyset$ .

example, (Heymann et al., 1997; Ramadge and Wonham, 1989)). To do this one would need to introduce output variables to the hybrid automaton and define formal semantics for composition. This is done for example in Alur and Henzinger (1996), Heymann et al. (1997), Lygeros (1996) and Lynch et al. (1996) for systems without state-dependent input constraints. The process is more complicated for the models considered here because of the presence of the state-dependent input constraints, encoded by  $\phi$ .

In this paper we assume that the entire state is available to the controller. In general this will not be the case. If a controller is to be implemented by a hybrid automaton, the information the controller has about the plant is obtained through the valuations of the output variables of the plant, which are not necessarily in one to one correspondence with the valuations of the state variables. The controller synthesis problem under partial observation (output feedback) is much more complicated than the full observation (state feedback) problem addressed here.

### 3. Controller synthesis

We present an algorithm for synthesizing hybrid controllers for hybrid automata when multiple specifications are imposed on the closed-loop system. For simplicity, we restrict our attention to two specifications, which we refer to as *safety* and *efficiency*. We assume that the safety specification is of the form  $\Box F$  and that it is assigned higher priority than the efficiency specification. To guarantee that the specifications are met despite the action of the disturbances, we cast the design problem as a zero sum dynamic game. The two players in the game are the control  $u$  and the disturbance  $d$  and they compete over cost functions that encode the safety and efficiency specifications. We seek to determine the best possible control action and the worst possible disturbance. Note that if the specifications can be met for this pair then they can also be met for any other disturbance. A similar approach has been used in the literature for purely discrete systems (Büchi and Landweber, 1969), for timed automata (Maler et al., 1995) and for purely continuous systems (see for example Başar and Olsder, 1995). To establish links with this work, we show how the proposed algorithm reduces to discrete and continuous variants of the Hamilton–Jacobi equation, when addressing reachability questions for finite state machines and continuous control systems, respectively.

#### 3.1. Overview of the synthesis procedure

As higher priority is given to safety, the safety game is solved first. Consider a controller synthesis problem  $(H, \Box F)$ . The game can be cast in the standard min-max

setting by introducing a cost function induced by the discrete metric. Consider the map

$$m: \mathbf{X} \times \mathbf{X} \rightarrow \mathbb{R},$$

$$(x_1, x_2) \mapsto \begin{cases} 0 & \text{if } x_1 = x_2, \\ 1 & \text{if } x_1 \neq x_2. \end{cases}$$

It is easy to check that  $m$  satisfies the axioms of a metric. The metric induces a map on subsets of  $\mathbf{X}$  by defining

$$M: 2^{\mathbf{X}} \times 2^{\mathbf{X}} \rightarrow \mathbb{R},$$

$$(W_1, W_2) \mapsto \min_{(x_1, x_2) \in W_1 \times W_2} m(x_1, x_2).$$

In other words,  $M(W_1, W_2) = 0$  if  $W_1 \cap W_2 \neq \emptyset$  and  $M(W_1, W_2) = 1$  if  $W_1 \cap W_2 = \emptyset$ . By abuse of notation, we use  $M(x, W_1)$  to denote  $M(\{x\}, W_1)$ .

Consider an execution,  $\chi = (\tau, x, (u, d))$ , of the hybrid automaton  $H$  starting at an initial state  $x^0 \in I$ . Define the *cost* of this execution by

$$J_1: \mathcal{H} \rightarrow \mathbb{R},$$

$$\chi \mapsto \min_{t \in \tau} M(x(t), F^c).$$

Note that the cost function  $J_1$  implicitly defines a property of the hybrid automaton  $H$  by

$$P(\chi) = \text{True} \Leftrightarrow J_1(\chi) = 1.$$

It follows that:

**Proposition 4.**  $P(\chi) = \text{True}$  if and only if  $\Box F(\chi) = \text{True}$ .

Intuitively,  $u$  tries to maximize the cost function  $J_1$  (prevent the state from leaving  $F$ ). Because we have no control over the actions of  $d$ , we assume that it tries to minimize  $J_1$  (force the state to leave  $F$ ). As we would like to establish conditions under which  $\Box F$  is guaranteed to be satisfied we bias the game in favor of the disturbance whenever there is ambiguity over how the game will proceed. For example, multiple executions may be possible for the same initial condition, control and disturbance trajectories, due to nondeterminism. Moreover, the order in which the two players play in the game may be important, if one player is assumed to have access to the decision of the other player before choosing his/her action. In both these cases we would like to give the disturbance the “benefit of the doubt”.

Consider the max–min solution

$$J_1^*(x^0) = \max_g \min_{d \in \mathcal{D}} \left( \min_{\chi = (\tau, x, (u, d)) \in \mathcal{H}_g} J_1(\chi) \right). \quad (2)$$

Motivated by Proposition 5 we restrict our attention to feedback strategies for  $u$  in Eq. (2). Following the standard game-theoretic convention, the “player” who appears first on the right-hand side of Eq. (2) (the controller  $g$ ) is also assumed to play first. The player who appears second (the disturbance  $d$ ) is assumed to have access to

the strategy of the first player, when called upon to make his/her decision. The minimum over  $\chi$  removes all non-determinism. Therefore, provided a solution to this equation can be found,  $J_1^*$  is a well-defined function of the initial state  $x^0$ . In addition, the minimum over  $\chi$  implicitly restricts attention to control and disturbance trajectories that satisfy the state-based input constraint  $\phi$ .

Using  $J_1^*$  we define a set  $W_1^* \subseteq \mathbf{X}$  by

$$W_1^* = \{x^0 \in \mathbf{X} \mid J_1^*(x^0) = 1\}. \quad (3)$$

**Proposition 5.**  $W_1^*$  is the maximal controlled invariant subset of  $F$ .

**Proof.** We first show  $W_1^*$  is controlled invariant. Assume for the sake of contradiction that it is not. Then for all  $g$  there exists an  $x^0 \in W_1^*$ , a  $d \in \mathcal{D}$ , a  $\chi = (\tau, x, (u, d)) \in \mathcal{H}_g$  and a  $t \in \tau$  with  $x(t) \notin W_1^*$ . By Proposition 4,  $J_1(\chi) \neq 0$ , which, by Eq. (2), implies that  $J_1^*(x^0) = 0$ . This contradicts the assumption that  $x^0 \in W_1^*$ .

Next we show that  $W_1^*$  is maximal. Assume for the sake of contradiction that it is not, that is, there exists another controlled invariant  $\hat{W}$  with  $W_1^* \subset \hat{W} \subseteq F$ . Then, by definition, there exists a controller  $g$  such that  $\mathcal{H}_g$  satisfies  $\square F$  with  $I = \hat{W}$ . In other words, for all  $x^0 \in \hat{W}$ , for all  $d \in \mathcal{D}$  and for all  $\chi = (\tau, x, (u, d)) \in \mathcal{H}_g$ ,  $\square F(\chi) = \text{True}$ , or, equivalently,  $J_1(\chi) = 1$ . But, from Eq. (2) this would imply that  $J_1^*(x^0) = 1$ . This contradicts the assumption that  $W_1^* \subset \hat{W}$ .  $\square$

By Proposition 2 there exists a least restrictive feedback controller  $g_1: \mathbf{X} \rightarrow 2^U$  that renders  $W_1^*$  invariant.  $g_1$ , however, does not take into account the requirement for efficiency. Assume that the efficiency specification can also be encoded by a cost function

$$J_2: \mathcal{H} \rightarrow \mathbb{R}. \quad (4)$$

We can again pose the controller synthesis problem as a zero sum game between  $u$  and  $d$  over cost function  $J_2$ . As we no longer know whether the game over  $J_2$  can always be solved by a feedback strategy for  $u$  we allow general controllers in this case. Consider the max–min solution

$$J_2^*(x^0) = \max_C \min_{d \in \mathcal{D}} \left( \min_{\chi = (\tau, x, (u, d)) \in (\mathcal{H}_{g_1})_C} J_2(\chi) \right). \quad (5)$$

Note that this time the second minimum is over executions of  $\mathcal{H}_{g_1}$ , therefore,  $\chi$  is guaranteed to satisfy the safety specification, as long as  $x^0 \in W_1^*$ .

As for  $J_1$  a threshold may be imposed on  $J_2$ . For example, an execution may be acceptably “efficient” only if  $J_2(\chi) \geq 0$ . In this case,  $J_2^*$  allows us to construct the set  $W_2^*$  of states for which both safety and efficiency specifications can be satisfied

$$W_2^* = \{x^0 \in W_1^* \mid J_2^*(x^0) \geq 0\}. \quad (6)$$

If the controller achieving the maximum in Eq. (5) can be obtained in feedback form, the process can be repeated, if additional specifications with priorities lower than the efficiency specification are introduced.

### 3.2. Technical issues and special cases

The synthesis procedure outlined above hinges on being able to solve Eq. (2) for  $u$  in feedback form. In Maler et al. (1995) and Wong-Toi (1997) this is shown to be possible for discrete, timed and special classes of hybrid systems (known as linear hybrid automata). In the next two sections we show how this can be done using dynamic programming techniques for special classes of hybrid automata, with purely discrete and purely continuous dynamics. This approach is extended to general hybrid systems in Tomlin et al. (1998). In the examples considered in Sections 4 and 5 (as well as the ones Lygeros et al. (1998) and Tomlin et al. (1998)) a feedback solution to these equations can be obtained analytically. In general, new and sophisticated algorithms for solving optimal control problems (Schwartz, 1996; Sethian, 1996) will make the numerical solution of more general problems feasible.

Two special cases of the above procedure deserve explicit mention. The first is the case in which there is no disturbance. The synthesis procedure then calls for the solution to a pair (one for each objective) of optimal control problems, rather than games. An application of this special case will be demonstrated in Section 5 with a flight vehicle management example. The second special case is one in which there is no control. This is, for example, the case when a controller has already been designed and we are asked to verify its operation, or to determine the sets of initial conditions for which the specifications are satisfied. The verification problem also reduces to a pair of optimal control problems. For further discussion of this special case the reader is referred to Lygeros et al. (1996).

### 3.3. Infinite games on finite automata

Finite automata are a special class of hybrid automata, with a finite number of discrete states ( $|\mathbf{X}_D| < \infty$ ) and no continuous state and input variables ( $X_C = V_C = \emptyset$ ). Dropping the continuous time dependence, an execution of a finite automaton can be characterized more compactly as an infinite sequence of state and input valuations,  $(x, u, d) \in \mathbf{X}^\omega \times \mathbf{U}^\omega \times \mathbf{D}^\omega$ , satisfying

$$\begin{aligned} x[0] \in I, \quad & (x[i], (u[i], d[i]), x[i+1]) \in E, \\ & (u[i], d[i]) \in \phi(x[i]) \end{aligned} \quad (7)$$

for all  $i \in \mathbb{N}$ .  $E$  and  $\phi$  define a *successor relation*,  $\delta: \mathbf{X} \times \mathbf{U} \times \mathbf{D} \rightarrow 2^{\mathbf{X}}$  by

$$x' \in \delta(x, (u, d)) \Leftrightarrow [(u, d) \in \phi(x)] \wedge [(x, (u, d), x') \in E].$$

The general controller synthesis problems for this class of systems have been studied extensively in the literature (Ramadge and Wonham, 1989), often in a game theoretic setting (see Thomas (1995) for an overview). Here we restrict our attention to controller synthesis problems of the form  $(H, \square F)$ . Traditionally, the set of states for which a controller can solve this synthesis problem is calculated using the following algorithm (Maler et al., 1995):

**Initialization:**  $W^0 = F$ ,  $W^{-1} = \emptyset$ ,  $i = 0$ .

**While**  $W^i \neq W^{i-1}$  **do**

$$W^{i-1} = W^i \cap \{x \in \mathbf{X} \mid \exists u \in \phi(x)|_U \forall d \in \phi(x)|_D$$

$$\delta(x, (u, d)) \subseteq W^i\}$$

$$i = i - 1$$

**end**

At each step of the algorithm, the set  $W^i$  contains the states for which the control has a sequence of actions which will ensure that the system remains in  $F$  for at least  $i$  steps, for all possible actions of the disturbance. We use a negative index  $i$  to indicate that we are computing the predecessor at each step. Since  $W^{i-1} \subseteq W^i$  for all  $i$  the algorithm terminates in at most  $|W^0| = |F| \leq |\mathbf{X}|$  steps. Let  $W_1^*$  denote the fixed point of  $W^i$  at termination.

To illustrate the connection between this algorithm and the synthesis procedure of the previous section, we provide a solution to the two player, zero sum game using a dynamic programming argument. Consider a value function

$$J: \mathbf{X} \times \mathbb{Z}_- \rightarrow \{0, 1\} \quad (8)$$

whose evolution is governed by the difference equation

$$J(x, 0) = \begin{cases} 1, & x \in F, \\ 0, & x \in F^c, \end{cases}$$

$$J(x, i-1) - J(x, i)$$

$$= \min \left\{ 0, \max_u \min_d \left[ \min_{x' \in \delta(x, (u, d))} J(x', i) - J(x, i) \right] \right\}. \quad (9)$$

We refer to Eq. (9) as a *discrete Hamilton–Jacobi* equation. The outermost minimum is added to ensure that once a state has been labeled as unsafe it will not be relabeled as safe at a later stage. It is easy to show that Eq. (9) produces the same result as the above algorithm.

**Proposition 6** (Winning states for  $u$ ). *A fixed point  $J_1^*(x)$  of Eq. (9) is reached in a finite number of iterations. The set of states produced by the algorithm is  $W_1^* = \{x \in \mathbf{X} \mid J_1^*(x) = 1\}$ .*

**Proof.** We show more generally that  $W^i = \{x \in \mathbf{X} \mid J(x, i) = 1\}$ . The proof is by induction, refer to Appendix A.  $\square$

A feedback controller,  $g_1$ , for  $u$  that renders  $W_1^*$  invariant can now be constructed. For all  $x \in W_1^*$  the controller allows exactly the  $u \in \phi(x)|_U$  for which the next state is guaranteed to be in  $W_1^*$  for all  $d$ :

$$g_1(x) =$$

$$\begin{cases} \{u \in \phi(x)|_U \mid \min_d \min_{x' \in \delta(x, (u, d))} J_1^*(x') = 1\} & \text{if } x \in W_1^*, \\ \phi(x)|_U & \text{if } x \in (W_1^*)^c. \end{cases}$$

**Proposition 7** (Characterization of  $W_1^*$  and  $g_1$ ).  *$W_1^*$  is the maximal controlled invariant subset of  $F$ .  $g_1$  is the unique, least restrictive, non-blocking feedback controller that renders  $W_1^*$  invariant.*

**Proof.** The proof that  $W_1^*$  is the maximal controlled invariant subset of  $F$  is along the lines of the proof of Proposition 5. By construction,  $g_1(x) \neq \emptyset$  and  $g_1(x) \subseteq \phi(x)|_U$  for all  $x \in \mathbf{X}$ , therefore,  $g_1$  is non-blocking. Moreover, if for some  $i \in \mathbb{N}$ ,  $x[i] \in W_1^*$  and  $u[i] \in g_1(x)$ , then  $x[i+1] \in W_1^*$ . By induction, if  $x[0] \in W_1^*$ , then  $x[i] \in W_1^*$  for all  $i \in \mathbb{N}$ , and therefore  $g_1$  renders  $W_1^*$  invariant.

To show  $g_1$  is least restrictive, simply observe that, by construction of  $g_1$ , for all  $x \in W_1^*$  and  $u \notin g_1(x)$ , there exists a  $d \in \phi(x)|_D$  and an  $x' \in \delta(x, (u, d))$  such that  $J_1^*(x') = 0$ . Therefore, if any  $u \notin g_1(x)$  is allowed by a controller the state can leave  $W_1^*$  in one transition.  $\square$

### 3.4. Dynamic games on nonlinear continuous systems

For continuous systems, the safety specifications considered here correspond to a class of dynamic games known as *pursuit–evasion* games, where the controller wins if it can keep the system from entering a “bad” subset of the state space, called the *capture set*, while the disturbance wins if it can drive the state into the bad set (if it can “capture” the controller).

When restricted to the continuous domain, a hybrid automaton becomes a standard, nonlinear dynamical system. We set  $X_D = V_D = \emptyset$ , and  $E = \bigcup_{x \in \mathbf{X}} \bigcup_{(u, d) \in U \times D} (x, (u, d), x)$  (thus disallowing any discrete transitions). The dynamics can now be compactly characterized by a set of initial conditions  $I \subset \mathbf{X} = \mathbb{R}^n$  and a differential inclusion:

$$\dot{x}(t) = f(x(t), u(t), d(t)) \quad \text{with } (u(t), d(t)) \in \phi(x(t)). \quad (10)$$



To recover the standard form of pursuit-evasion games we assume that:

**Assumption 2.**  $\phi(x) = \mathbf{U} \times \mathbf{D}$  for all  $x \in \mathbf{X}$ . The capture set is an open set  $G = \{x \in \mathbb{R}^n | l(x) < 0\}$  with boundary  $\partial G = \{x \in \mathbb{R}^n | l(x) = 0\}$  where  $l: \mathbb{R}^n \rightarrow \mathbb{R}$  is a differentiable function with  $(\partial l / \partial x)(x) \neq 0$  on  $\partial G$ .

Defining  $F = G^c$ , we say that a controller wins the game if it solves the synthesis problem  $(H, \square F)$ .

The states on  $\partial G$  which can be forced by  $d$  into  $G$  infinitesimally constitute the *usable part* (UP) of  $\partial G$  (Başar and Olsder, 1995). They are given by

$$\text{UP} = \{x \in \partial G | \forall u \exists d \frac{\partial l}{\partial x}(x) f(x, u, d) < 0\}. \quad (11)$$

Let  $t < 0$  and consider the *value function*

$$J(x^0, u, d, t): \mathbb{R}^n \times \mathcal{U} \times \mathcal{D} \times \mathbb{R}_- \rightarrow \mathbb{R} \quad (12)$$

such that  $J(x^0, u, d, t) = l(x(0))$ . This value function may be interpreted as the cost of an execution which starts at  $x^0$  at time  $t \leq 0$ , evolves according to Eq. (10) with input  $(u, d)$ , and ends at the final state  $x(0)$ . Note that the value function depends only on the final state; there is no running cost (*Lagrangian*). This encodes the fact that we are only interested in whether or not the state eventually enters  $G$ . Let

$$J^*(x, t) = \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} J(x, u(\cdot), d(\cdot), t), \quad (13)$$

where, following Eq. (2), we implicitly restrict attention to feedback strategies for  $u$ . The set

$$\left\{ x \in \mathbf{X} \mid \min_{t' \in [t, 0]} J^*(x, t') \geq 0 \right\} \quad (14)$$

contains the states for which the system can be forced by  $u$  to stay in  $F$  for at least  $|t|$  time units, regardless of the disturbance  $d$ .

$J^*(x, t)$  can be computed using standard results from optimal control theory. First, define the *Hamiltonian* of the system as

$$H(x, p, u, d) = p^T f(x, u, d) \quad (15)$$

where  $p$  is a vector in  $\mathbb{R}^n$  called the *costate*. The optimal Hamiltonian is

$$H^*(x, p) = \max_{u \in \mathbf{U}} \min_{d \in \mathbf{D}} H(x, p, u, d). \quad (16)$$

If  $J^*(x, t)$  is a differentiable function of  $x$  and  $t$ , then it may be calculated for all  $x$  and  $t$  using the *Hamilton–Jacobi* partial differential equation:

$$-\frac{\partial J^*(x, t)}{\partial t} = H^* \left( x, \frac{\partial J^*(x, t)}{\partial x} \right) \quad (17)$$

with  $J^*(x, 0) = l(x)$ . The derivation of Eq. (17) may be found in most textbooks on optimal control (for example

Bryson and Ho, 1975). To prevent states from being relabeled as safe once they have been labeled as unsafe, we again introduce an additional minimum operation on the right-hand side of Eq. (17). The Hamilton–Jacobi equation then becomes

$$J^*(x, 0) = l(x), \quad (18)$$

$$-\frac{\partial J^*(x, t)}{\partial t} = \min \left\{ 0, H^* \left( x, \frac{\partial J^*(x, t)}{\partial x} \right) \right\}.$$

Eq. (18) is the continuous analog of Eq. (9). It should be noted that, unlike the algorithm for the discrete case, the calculation for Eq. (18) may not terminate. Computing the solution to the Hamilton–Jacobi partial differential equation is likely to be impossible using a finite computation; moreover, the algorithm involves taking the limit of the solution as  $t \rightarrow -\infty$ . A more thorough discussion of the technical problems and computational issues associated with solving Hamilton–Jacobi equations is given in the concluding section.

Assume that Eq. (18) has a differentiable solution  $J^*(x, t)$  which converges as  $t \rightarrow -\infty$  to a unique function. Let  $J_1^*(x)$  denote this function, and define  $W_1^* = \{x | J_1^*(x) \geq 0\}$ . A feedback controller that renders  $W_1^*$  invariant can now be constructed. The controller should be such that for  $x \in \partial W_1^*$  only the  $u$  for which the vector field is either tangential or points inside  $W_1^*$  can be applied:

$$g_1(x) = \begin{cases} \left\{ u \in \phi(x) \mid \min_{d \in \phi(x)|_{\mathbf{D}}} \left( \frac{\partial J_1^*(x)}{\partial x} \right)^T f(x, u, d) \geq 0 \right\} & \text{if } x \in \partial W_1^*, \\ \phi(x)|_{\mathbf{U}} & \text{if } x \in (W_1^*)^o \cup (W_1^*)^c. \end{cases}$$

**Proposition 8** (Characterization of  $W_1^*$  and  $g_1$ ).  $W_1^*$  is the maximal controlled invariant subset of  $F$ .  $g_1$  is the unique, least restrictive, nonblocking feedback controller that renders  $W_1^*$  invariant.

**Proof.** The claim that  $W_1^*$  is the maximal controlled invariant subset of  $F$  and that  $g_1$  is nonblocking and renders it invariant follow by a standard dynamic programming argument for pursuit-evasion games. To show that  $g_1$  is least restrictive, observe that for all  $x \in W_1^*$  and  $u \notin g_1(x)$ , there exists a  $d \in \phi(x)|_{\mathbf{D}}$  such that the state leaves  $W_1^*$  instantaneously.  $\square$

## 4. The steam boiler

### 4.1. Problem description

Our analysis of the steam boiler benchmark problem is based on the specification of Henzinger and Wong-Toi

(1996), which is simpler than the original specification of Abrial (1996) in that the effect of faults on the system is not considered. The steam boiler consists of a tank containing water and a heating element that causes the water to boil and escape as steam. The water is replenished by two pumps which at time  $t$  pump water into the boiler at rates  $q_1(t)$  and  $q_2(t)$  respectively. At every time  $t$ , pump  $i$  can either be on ( $q_i(t) = P_i$ ) or off ( $q_i(t) = 0$ ). There is a delay  $T_{p_i}$  between the time pump  $i$  is ordered to switch on and the time  $q_i$  switches to  $P_i$ . There is no delay when the pumps are switched off. The requirement is that the pumps are switched on and off so that the water level remains between two values  $M_1$  and  $M_2$ .

Here we will use three hybrid automata to describe the system, one for the boiler and one for each of the pumps. The specification of Henzinger and Wong-Toi (1996) also includes a valve that, together with the pumps, can be used to bring the water level to a desirable initial condition before the heating element is turned on and the boiling starts. As the valve is only used to set the initial condition, its operation will be ignored in our safety calculations.

The controller design we obtain for the steam boiler is close in spirit to the design of Heymann et al. (1997). Some differences in the model used to describe the system make it difficult to directly compare the two designs, however. For example, the model we consider does not assume that observation and control take place only at fixed sample times. In particular, we assume that the system state can be observed continuously, and that the controller can act instantaneously based on this information. The controller proposed here can therefore render more states safe, as actions can be delayed until the very last real time (as opposed to the last sample time).

Our model also assumes that not only the steam rate, but also its derivative are bounded in known ranges. This additional restriction on the disturbance is faithful to the original problem posed in Abrial (1996) and allows us to render even more states safe. The tradeoff is that the resulting automaton is no longer in the class of linear hybrid automata, and therefore is not directly amenable to algorithmic (and potentially automatic) synthesis procedures. The model considered in Heymann et al. (1997) can be thought of as a conservative “decidable” approximation of the model presented here. For a more thorough discussion of this issue see Henzinger and Wong-Toi (1996).

Finally, our controller does not involve an *a priori* restriction on the order in which pumps are switched on and off. This allows us to deal with the slightly more general case in which  $P_1 \neq P_2$ . The tradeoff is that we have to tolerate some non-determinism in the control scheme, in cases for example where the system is safe with either one or the other pump on, but not neither.

#### 4.2. System model

The boiler is modeled by a hybrid automaton,  $H_B = \{X_B, V_B, I_B, f_B, E_B, \phi_B\}$ , with two continuous states, the water level  $w$  and the rate at which steam escapes,  $r$ . The system evolution is influenced by two discrete inputs,  $q_1$  and  $q_2$ , and one continuous input, the derivative of the steam rate,  $d$ . The physical properties of the boiler impose bounds on the states and inputs:  $x_B = [w \ r]^T \in X_B = \mathbb{R} \times [0, W]$  and  $v_B = [q_1 \ q_2 \ d]^T \in V_B = \{0, P_1\} \times \{0, P_2\} \times [-U_2, U_1]$ , where  $W$ ,  $U_1$ ,  $U_2$ ,  $P_1$  and  $P_2$  are positive constants. Following (Henzinger and Wong-Toi, 1996) the dynamics are given by

$$f_B(x_B, v_B) = \begin{bmatrix} q_1 + q_2 - r \\ d \end{bmatrix} \quad \text{and} \quad E_B = \bigcup_{\substack{x_B \in X_B \\ v_B \in V_B}} (x_B, v_B, x_B).$$

Note that the set  $E_B$  does not allow any discrete jumps of the state. To ensure that the state constraints on  $r$  are not violated we introduce the state-dependent input constraint:

$$\phi_B(x)|_{\{d\}} = \begin{cases} [-U_2, U_1] & \text{if } r \in (0, W), \\ [-U_2, 0] & \text{if } r = W, \\ [0, U_1] & \text{if } r = 0. \end{cases}$$

We also assume that  $I_B \subseteq \mathbb{R} \times [0, W]$ . There are no state-dependent constraints on  $q_1$  and  $q_2$ .

Each pump can also be modeled by a hybrid automaton,  $H_{p_i} = \{X_{p_i}, V_{p_i}, I_{p_i}, f_{p_i}, E_{p_i}, \phi_{p_i}\}$ , with two discrete states  $q_i = 0$  and  $q_i = P_i$  that reflect if the pump is on or off and one continuous state,  $T_i$ , that reflects the time that has elapsed since the pump was commanded to switch on, hence  $x_{p_i} = (q_i, T_i) \in X_{p_i} = \{0, P_i\} \times \mathbb{R}_+$ . The evolution of the state is affected by a discrete input,  $u_{p_i} = u_i \in V_{p_i} = \{0, 1\}$  that takes the value 0 if the pump is commanded to switch off and 1 if the pump is commanded to switch on. For consistency we restrict the pump initial conditions to

$$I_{p_i} \subseteq \left( \bigcup_{T_i \leq T_{p_i}} (0, T_i) \right) \cup \left( \bigcup_{T_i \geq T_{p_i}} (P_i, T_i) \right).$$

The dynamics are given by  $\dot{T}_i = f_{p_i}(x_{p_i}, u_i) = u_i$  and

$$E_{p_i} = ((0, T_{p_i}), 1, (P_i, T_{p_i})) \\ \cup \left( \bigcup_{T_i \leq T_{p_i}} ((0, T_i), 0, (0, 0)) \right) \\ \cup \left( \bigcup_{T_i \leq T_{p_i}} ((0, T_i), 1, (0, T_i)) \right) \\ \cup \left( \bigcup_{T_i \geq T_{p_i}} ((P_i, T_i), 1, (P_i, T_i)) \right) \\ \cup \left( \bigcup_{T_i \geq T_{p_i}} ((P_i, T_i), 0, (0, 0)) \right).$$

There are no state-dependent input constraints.

The combined system can be obtained as the composition of  $H_B$ ,  $H_{p_1}$  and  $H_{p_2}$  through the shared variables  $q_1$  and  $q_2$  (for the formal discussion see Lygeros, 1996; Lygeros et al., 1998). The resulting automaton  $H = \{X, V, I, f, E, \phi\}$ , will have two discrete and four continuous states ( $X = X_B \cup X_{p_1} \cup X_{p_2}$ ). We will use  $x = ((q_1, q_2), [w \ r \ T_1 \ T_2]^T)$  to denote the overall state.  $H$  has two discrete control inputs  $U = \{u_1, u_2\}$  and one continuous disturbance input  $D = \{d\}$  ( $V = U \cup D = V_B \cup V_{p_1} \cup V_{p_2}$ ).  $I$  and  $E$  are simply the products of the corresponding sets of  $H_B$ ,  $H_{p_1}$  and  $H_{p_2}$  and  $f$  and  $\phi$  the products of the corresponding maps. Our goal is to design a feedback controller for  $u_1$  and  $u_2$  that keeps the water level in the interval  $w(t) \in [M_1, M_2]$  for all  $t \geq 0$ . This requirement can be encoded by a safety property  $\square F$  with  $F = \{x \in \mathbf{X} | w \in [M_1, M_2]\}$ .

### 4.3. Saddle solutions and set of safe states

The automaton  $H$  is deterministic, in the sense that there exists a unique execution corresponding to each choice of initial state  $x^0 \in I$  and control and disturbance trajectories (modulo identity transitions). Therefore, following the procedure of Section 3 we characterize the safety property  $\square F$  using two cost functions:

$$J_1^1(x^0, u_1, u_2, d) = \inf_{t \geq 0} w(t) \quad \text{and}$$

$$J_1^2(x^0, u_1, u_2, d) = - \sup_{t \geq 0} w(t). \quad (19)$$

For a given execution,  $\square F$  is satisfied if and only if  $J_1^1 \geq M_1$  and  $J_1^2 \geq -M_2$ .

It turns out that for this example one may not only solve Eq. (2) analytically in feedback form, but one also obtains a *saddle solution*. Informally, a saddle solution to the two player zero sum game is a solution to Eq. (2) where the minimum over  $d$  and the maximum over  $u$  ‘commute’, or, in other words, a solution where the order in which the players make their decision is unimportant (for a formal discussion see Başar and Olsder, 1995).

Consider the following candidate feedback saddle solution for the game with cost  $J_1^1$ :

$$u_i^{1*}(x) = 1 \quad \text{for all } x, \quad \text{and} \quad d^{1*}(x) = \begin{cases} U_1 & \text{if } r < W, \\ 0 & \text{if } r = W. \end{cases} \quad (20)$$

**Lemma 9.**  $(u_1^{1*}, u_2^{1*}, d^{1*})$  is globally a saddle solution for the game between  $(u_1, u_2)$  and  $d$  over  $J_1^1$ .

**Proof.** We check that the following inequalities hold for all  $x^0, u_1, u_2$  and  $d$ :

$$J_1^1(x^0, u_1, u_2, d^{1*}) \leq J_1^1(x^0, u_1^{1*}, u_2^{1*}, d^{1*})$$

$$\leq J_1^1(x^0, u_1^{1*}, u_2^{1*}, d).$$

The details are given in Appendix A.  $\square$

For cost  $J_1^2$  the saddle solution can be similarly calculated. Consider the candidate

$$u_i^{2*}(x) = 0 \quad \text{for all } x \quad \text{and}$$

$$d^{2*}(x) = \begin{cases} -U_2 & \text{if } r > 0, \\ 0 & \text{if } r = 0. \end{cases} \quad (21)$$

**Lemma 10.**  $(u_1^{2*}, u_2^{2*}, d^{2*})$  is a saddle solution for the game between  $(u_1, u_2)$  and  $d$  over  $J_1^2$ .

**Proof.** Similar to the proof of Lemma 9, refer to Appendix A.  $\square$

It should be noted that the saddle solution is not unique in the latter case, as far as the  $u_i$  are concerned. For example, any  $u_i$  such that  $\dot{w}(t) \leq 0$  for all  $t$  will produce the same maximum water level (equal to the initial water level).

The saddle solutions allow us to determine the set of states for which there exist inputs for the pumps such that the water level is guaranteed to remain between the specified limits for any steam rate. To accomplish this we first need to determine the costs under the saddle solutions. Let  $J_1^{1*}(x^0) = J_1^1(x^0, u_1^{1*}, u_2^{1*}, d^{1*})$ ,  $x^*$  be the state trajectory generated by  $(u_1^{1*}, u_2^{1*}, d^{1*})$  with initial condition  $x^0$ , and  $D_i = T_{p_i} - T_i^0$  be the time at which pump  $i$  starts pumping water under input  $u_i^{1*}$ .

**Lemma 11.** If  $W \leq P_1 + P_2$  then  $J_1^{1*}(x^0) = \min\{w^*(D_1), w^*(D_2)\}$ .

**Proof.** By direct calculation, refer to Appendix A for details.  $\square$

Lemma 11 allows us to calculate the boundary between safe and unsafe initial conditions. In particular, we would like  $J_1^{1*}(x) \geq M_1$ , therefore:

$$w \geq \max\{M_1 - w^*(D_1) + w, M_1 - w^*(D_2) + w\}.$$

The calculations in Appendix A indicate that  $w^*(D_i) - w$  is independent of  $w$  and is uniquely specified by the values of  $r, T_1$  and  $T_2$ . Therefore, the boundary between safe and unsafe states can be thought of as a function  $\hat{w}: [0, W] \times \mathbb{R}_+^2 \rightarrow \mathbb{R}$ , which maps  $(r, T_1, T_2)$  to the minimum water level required for safety. The level sets of  $\hat{w}$  for  $T_2 = 0$  (i.e. pump 2 initially off) and for  $T_2 \geq T_{p_2}$  (i.e. pump 2 initially fully on) are shown in Fig. 1. The parameter values used in the figure were  $M_1 = 0$ ,  $U_1 = 0.5$ ,  $W = 4$ ,  $P_1 = P_2 = 2.5$  and  $T_{p_1} = T_{p_2} = 5$ . The safety boundary for any other value of  $T_2$  will be a similar surface lying between the two surfaces of the figure. Safety ( $w(t) \geq M_1$ ) can be maintained as long as the water level is on or above the corresponding surface. As expected the higher the value of  $T_2$  the more states are

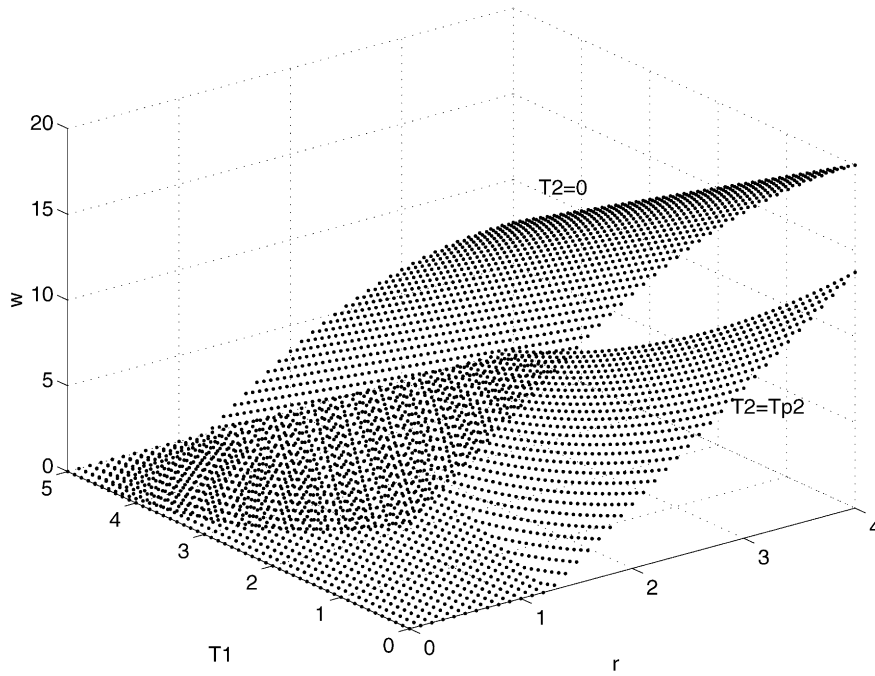


Fig. 1. Lower limit on  $w$  to avoid draining.

safe (the surface moves down). Let

$$W_1^{1*} = \{x \in \mathbf{X} | w \geq \hat{w}(r, T_1, T_2)\} \\ = \{x \in \mathbf{X} | J_1^{1*}(x) \geq M_1\}. \quad (22)$$

To determine  $J_1^{2*}$ , note that  $w^{2*}(t) \leq w$  for any  $t$ , therefore,  $J_1^{2*}(x) = -w$  and any state with  $w \leq M_2$  is safe with respect to  $J_1^2$ . However, as noted earlier, the  $u_i^{2*}$  are not the unique minimizers of  $J_1^2$ , as any control such that  $\dot{w} \leq 0$  whenever  $w$  returns to its initial value achieves the same value of  $J_1^2$ . This observation leads to a boundary between safe and unsafe states. On the boundary,  $w = M_2$  (the only situation where  $J_1^2$  becomes safety critical) and  $r = \hat{r}(T_1, T_2)$  where

$$\hat{r}: \mathbb{R}^2 \rightarrow \mathbb{R}, \\ (T_1, T_2) \mapsto \begin{cases} 0 & \text{if } T_1 < T_{p_1} \text{ and } T_2 < T_{p_2}, \\ P_1 & \text{if } T_1 \geq T_{p_1} \text{ and } T_2 < T_{p_2}, \\ P_2 & \text{if } T_1 < T_{p_1} \text{ and } T_2 \geq T_{p_2}, \\ P_1 + P_2 & \text{if } T_1 \geq T_{p_1} \text{ and } T_2 \geq T_{p_2}. \end{cases}$$

Pictorially, this boundary is shown in Fig. 2. Let

$$W_1^{2*} = \{x \in \mathbf{X} | (w < M_2) \vee [(w = M_2) \wedge (r \geq \hat{r}(T_1, T_2))]\} \\ = \{x \in \mathbf{X} | J_1^{2*}(x) \geq -M_2\}. \quad (23)$$

#### 4.4. Least restrictive safe controller

The calculation of the safe set also allows us to classify the controls that render this set invariant. To ensure the

existence of controls that are safe with respect to both draining and overflow we assume that:

**Assumption 3.**  $W \leq P_1 + P_2$  and  $\hat{w}(W, 0, 0) < M_2$ .

**Lemma 12.** The feedback controller  $g_1^1$  given by

$$u_1 \in \{0, 1\} \text{ and } u_2 \in \{0, 1\} \\ \text{if } [w > \hat{w}(r, 0, 0)] \vee [w < \hat{w}(r, T_1, T_2)], \\ u_1 = 1 \text{ and } u_2 \in \{0, 1\} \text{ if } \hat{w}(r, 0, 0) \geq w > \hat{w}(r, T_1, 0), \\ u_1 \in \{0, 1\} \text{ and } u_2 = 1 \text{ if } \hat{w}(r, 0, 0) \geq w > \hat{w}(r, 0, T_2), \\ u_1 = 1 \text{ and } u_2 = 1 \text{ if } w = \hat{w}(r, T_1, T_2).$$

is the unique, least restrictive, non-blocking, feedback controller that renders  $W_1^{1*}$  invariant.

**Proof.** The proof is a corollary of Lemma 9 and the properties of the saddle solution.  $g_1^1$  is clearly non-blocking. To show that it renders  $W_1^{1*}$  invariant, it suffices to show that any  $u$  satisfying the above conditions leads to a state trajectory with  $w(t) \geq \hat{w}(r, T_{p_1}, T_{p_2}) = M_1$  for all  $t$ . The first three cases are relevant only if  $w(t) \geq \hat{w}(r, 0, 0)$ ,  $w(t) \geq \hat{w}(r, T_1, 0)$  and  $w(t) \geq \hat{w}(r, 0, T_2)$ , respectively.<sup>3</sup> As  $\hat{w}$  is monotone in  $T_1$  and  $T_2$ , the lower bounds are greater than or equal to  $\hat{w}(r, T_{p_1}, T_{p_2})$  in all three cases. Therefore, the last case

<sup>3</sup>Or if it is already “too late”,  $w < \hat{w}(r, T_1, T_2)$ .

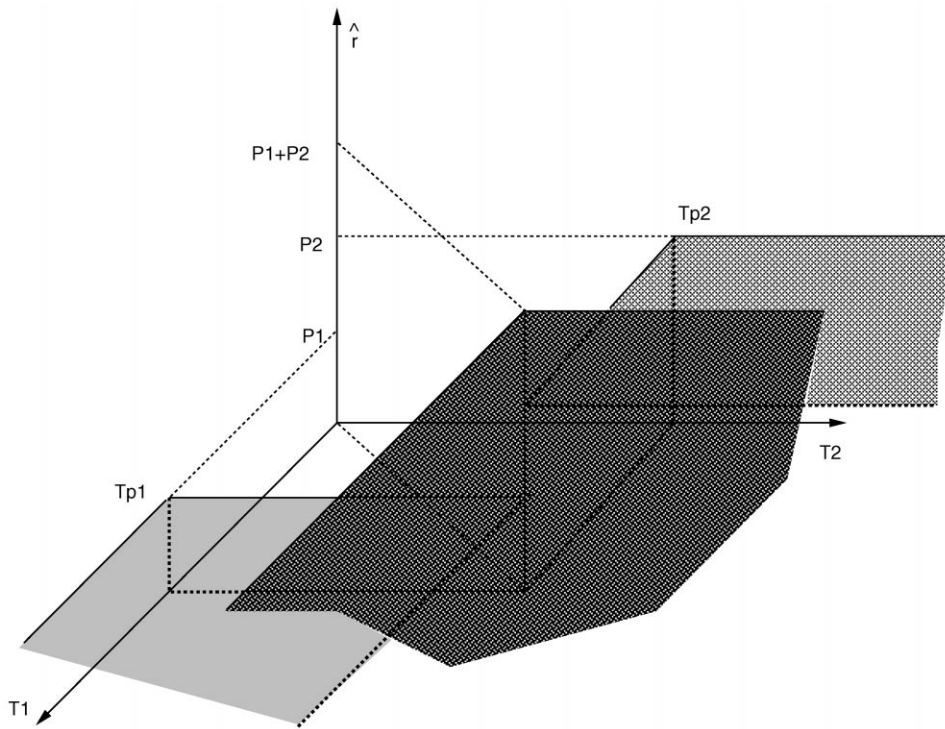


Fig. 2. Lower limit on  $r$  to avoid overflow.

( $w = \hat{w}(r, T_1, T_2)$ ) is the only one we have to worry about. Safety in this case is guaranteed by Lemma 9.

To show that  $g_1^1$  is the unique least restrictive controller that renders  $W_1^{1*}$  invariant, we only need to worry about the last three cases (the first case is trivially least restrictive). In the last three cases at least one of the  $u_i$  is restricted to be  $u_i = 1$ . If  $u_i = 0$  is used instead, the conditions on the three cases and the monotonicity of  $\hat{w}$  with respect to  $T_i$  imply that the resulting jump to  $T_i = 0$  will result in  $w < \hat{w}(r, T_1, T_2)$ . In this situation, Lemma 9 guarantees that there exists a  $d$  (for example  $d = d^{1*}$ ) and a  $t \geq 0$  such that  $w(t) < M_1$ . Therefore, any control scheme violating the proposed restrictions is unsafe.  $\square$

Note that the first term applies to states in the interior of the safe set ( $w > \hat{w}(r, 0, 0)$ ) as well as all the states outside the safe set ( $w < \hat{w}(r, T_1, T_2)$ ). The expression for  $\hat{w}$  (given in the appendix) suggests that  $\hat{w}$  is monotone in  $T_1$  and  $T_2$ . Therefore, the condition on the last case is enabled if and only if all other conditions fail. The two middle conditions may overlap, however. Therefore there is some nondeterminism in the choice of safe controls (some states may be safe with either one or the other pump on, but not neither).

**Lemma 13.** The feedback controller  $g_1^2$  given by

$$u_1 \in \{0, 1\} \text{ and } u_2 \in \{0, 1\}$$

$$\text{if } [(w < M_2) \vee (r > \hat{r}(T_1, T_2))] \vee [w > M_2],$$

$$u_1 = 0 \text{ and } u_2 \in \{0, 1\}$$

$$\text{if } [w = M_2] \wedge [\hat{r}(T_1, T_2) \geq r > \hat{r}(0, T_2)],$$

$$u_1 \in \{0, 1\} \text{ and } u_2 = 0$$

$$\text{if } [w = M_2] \wedge [\hat{r}(T_1, T_2) \geq r > \hat{r}(T_1, 0)],$$

$$u_1 = 0 \text{ and } u_2 = 0$$

$$\text{if } [w = M_2] \wedge [r \leq \min\{\hat{r}(T_1, 0), \hat{r}(0, T_2)\}]$$

is the unique, least restrictive, nonblocking, feedback controller that renders  $W_1^{2*}$  invariant.

**Proof.** The proof follows as a corollary of Lemma 10.  $g_1^2$  is clearly non-blocking. To show that it renders  $W_1^{2*}$  invariant, note that if  $w = M_2$  (the only safety critical situation), the restrictions imposed on  $u$  guarantee that  $\dot{w} \leq 0$  for  $d^{2*}$  (and hence any  $d$ ). To show that it is the unique least restrictive controller that does so note that any control violating the conditions of the lemma will result in  $\dot{w} > 0$  when  $w = M_2$ .  $\square$

Again the first term applies to states in the interior of the safe set, as well as unsafe states. Note again the nondeterminism in the choice of control in the middle two cases (the system may be safe with either one or the other pump on, but not both). Summarizing:

**Theorem 14.** The set  $W_1^* = W_1^{1*} \cap W_1^{2*}$  is the maximal controlled invariant subset of  $F$ . The controller  $g_1$  with  $g_1(x) = g_1^1(x) \cap g_1^2(x)$  for all  $x \in X$  is the unique, least

restrictive, nonblocking, feedback controller that renders  $W_1^*$  invariant.

**Proof.** Under Lemmas 12 and 13, it suffices to show that  $g(x) \neq \emptyset$  for all  $x \in \mathbf{X}$ . This, however, follows from Assumption 3 and the monotonicity of  $\hat{w}$  with respect to  $r$ ,  $T_1$  and  $T_2$ .  $\square$

## 5. Flight vehicle management systems

### 5.1. Problem description

The flight vehicle management system (FVMS) example is based on the dynamic aircraft equations and the design specification of Hynes and Sherry (1996). The equations model the speed and the flight path angle dynamics of a commercial aircraft in still air. The control inputs to the equations are the thrust  $T$ , accessed through the engine throttle, and the pitch angle  $\theta$ , accessed through the elevators. The outputs we wish to control are the speed  $V$  and the flight path angle  $\gamma$ . There are three primary modes of operation:

- (1) *Mode 1:* The thrust  $T$  is between its specified operating limits ( $T_{\min} < T < T_{\max}$ ), the control inputs are  $T$  and  $\theta$ , and both  $V$  and  $\gamma$  are controlled outputs.
- (2) *Mode 2:* The thrust saturates ( $T = T_{\min} \vee T = T_{\max}$ ) and thus it is no longer available as a control input; the only input is  $\theta$ , and the only controlled output is  $V$ .
- (3) *Mode 3:* The thrust saturates ( $T = T_{\min} \vee T = T_{\max}$ ); the input is again  $\theta$ , and the controlled output is  $\gamma$ .

Within Modes 2 and 3 there are two submodes depending on whether  $T = T_{\min}$  or  $T = T_{\max}$ .

Safety regulations for the aircraft dictate that  $V$  and  $\gamma$  must remain within specified limits. For ease of presentation we simplify this *safety envelope*,  $F$ , of Hynes and Sherry (1996) to

$$F = \{(V, \gamma) | (V_{\min} \leq V \leq V_{\max}) \wedge (\gamma_{\min} \leq \gamma \leq \gamma_{\max})\}, \quad (24)$$

where  $V_{\min}$ ,  $V_{\max}$ ,  $\gamma_{\min}$ ,  $\gamma_{\max}$  are constants. We would like to design a control scheme, an FVMS, to drive the aircraft between operating points in  $F$ . The resulting trajectory  $(V(t), \gamma(t))$  should stay within  $F$  at all times and should satisfy constraints on the linear and angular acceleration:

$$|\dot{V}| \leq 0.1g, \quad |V\dot{\gamma}| \leq 0.1g \quad (25)$$

imposed for passenger comfort.

### 5.2. System model

We model the aircraft by a hybrid automaton  $H = (X, V, I, f, E, \phi)$ . Following (Hynes and Sherry,

1996), the dynamics can be captured by two continuous state variables  $x = [V, \gamma]^T \in \mathbf{X} = \mathbb{R}^+ \times S^1$ , evolving according to a vector field  $f$  given by

$$\dot{V} = \frac{T - D}{m} - g \sin \gamma, \quad \dot{\gamma} = \frac{L}{mV} - \frac{g \cos \gamma}{V}, \quad (26)$$

where  $V$  (m/s) is the airspeed,  $\gamma$  (rad) is the flight path angle,  $T$  (N) is the thrust,  $m$  (kg) is the mass of the aircraft,  $g$  (m/s<sup>2</sup>) is gravitational acceleration and  $L$  and  $D$  are the aerodynamic lift and drag forces. The aerodynamic forces can be modeled by

$$L = a_L V^2 (1 + c(\theta - \gamma)), \\ D = a_D V^2 (1 + b(1 + c(\theta - \gamma))^2) \quad (27)$$

where  $a_L$  and  $a_D$  are the lift and drag coefficients,  $b$  and  $c$  are small positive constants, and  $\theta$  is the aircraft pitch angle.

We assume that the pilot has direct control over the thrust  $T$  and the pitch angle  $\theta$ , thus there are two continuous control inputs:<sup>4</sup>

$$u = [T, \theta]^T \in \mathbf{U} = [T_{\min}, T_{\max}] \times [\theta_{\min}, \theta_{\max}]. \quad (28)$$

There are no disturbance inputs ( $D = \emptyset$ ), no state-dependent input constraints ( $\phi(x) \equiv \mathbf{U}$ ) and no discrete transitions.

Substituting Eq. (27) into Eq. (26) and assuming that  $b$  is small enough to neglect the quadratic term in  $(\theta - \gamma)$  in the drag, leads to

$$f(x, u) = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{-a_D x_1^2}{m} - g \sin x_2 + \frac{1}{m} u_1 \\ \frac{a_L x_1 (1 - c x_2)}{m} - \frac{g \cos x_2}{x_1} + \frac{a_L c x_1}{m} u_2 \end{bmatrix}. \quad (29)$$

In our calculations we use the following parameter values, which correspond to a DC-8 at cruising speed, at an altitude of 35 000 ft:  $m = 85\,000$  kg,  $c = 6$ ,  $a_L = 30$ ,  $a_D = 2$ ,  $T_{\min} = 40\,000$  N,  $T_{\max} = 80\,000$  N,  $\theta_{\min} = -22.5^\circ$ ,  $\theta_{\max} = 22.5^\circ$ ,  $V_{\min} = 180$  m/s,  $V_{\max} = 240$  m/s,  $\gamma_{\min} = -22.5^\circ$  and  $\gamma_{\max} = 22.5^\circ$ . The bounds on the pitch angle  $\theta$  and the flight path angle  $\gamma$  are chosen to be symmetric about zero for ease of computation. In actual flight systems, the positive bound on these angles is greater than the negative bound.

<sup>4</sup>The bounds on the dynamics which arise from the relationship between the engine throttle and the forward thrust, and the elevators and the aircraft pitch are introduced in the calculation through the constraints on the inputs and the state variables.

5.3. Optimal controls and safe set of states

In this system, the controller “wins” if it can keep the state of the aircraft from leaving the envelope  $F$ . Since there are no disturbances in our model, the dynamic game of Section 3.4 which is used to calculate the safe set of states reduces to an optimal control problem. Let

$$\begin{aligned} l_1^1(x) &= x_1 - V_{\min}, l_1^2(x) = -x_2 + \gamma_{\max}, \\ l_1^3(x) &= -x_1 + V_{\max}, l_1^4(x) = x_2 - \gamma_{\min}. \end{aligned} \tag{30}$$

Thus

$$F = \{x \in \mathbf{X} \mid \forall i \in \{1, 2, 3, 4\}, l_1^i(x) \geq 0\}. \tag{31}$$

Note that  $\partial F$  is only piecewise smooth. This makes the formal solution to Eq. (18) more challenging technically. Here we show that, for this example, the calculation can in fact be performed one piece of the boundary at a time. We calculate the safe set of states with respect to each  $l_1^i$  separately, and prove that the intersection of the resulting sets is  $W_1^*$ .

Consider the system (29) over the time interval  $[t, 0]$ , where  $t < 0$ , and let us first calculate the safe set of states with respect to  $l_1^1(x)$ . The value function is given by

$$J_1^1(x, u(\cdot), t): \mathbb{R}^n \times \mathcal{U} \times \mathbb{R}_- \rightarrow \mathbb{R} \tag{32}$$

such that  $J_1^1(x, u(\cdot), t) = l_1^1(x(0))$ . This value function may be interpreted as the cost of a trajectory  $x(\cdot)$  which starts at  $x$  at time  $t \leq 0$ , evolves according to Eq. (29) with input  $u(\cdot)$ , and ends at the final state  $x(0)$ . The optimal cost is found by maximizing with respect to  $u$ :

$$J_1^{1*}(x, t) = \max_{u(\cdot) \in \mathcal{U}} J_1^1(x, u(\cdot), t) \tag{33}$$

and the safe set of states is therefore  $\{x \mid J_1^{1*}(x) \geq 0\}$ . Following the Hamilton–Jacobi method of Section 3.4, the optimal Hamiltonian is given by

$$\begin{aligned} H_1^{1*}(x, p) &= \max_{u \in U} \left[ p_1 \left( -\frac{a_D x_1^2}{m} - g \sin x_2 + \frac{1}{m} u_1 \right) \right. \\ &\quad \left. + p_2 \left( \frac{a_L x_1 (1 - c x_2)}{m} - \frac{g \cos x_2}{x_1} + \frac{a_L c x_1}{m} u_2 \right) \right]. \end{aligned} \tag{34}$$

The Hamilton–Jacobi equation describing the evolution of the  $J_1^{1*}(x, t)$  is

$$\begin{aligned} -\frac{\partial J_1^{1*}(x, t)}{\partial t} &= \min \left\{ 0, H_1^{1*} \left( x, \frac{\partial J_1^{1*}(x, t)}{\partial x} \right) \right\}, \\ J_1^{1*}(x, 0) &= l_1^1(x). \end{aligned} \tag{35}$$

We may compute the optimal control inputs at the time  $t = 0$  using Eq. (34). The optimal thrust input may be calculated directly from this equation:  $u_1^*(0) = T_{\max}$ . The optimal pitch input must be calculated indirectly.<sup>5</sup>

Define  $(V_{\min}, \gamma_a) = \{x \in F \mid l_1^1(x) = 0 \wedge H_1^{1*}(x) = 0\}$ . Then

$$\gamma_a = \sin^{-1} \left( \frac{T_{\max}}{mg} - \frac{a_D V_{\min}^2}{mg} \right). \tag{36}$$

Integrate the system dynamics (29) with  $x(0) = (V_{\min}, \gamma_a)$ ,  $u = u^*$ , backwards from  $t = 0$  to  $t = -T$ , where  $T$  is chosen to be large enough so that the solution intersects  $\{x \mid l_1^2(x) = 0\}$ . The optimal control  $u_2^*$  is required for this calculation. At the abnormal external  $(V_{\min}, \gamma_a)$ , any  $u_2 \in [\theta_{\min}, \theta_{\max}]$  may be used. However, as we integrate the system, we leave the abnormal extremal regardless of the choice of  $u_2$  instantaneously, and  $u_2^*$  is uniquely determined. For all  $u_2 \in [\theta_{\min}, \theta_{\max}]$ , for all  $\delta \in \mathbb{R}^+$ , the inward pointing normal to  $f(x(-\delta), [u_1^* u_2]^T)$  is such that  $p_2$  is negative, thus,  $u_2^* = \theta_{\min}$ . Denote the point of intersection of the solution of Eq. (29) with  $\{x \in F \mid l_1^2(x) = 0\}$  as  $(V_a, \gamma_{\max})$ , and the solution to Eq. (29) between  $(V_{\min}, \gamma_a)$  and  $(V_a, \gamma_{\max})$  as  $\partial W_1^a$ , as shown in Fig. 3.

The calculation can be repeated for the remaining boundaries. Of the remaining three, only  $\{x \in F \mid l_1^3(x) = 0\}$  contains a point at which the associated optimal Hamiltonian,  $H_1^{3*}(x)$ , becomes zero. We denote this point as  $(V_{\max}, \gamma_b)$  where

$$\gamma_b = \sin^{-1} \left( \frac{T_{\min}}{mg} - \frac{a_D V_{\max}^2}{mg} \right) \tag{37}$$

and similarly calculate  $\partial W_1^b$  and  $V_b$ , as shown in Fig. 3.

**Theorem 15.** For the aircraft dynamics (29) with flight envelope  $F$  given by (24) and input constraints (28), the maximal controlled invariant subset of  $F$  is the set  $W_1^*$ , enclosed by  $\partial W_1^*$ , given by

$$\begin{aligned} \partial W_1^* &= \{(V, \gamma) \mid (V = V_{\min}) \wedge (\gamma_{\min} \leq \gamma \leq \gamma_a) \\ &\quad \vee (V, \gamma) \in \partial W_1^a \\ &\quad \vee (\gamma = \gamma_{\max}) \wedge (V_a \leq V \leq V_{\max}) \end{aligned}$$

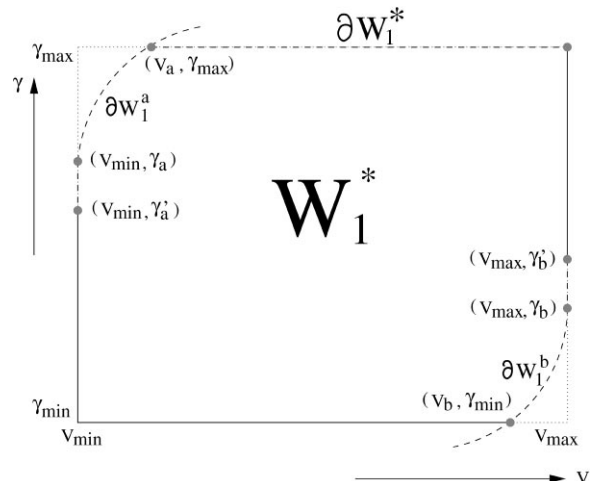


Fig. 3. The safe set of states,  $W_1^*$ , and its boundary  $\partial W_1^*$ .

<sup>5</sup>Since  $H_1^{1*}(x, p)$  loses dependence on  $u_2$  on the set  $\{x \mid l_1^1(x) = 0\}$ , the calculations involve computing the so-called abnormal extremals.

$$\begin{aligned}
& \vee (V = V_{\max}) \wedge (\gamma_b \leq \gamma \leq \gamma_{\max}) \\
& \vee (V, \gamma) \in \partial W_1^b \\
& \vee (\gamma = \gamma_{\min}) \wedge (V_{\min} \leq V \leq V_b). \quad (38)
\end{aligned}$$

**Proof.** Refer to Appendix A.  $\square$

#### 5.4. The class of least restrictive, safe controls

**Theorem 16.** *The unique, least restrictive, nonblocking, feedback controller that renders  $W_1^*$  invariant is  $g_1(x) = \mathbf{U} \cap \hat{g}_1(x)$ , where*

$$\begin{aligned}
\hat{g}_1(V, \gamma) = & \\
\{ T \geq T_a(\gamma) & \quad \text{if } (V = V_{\min}) \wedge (\gamma_{\min} \leq \gamma \leq \gamma_a) \\
\theta = \theta_{\min} \wedge T = T_{\max} & \quad \text{if } (V, \gamma) \in \partial W_1^a \\
\theta \leq \theta_c(V) & \quad \text{if } (\gamma = \gamma_{\max}) \wedge (V_a \leq V \leq V_{\max}) \\
T \leq T_b(\gamma) & \quad \text{if } (V = V_{\max}) \wedge (\gamma_b \leq \gamma \leq \gamma_{\max}) \\
\theta = \theta_{\max} \wedge T = T_{\min} & \quad \text{if } (V, \gamma) \in \partial W_1^b \\
\theta \geq \theta_d(V) & \quad \text{if } (\gamma = \gamma_{\min}) \wedge (V_{\min} \leq V \leq V_b) \\
\mathbf{U} & \quad \text{else} \} \quad (39)
\end{aligned}$$

with

$$T_a(\gamma) = a_D V_{\min}^2 + mg \sin \gamma, \quad (40)$$

$$T_b(\gamma) = a_D V_{\max}^2 + mg \sin \gamma, \quad (41)$$

$$\theta_c(V) = \frac{m}{a_L V_C} \left( \frac{g \cos \gamma_{\max}}{V} - \frac{a_L V (1 - c \gamma_{\max})}{m} \right), \quad (42)$$

$$\theta_d(V) = \frac{m}{a_L V_C} \left( \frac{g \cos \gamma_{\min}}{V} - \frac{a_L V (1 - c \gamma_{\min})}{m} \right). \quad (43)$$

**Proof.** Refer to Appendix A.  $\square$

In Fig. 3, the portions of  $\partial W_1^*$  for which all control inputs are safe ( $g_1(x) = \mathbf{U}$ ) are indicated with solid lines; those for which only a subset are safe ( $g_1(x) \subset \mathbf{U}$ ) are indicated with dashed lines. The map defines the *least restrictive safe control scheme* and determines the mode switching logic. On  $\partial W_1^a$  and  $\partial W_1^b$ , the system must be in Mode 2 or Mode 3. Anywhere else in  $W_1^*$ , any of the three modes is allowed as long as the input constraints of Eq. (39) are satisfied. In the regions  $F \setminus W_1^*$  (the upper left and lower right corners of  $F$ ), no control inputs are safe.

#### 5.5. Additional constraints for passenger comfort

Cost functions involving the linear and angular accelerations can be used to encode the requirement for passenger comfort. For  $\chi \in \mathcal{H}_{g_1}$  we define

$$J_2^1(\chi) = -\max_{t \geq 0} |\dot{x}_1(t)|, \quad J_2^2(\chi) = -\max_{t \geq 0} |x_1(t) \dot{x}_2(t)|. \quad (44)$$

The requirement that the linear and angular acceleration remain within the limits determined for comfortable travel are encoded by thresholds:

$$J_2^1(\chi) \geq -0.1g, \quad J_2^2(\chi) \geq -0.1g. \quad (45)$$

Within the class of safe controls, a control scheme which addresses the passenger comfort (efficiency) specification can be constructed. To do this, we solve the optimal control problem:

$$J_2^{1*}(x) = \max_C J_2^1(\chi), \quad J_2^{2*}(x) = \max_C J_2^2(\chi), \quad (46)$$

where  $\chi \in (\mathcal{H}_{g_1})_C$ . From this calculation, it is straightforward to determine the set of “comfortable” states:

$$W_2^* = \{x \in W_1^* \mid J_2^{1*}(x) \geq -0.1g \wedge J_2^{2*}(x) \geq -0.1g\}. \quad (47)$$

The set of comfortable controls may be calculated by substituting the bounds on the accelerations into Eq. (29) to get

$$\begin{aligned}
-0.1mg + a_D V^2 + mg \sin \gamma & \leq T \\
& \leq 0.1mg + a_D V^2 + mg \sin \gamma \\
-\frac{0.1mg}{a_L V^2 c} - \frac{1 - c\gamma}{c} + \frac{mg \cos \gamma}{a_L V^2 c} & \leq \theta \\
& \leq \frac{0.1mg}{a_L V^2 c} - \frac{1 - c\gamma}{c} + \frac{mg \cos \gamma}{a_L V^2 c}. \quad (48)
\end{aligned}$$

These constraints provide lower and upper bounds on the thrust and the pitch angle which may be applied at any point  $(V, \gamma)$  in  $W_2^*$  while maintaining comfort. Note that, despite the fact that comfort is not a safety specification (in the sense of Section 2), the set of comfortable controls can still be characterized in feedback form.

## 6. Concluding remarks and computational issues

The notions of “maximal controlled invariant set” and “least restrictive safe controller” are central to the controller synthesis methodology discussed in this paper. These notions allow us to deal with the multi-objective nature of the problem by solving a sequence of nested two player, zero sum games. These notions are also important in the hierarchical control context. Assume that a number of controllers are synthesized (using the methodology introduced here for example), each designed to deal with a particular situation, and we are asked to develop a discrete supervisor to switch between them. The maximal safe sets for each controller provide necessary enabling conditions for the transitions of the supervisor; a particular controller should be invoked only if the current value of the state lies in the corresponding safe set.

In the examples considered here the maximal safe sets and least restrictive safe controllers naturally emerged from the calculations. We would like to develop a formal



methodology to capture this procedure. The dynamic programming techniques used in the last example (FVMS) seem to be the most promising in this respect. In Tomlin et al. (1998) we show how in certain cases these techniques can be extended to hybrid systems. An interesting feature of the results presented there is that the discrete and continuous calculations effectively decouple (which is also the case for the FVMS example). We are currently working on generalizing this approach to more technically challenging cases (for example, safe sets that are not smooth).

The methods presented in this paper may also prove useful for the introduction of new controllers into so-called *legacy systems* for real time control. Legacy systems come equipped with a controller with a guaranteed domain of validity, say  $W_1$ .<sup>6</sup> Assume one would like to retrofit the system with a new experimental controller with unknown domain of validity, presumably in an attempt to improve performance. This addition should be done in a way that does not compromise the safety of the system. One way of accomplishing this is to utilize the experimental controller only in the interior of the validity set  $W_1$  and resort to the legacy controller as soon as the state approaches the boundary of  $W_1$ . Our methods are useful for systematically computing the switching logic among the controllers and determining the switching boundaries. In this context, it may be useful to explore the links between our work and the Simplex work of Seto et al. (1998), on safely executing control code upgrades for complex systems.

In practice, the usefulness of the algorithm for hybrid controller synthesis presented in this paper depends on our ability to efficiently compute solutions the Hamilton–Jacobi equation. We conclude this paper with a brief discussion of some of the computational issues which we are currently investigating.

Numerical methods for computing solutions to the Hamilton–Jacobi partial differential equation have been studied extensively. Sethian (1996) presents a set of computation schemes based on a *level set method* for propagating curves, which uses numerical techniques derived from conservation laws. The approach requires gridding the state space, so while these techniques have been shown to be efficient in two or three dimensions, they may become computationally intractable in higher dimensions. Also, it is essential that a bound on the error due to approximation be known at each step of the algorithm, in order to guarantee that the computed surface is a conservative approximation to the actual surface.

Numerical solutions are potentially complicated by the fact that the right-hand side of Eq. (18) is nonsmooth. This is also the case for the optimal Hamiltonian  $H^*(x, p)$ . Moreover, as  $t$  evolves the solution  $J^*(x, t)$  to the Hamilton–Jacobi equation can develop discontinuities (known as *shocks*) or lose smoothness as a function of  $x$ . Finally, additional numerical complications may be generated if the “capture” set happens to be nonsmooth. Computing solutions with discontinuous Hamiltonian functions is dealt with in Sethian (1996) using an evolution function which varies across the grid space. Methods to compute solutions in the presence of shocks are presented in Berg et al. (1996), and a “viscosity” method to systematically deal with shocks is presented in Lions (1982).

Even if all of the issues associated with computing solutions to the continuous Hamilton–Jacobi equation are resolved, there are more computational challenges arising from the discrete dynamics. The first is *undecidability*; in general, one cannot expect to solve Eq. (18) 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 (Henzinger et al., 1995). Techniques have been proposed to resolve this problem, making use of approximation schemes to obtain estimates of the solution (Puri and Varaiya, 1995). Another problem is the requirement that the controller resulting from our algorithm is *nonZero*, i.e. does not enforce the safety requirement by preventing time from diverging. The algorithm proposed here has no way of preventing such behavior. Adding this requirement to our controllers is likely to be a major challenge, as it involves reasoning with infinite executions and liveness specifications, that we have not considered in this paper. This is the subject of on-going research.

## Acknowledgements

Research supported by the Office of Naval Research under grant N0014-97-1-0946, by the PATH program under MOU-238, MOU-288 and MOU-319, by NASA under grant NAG 2-1039 and by ZONTA and NSERC Postgraduate Fellowships.

## Appendix A. Additional proofs

**Proof of Proposition 2.** The if part is obvious. For the only if part, assume that there exists a controller  $C$  that solves the synthesis problem  $(H, \square F)$ , but there does not exist a feedback controller that solves the synthesis problem. Therefore, there must exist  $x \in F$  and two different finite executions  $\chi_1 = (\tau_1, x_1, (u_1, d_1)) \in \mathcal{H}_C$  and  $\chi_2 = (\tau_2, x_2, (u_2, d_2)) \in \mathcal{H}_C$  ending in  $x$  such that  $C(x_1) \neq C(x_2)$ . Moreover, the “information” about

<sup>6</sup>Actually, the legacy controllers frequently involve state based switching between controllers designed for “safety” and “performance”, not unlike the systems discussed here.

whether  $x$  was reached via  $\chi_1$  or whether it was reached via  $\chi_2$  must be essential for subsequent control decisions.

More formally, assume  $x$  is reached via  $\chi_2$ , and let  $\chi'$  denote a subsequent execution, that is assume that the concatenation  $\chi_2\chi'$  belongs to  $\mathcal{H}$ . Note that, since  $\chi_1$  also ends in  $x$ ,  $\chi_1\chi'$  also belongs to  $\mathcal{H}$ . Let  $\chi_2\chi' = (\tau'_2, x'_2, (u'_2, d'_2))$  and  $\chi_1\chi' = (\tau'_1, x'_1, (u'_1, d'_1))$ . Assume that for all  $t \in \tau'_2 \setminus \tau_2$ , a control  $u(t) \in C(x'_1 \downarrow_t)$  is applied (instead of a control  $u(t) \in C(x'_2 \downarrow_t)$ ). Then, as the fact that  $x$  was reached via  $\chi_2$  is essential, there must exist a subsequent execution  $\chi'$  such that  $\chi_2\chi' \in \mathcal{H}$  (in fact  $\chi_2\chi' \in \mathcal{H} \setminus \mathcal{H}_C$ ) and  $\square F(\chi_2\chi') = \text{False}$ . This implies that there exists  $t \in \tau'_2$  such that  $x'_2(t) \in F^c$ . Since  $C$  is assumed to solve the synthesis problem and  $\chi_2 \in \mathcal{H}_C$ ,  $\square F(\chi_2) = \text{True}$ , therefore  $t \in \tau'_2 \setminus \tau_2$ .

However, since for all  $t \in \tau'_2 \setminus \tau_2$ ,  $u(t) \in C(x'_1 \downarrow_t)$ , and  $(\tau'_1, x'_1, (u'_1, d'_1)) \in \mathcal{H}$ , we have that  $\chi_1\chi' \in \mathcal{H}_C$ . But the above discussion indicates that there exists  $t \in \tau'_1$  (in fact  $t \in \tau'_1 \setminus \tau_1$ ) such that  $x'_1(t) \in F^c$ . This contradicts the assumption that  $C$  solves the synthesis problem  $(H, \square F)$ .  $\square$

**Proof of Proposition 3.** If there exists any control invariant  $W \subseteq F$  (in particular, if there exists a unique maximal one) then, by definition, the synthesis problem  $(H, \square F)$  can be solved for  $I = W$ .

For the only if part, if the solution to the problem is not “None” for all  $I$ , there exists a set  $\hat{I}$  and a feedback controller  $g$  such that for all  $d$  and for all  $x^0 \in \hat{I}$  the execution  $(\tau, x, (u, d))$  with  $u(t) \in g(x(t))$  for all  $t \in \tau$  satisfies  $x(t) \in F$  for all  $t \in \tau$ . Consider the set

$$W = \bigcup_d \bigcup_{x^0 \in \hat{I}} \bigcup_{t \in \tau} x(t).$$

Then clearly  $W \subseteq F$ . Moreover, for any  $x^0 \in W$  consider the execution  $(\tau, x, (u, d))$  with arbitrary  $d \in \mathcal{D}$  and  $u(t) \in g(x(t))$ . Then, by definition of  $W$ ,  $x(t) \in W$  for all  $t \in \tau$ . Therefore, controller  $g$  renders the set  $W$  invariant.

Having established the existence of controlled invariant subsets of  $F$ , consider now two such sets  $W_1 \subseteq F$  and  $W_2 \subseteq F$ . We show that their union is also a controlled invariant subset of  $F$ . Clearly  $W_1 \cup W_2 \subseteq F$ . For  $i = 1, 2$ , as  $W_i$  is controlled invariant, there exists a feedback controller  $g_i$  that solves the controller synthesis problem  $(H, \square W_i)$ , with  $I = W_i$ . Consider the feedback controller  $g$  with

$$g(x) = \begin{cases} g_1(x) & \text{if } x \in W_1, \\ g_2(x) & \text{otherwise.} \end{cases}$$

Consider an arbitrary  $x^0 \in W_1 \cup W_2$ . Then either  $x^0 \in W_1$  or  $x^0 \in (W_1 \cup W_2) \setminus W_1 \subseteq W_2$ . In the first case, all executions are guaranteed to satisfy  $\square W_1$  as  $g_1$  renders  $W_1$  invariant. For the second case, consider an arbitrary execution  $\chi = (\tau, x, (u, d))$  with  $u(t) \in g(x(t))$  for all  $t \in \tau$ . Since  $g_2$  solves the controller synthesis problem  $(H, \square W_2)$  with  $I = W_2$ , either  $\square(W_2 \setminus W_1)(\chi) = \text{True}$  or

$x \in W_2 \setminus W_1$  until  $x \in W_1$ , which brings us back to the first case. Hence,  $g$  solves the controller synthesis problem  $(H, \square(W_1 \cup W_2))$  with  $I = W_1 \cup W_2$ , and the set  $W_1 \cup W_2$  is controlled invariant.

Summarizing, the class of controlled invariant subsets of  $F$  is closed under union. Hence, it possesses a unique maximal element.  $\square$

**Proof of Proposition 6.** We show that  $W^i = \{x \in \mathbf{X} \mid J(x, i) = 1\}$ . The proof is by induction.  $W^0 = \{x \in \mathbf{X} \mid J(x, 0) = 1\}$  by the definition of  $J(x, 0)$ . We assume that the claim holds for  $i = k$  and show that it also holds for  $i = k - 1$ . Then, by induction  $W^i = \{x \in \mathbf{X} \mid J(x, i) = 1\}$  for all  $i \in \mathbb{Z}_-$ . The proposition follows by the properties of the algorithm.

Consider an arbitrary state  $x$ . If  $J(x, k) = 0$  then

$$J(x, k - 1) = \min \left\{ 0, \max_u \min_d \min_{x' \in \delta(x, u, d)} J(x', k) \right\} = 0$$

as  $J(x', k) \in \{0, 1\}$ . Therefore,  $\{x \in \mathbf{X} \mid J(x, k - 1) = 1\} \subseteq W^k$ . If  $J(x, k) = 1$ , then

$$J(x, k - 1) = \min \left\{ 0, \max_u \min_d \left[ \min_{x' \in \delta(x, u, d)} J(x', k) - 1 \right] \right\} + 1.$$

Therefore,  $J(x, k - 1) = 0$  if  $\max_u \min_d \min_{x' \in \delta(x, u, d)} J(x', k) = 0$ , and  $J(x, k - 1) = 1$  if  $\max_u \min_d \min_{x' \in \delta(x, u, d)} J(x', k) = 1$ . Parsing the expressions (using the standard game theoretic conventions) reveals that

$$\max_u \min_d \min_{x' \in \delta(x, u, d)} J(x', k) = 1$$

is equivalent to “there exists a  $u$  such that for all  $d$  and for all  $x' \in \delta(x, u, d)$ ,  $J(x', k) = 1$ ”, or, in other words “there exists  $u$  such that for all  $d$ ,  $\delta(x, u, d) \subseteq W^k$ ”. Similarly,

$$\max_u \min_d \min_{x' \in \delta(x, u, d)} J(x', k) = 0$$

is equivalent to “for all  $u$  there exists  $d$  such that  $\delta(x, u, d) \cap (W^k)^c \neq \emptyset$ ”. The claim follows.  $\square$

**Proof of Lemma 9.** We check that the following inequalities hold for all  $x^0, u_1, u_2$  and  $d$ :

$$\begin{aligned} J_1(x^0, u_1, u_2, d^{1*}) &\leq J(x^0, u_1^{1*}, u_2^{1*}, d^{1*}) \\ &\leq J_1(x^0, u_1^{1*}, u_2^{1*}, d). \end{aligned}$$

Let  $x^*(t)$  denote the state evolution under the candidate saddle inputs. Then:

$$q_i^*(t) = \begin{cases} 0 & \text{if } t \leq T_{p_i} - T_i^0, \\ P_i & \text{if } t \geq T_{p_i} - T_i^0, \end{cases}$$

$$r^*(t) = \begin{cases} r^0 + U_1 t & \text{if } t \leq T_r, \\ W & \text{if } t \geq T_r, \end{cases} \quad T_i^* = T_i^0 + t,$$

$$w^*(t) = w^0 + \begin{cases} -\frac{U_1 t^2}{2} - r^0 t & \text{if } t \leq T_r \wedge t \leq D_1 \wedge t \leq D_2, \\ -W(t - T_r) - \frac{U_1 T_r^2}{2} - r^0 T_r & \text{if } t \geq T_r \wedge t \leq D_1 \wedge t \leq D_2, \\ P_1(t - D_1) - \frac{U_1 t^2}{2} - r^0 t & \text{if } t \leq T_r \wedge t \geq D_1 \wedge t \leq D_2, \\ P_1(t - D_1) - W(t - T_r) - \frac{U_1 T_r^2}{2} - r^0 T_r & \text{if } t \geq T_r \wedge t \geq D_1 \wedge t \leq D_2, \\ P_2(t - D_2) - \frac{U_1 t^2}{2} - r^0 t & \text{if } t \leq T_r \wedge t \leq D_1 \wedge t \geq D_2, \\ P_2(t - D_2) - W(t - T_r) - \frac{U_1 T_r^2}{2} - r^0 T_r & \text{if } t \geq T_r \wedge t \leq D_1 \wedge t \geq D_2, \\ P_1(t - D_1) + P_2(t - D_2) - \frac{U_1 t^2}{2} - r^0 t & \text{if } t \leq T_r \wedge t \geq D_1 \wedge t \geq D_2, \\ P_1(t - D_1) + P_2(t - D_2) - W(t - T_r) - \frac{U_1 T_r^2}{2} - r^0 T_r & \text{if } t \geq T_r \wedge t \geq D_1 \wedge t \geq D_2, \end{cases}$$

where  $T_r = (W - r^0)/U_1$  is the time at which the steam rate reaches its maximum value under disturbance  $d(t) = U_1$  and  $D_i = T_{p_i} - T_i^0$  is the time at which pump  $i$  starts pumping water under input  $u_i^{1*}(t)$ . Even though there are eight possible expressions for  $w(t)$ , once the initial condition (and hence  $D_1, D_2$  and  $T_r$ ) is fixed only four of them need concern us. In subsequent proofs we refer to these eight expressions as “expressions (1)–(8)”.

First fix  $u_i = u_i^{1*}$  and allow  $d$  to vary. Let  $x(t)$  denote the state evolution starting at  $x^0$  under  $(u_1^{1*}, u_2^{1*}, d)$ . Then

$$w(t) = w^*(t) - \int_0^t (r(t') - r^*(t')) dt'$$

But, under the assumed constraints on  $r$  and  $d$ ,  $r(t) \leq r^*(t)$  for all  $t$ . Therefore  $w(t) \geq w^*(t)$  for all  $t$  and  $J_1^1(x^0, u_1^{1*}, u_2^{1*}, d) = \inf_{t \geq 0} w(t) \geq J_1^1(x^0, u_1^{1*}, u_2^{1*}, d^{1*})$ .

Now fix  $d = d^{1*}$  and allow  $u_1$  and  $u_2$  to vary. Again let  $x(t)$  denote the state evolution starting at  $x^0$  under  $(u_1, u_2, d^{1*})$ . The constraints on the switching imply that  $q_i^*(t) \geq q_i(t)$  for all  $t \geq 0$ . But

$$w(t) = w^*(t) - \int_0^t (q_1^*(t') + q_2^*(t') - q_1(t') - q_2(t')) dt'.$$

Therefore,  $w(t) \leq w^*(t)$  and  $J_1^1(x^0, u_1, u_2, d^{1*}) = \inf_{t \geq 0} w(t) \leq J_1^1(x^0, u_1^{1*}, u_2^{1*}, d^{1*})$ .  $\square$

**Proof of Lemma 10.** Let  $x^*(t)$  denote the state evolution under the candidate saddle inputs. As there is no delay in switching the pumps off:

$$q_i^*(t) = 0 \quad \text{for all } t \quad T_i^* = 0 \quad \text{for all } t,$$

$$w^{2*}(t) = w^0 + \begin{cases} \frac{U_2 t^2}{2} - r^0 t & \text{if } t \leq \frac{r^0}{U_2}, \\ -\frac{(r^0)^2}{2U_2} & \text{if } t \geq \frac{r^0}{U_2}, \end{cases}$$

$$r^*(t) = \begin{cases} r^0 - U_2 t & \text{if } t \leq \frac{r^0}{U_2}, \\ 0 & \text{if } t \geq \frac{r^0}{U_2}. \end{cases}$$

As above, first fix  $u$  and allow  $d$  to vary. The resulting state trajectory will satisfy  $r(t) \geq r^*(t)$  and therefore  $w(t) \leq w^*(t)$  for all  $t$ . Hence,  $J_1^2(x^0, u_1^{2*}, u_2^{2*}, d) \geq J_1^2(x^0, u_1^{2*}, u_2^{2*}, d^{2*})$ . Likewise, if we fix  $d = d^{2*}$  and allow  $u$  to vary, the resulting trajectory will satisfy

$q_i(t) \geq q_i^*(t)$  and therefore  $w(t) \geq w^*(t)$  for all  $t$ . Hence,  $J_1^2(x^0, u_1, u_2, d^{2*}) \leq J_1^2(x^0, u_1^{2*}, u_2^{2*}, d^{2*})$ .  $\square$

**Proof of Lemma 11.** Consider the derivative of the water level  $\dot{w}^*$ , obtained by differentiating expressions (1)–(8) in the proof of Lemma 9. Without loss of generality we assume that  $D_1 \leq D_2$  and distinguish the following cases:

*Case 1:* If  $W > P_1 + P_2$ , then, for  $t$  large enough (in particular  $t \geq \max\{T_r, D_1, D_2\}$ ),  $\dot{w}^*(t) = P_1 + P_2 - W < 0$  therefore  $w^* \rightarrow -\infty$ . Clearly in this case a game-winning strategy does not exist for  $u_i$  for any initial condition, as  $d$  can always force the water to drop below any level.

*Case 2:* If  $W \leq P_1 + P_2$  we can distinguish three further cases

*Case 2.1:* If  $W \leq \min\{P_1, P_2\}$ , then  $\dot{w}^*(t) < 0$  if  $t \leq \min\{D_1, D_2\}$  and  $\dot{w}^*(t) \geq 0$  if  $t \geq \min\{D_1, D_2\}$ . Therefore,  $J_1^*(x^0) = w^*(\min\{D_1, D_2\})$ .

*Case 2.2:* If  $P_1 \leq W \leq P_2$ , then

$$\dot{w}^*(t) \begin{cases} < 0 & \text{if } t \leq D_1, \\ \geq 0 & \text{if } D_1 \leq t \leq D_2 \wedge t \leq \frac{P_1 - r^0}{U_1}, \\ < 0 & \text{if } D_1 \leq t \leq D_2 \wedge t > \frac{P_1 - r^0}{U_1}, \\ \geq 0 & \text{if } t \geq D_2. \end{cases}$$

Therefore,  $J_1^*(x^0) = \min\{w^*(D_1), w^*(D_2)\}$ . By symmetry, the same will be true if  $P_2 \leq W \leq P_1$ .

*Case 2.3:* If  $\max\{P_1, P_2\} \leq W$ , then:

$$\dot{w}^*(t) \begin{cases} < 0 & \text{if } t \leq D_1, \\ \geq 0 & \text{if } D_1 \leq t \leq D_2 \wedge t \leq \frac{P_1 - r^0}{U_1}, \\ < 0 & \text{if } D_1 \leq t \leq D_2 \wedge t > \frac{P_1 - r^0}{U_1}, \\ \geq 0 & \text{if } t \geq D_2. \end{cases}$$

Again,  $J_1^*(x^0) = \min\{w^*(D_1), w^*(D_2)\}$ .

Overall, if we restrict our attention to Case 2 (where there is some hope that the system will be safe), the above relations indicate that

$$J_1^*(x^0) = \min\{w^*(D_1), w^*(D_2)\}.$$

An analytical expression for  $J_1^*$  can be obtained from expressions (1)–(8) (proof of Lemma 10).  $\square$

**Proof of Theorem 15.** We first prove that the set  $\bigcap_{i \in \{1, 2, 3, 4\}} \{x | J_1^{i*}(x) \geq 0\}$  is exactly the construction described by Eq. (38). We then prove that this set is equal to  $W_1^*$ , the maximal controlled invariant set contained in  $F$ .

Consider first the edge  $\{x \in F | l_1^1(x) = 0\}$  in  $\partial F$ . We prove that  $\{x \in F | J_1^{1*}(x) = 0\}$  is equal to  $\{x | (V = V_{\min}) \wedge (\gamma_{\min} \leq \gamma \leq \gamma_a)\} \cup \partial W_1^a$ . The optimal Hamiltonian  $H_1^{1*}(x, p)$  satisfies

$$H_1^{1*}(x, p) \begin{cases} < 0, & x \in F \wedge l_1^1(x) = 0 \wedge \gamma > \gamma_a, \\ = 0, & x \in F \wedge l_1^1(x) = 0 \wedge \gamma = \gamma_a, \\ > 0, & x \in F \wedge l_1^1(x) = 0 \wedge \gamma < \gamma_a. \end{cases} \quad (\text{A.1})$$

Thus, the set  $\{x | (V = V_{\min}) \wedge (\gamma_{\min} \leq \gamma \leq \gamma_a)\}$  remains unchanged under the evolution of the Hamilton–Jacobi equation (35), since  $H_1^{1*}(x, \partial J_1^{1*}(x, t) / \partial x) > 0$  for this set. We now prove that for  $x \in \partial W_1^a$ ,  $J_1^{1*}(x) = 0$ .  $J_1^{1*}(x)$  satisfies

$$\left( \frac{\partial J_1^{1*}(x)}{\partial x} \right) f(x, u^*) = 0, \quad (\text{A.2})$$

where  $\partial J_1^{1*}(x) / \partial x$  is the inward pointing normal to  $\{x | J_1^{1*}(x) = 0\}$ . At each  $x$  in  $\{x | J_1^{1*}(x) = 0\}$ ,  $f(x, u^*)$  is tangent to  $\{x | J_1^{1*}(x) = 0\}$ . Thus, the solution  $x(t)$  to  $\dot{x} = f(x, u^*)$  evolves along  $J_1^{1*}(x) = 0$ . By construction,  $x \in \partial W_1^a$  satisfies  $J_1^{1*}(x) = 0$ .

Repeating this analysis for  $\{x \in F | l_1^3(x) = 0\}$ , we can prove that  $\{x \in F | J_1^{3*}(x) = 0\}$  is equal to  $\{x | (V = V_{\max}) \wedge (\gamma_b \leq \gamma \leq \gamma_{\max})\} \cup \partial W_1^b$ . On the remaining boundaries,  $H_1^{2*}(x, p) > 0$  and  $H_1^{4*}(x, p) > 0$ , so these boundaries remain unchanged under the evolution of their respective Hamilton–Jacobi equations.

It remains to prove that  $W_1^* = \bigcap_{i \in \{1, 2, 3, 4\}} \{x | J_1^{i*}(x) \geq 0\}$ . Clearly, any state  $x$  for which there exists an  $i$  such that  $J_1^{i*}(x) < 0$  must be excluded from  $W_1^*$ , since a trajectory exists which starts from this state and drives the system out of  $\bigcap_{i \in \{1, 2, 3, 4\}} \{x | J_1^{i*}(x) \geq 0\}$ . Thus,  $W_1^* \subset \bigcap_{i \in \{1, 2, 3, 4\}} \{x | J_1^{i*}(x) \geq 0\}$ . To prove equality, we need only show that at the points of intersection of the four boundaries,  $(V_a, \gamma_{\max})$ ,  $(V_{\max}, \gamma_{\max})$ ,  $(V_b, \gamma_{\min})$ ,  $(V_{\min}, \gamma_{\min})$  there exists a control input which keeps the system state inside  $\bigcap_{i \in \{1, 2, 3, 4\}} \{x | J_1^{i*}(x) \geq 0\}$ . Consider the point  $(V_a, \gamma_{\max})$ . At this point, the set of control inputs which keeps the system state inside the set  $\{x | J_1^{1*}(x) \geq 0\}$  is  $\{(T_{\max}, \theta_{\min})\}$ , and the set of control inputs which keeps the system state inside  $\{x | J_1^{2*}(x) \geq 0\}$  is the set  $\{(T, \theta) | T \in [T_{\min}, T_{\max}], \theta \in [\theta_{\min}, \frac{m}{a_l V_a c} (\frac{g \cos \gamma_{\max}}{V_a} - \frac{a_l V_a (1 - c \gamma_{\max})}{m})]\}$ . Since these two sets have nonempty intersection, the intersection point  $(V_a, \gamma_{\max}) \in W_1^*$ . Similar analysis holds for the remaining three intersection points. Thus  $W_1^* = \bigcap_{i \in \{1, 2, 3, 4\}} \{x | J_1^{i*}(x) \geq 0\}$ .  $\square$

**Proof of Theorem 16.** Consider the left side of  $\partial F$ . For each  $x$  in  $\{x \in F | l_1^1(x) = 0\}$ , denote by  $(T_a(\gamma), \theta_a(\gamma))$  the values of  $(T, \theta)$  for which the vector field  $f(x, [T, \theta]^T)$  becomes tangent to  $l_1^1(x) = 0$  (i.e.  $\dot{V} = 0$ ). Setting  $\dot{V} = 0$  leads to Eq. (40), for all  $\theta_a(\gamma) \in [\theta_{\min}, \theta_{\max}]$ . Therefore, the safe set of inputs along  $\{x \in F | l_1^1(x) = 0\}$  are all  $T \in [T_{\min}, T_{\max}]$  with  $T \geq T_a(\gamma)$  and all  $\theta \in [\theta_{\min}, \theta_{\max}]$ . At the point  $(V_{\min}, \gamma'_a)$ , where  $\gamma'_a = \{\gamma | T_a(\gamma) = T_{\min}\}$  the cone of vector fields  $f([V_{\min}, \gamma'_a], U)$  points completely inside  $F$ . At  $\gamma_a = \{\gamma | T_a(\gamma) = T_{\max}\}$  the cone of vector fields points completely outside  $F$ , and  $T_{\max}$  is the unique thrust which keeps the system trajectory tangent to  $F$ .

The calculation may be repeated for the right side of  $\partial F$ :  $\{x \in F | l_1^3(x) = 0\}$ . Here, let  $T_b(\gamma)$  be the value of the input thrust for which  $f(x, [T_b(\gamma), \theta]^T)$  is tangent to  $l_1^3(x) = 0$ , thus  $T_b(\gamma)$  is given by Eq. (41). The safe set of inputs along  $\{x \in F | l_1^3(x) = 0\}$  are all  $T \in [T_{\min}, T_{\max}]$  with  $T \leq T_b(\gamma)$  and all  $\theta \in [\theta_{\min}, \theta_{\max}]$ . At the point  $(V_{\max}, \gamma_b)$ , where  $\gamma_b = \{\gamma | T_b(\gamma) = T_{\min}\}$ ,  $T_{\min}$  is the unique thrust which keeps the system trajectory tangent to  $F$  (lower right boundary of the safe set).

Similar calculations along the upper and lower sides of  $\partial F$  yield that the values of  $\theta$  for which the vector field becomes tangent to  $\partial F$  are  $\theta_c(V)$  and  $\theta_d(V)$  of Eqs. (42) and (43).  $\square$

## References

- Abrial, J.-R. (1996). The steam-boiler control specification problem. In J.-R. Abrial, E. Börger, & H. Langmaack (Eds.), *Formal methods for industrial applications: Specifying and programming the steam boiler control*, Lecture Notes in Computer Science, (Vol. 1165). Berlin: Springer.
- Alur, R., Courcoubetis, C., Henzinger, T. A., & Ho, P. H. (1993). Hybrid automaton: An algorithmic approach to the specification and verification of hybrid systems. In R. L. Grossman, A. Nerode, A. P. Ravn, & H. Rischel (Eds.), *Hybrid systems*, (Vol. 736, pp. 209–229). New York: Springer.
- Alur, R., & Dill, D. (1994). A theory of timed automata. *Theoret. Comput. Sci.*, 126, 183–235.
- Alur, R., & Henzinger, T. A. (1996). Reactive modules. *Proc. 11th Ann. Symp. on Logic in Computer Science*, (pp. 207–218). Silver Spring, MD: IEEE Computer Society Press.
- Anderson, M., Bruck, D., Mattsson, S. E., & Schonthal, T. (1994). Omsim- an integrated interactive environment for object-oriented modeling and simulation. *IEEE/IFAC Joint Symp. on Computer Aided Control System Design*. (pp. 285–290).
- Aubin, J.-P. (1991). *Viability theory*. Boston: Birkhäuser.
- Başar, T., & Olsder, G. J. (1995). *Dynamic non-cooperative game theory* (2nd ed.). New York: Academic Press.
- Berg, J. M., Yezzi, A., & Tannenbaum, A. R. (1996). Phase transitions, curve evolution, and the control of semiconductor manufacturing processes. *IEEE Conference on Decision and Control* (pp. 3376–3381). Kobe, Japan, 11–13 December.
- Branicky, M., Dolginova, E., & Lynch, N. (1997). A toolbox for proving and maintaining hybrid specifications. In A. Nerode P. Antsaklis, W. Kohn, & S. Sastry (Eds.), *Hybrid systems IV*, Lecture Notes in Computer Science (Vol. 1273, pp. 18–30). Berlin: Springer.
- Branicky, M. S. (1998). Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Automat. Control*, 43(4), 475–482.

- Branicky, M. S., Borzkar, V. S., & Mitter, S. K. (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Trans. Automat. Control*, 43(1), 31–45.
- Brockett, R. W. (1993). Hybrid models for motion control systems. In H. L. Trentelman, & J. C. Willems (Eds.), *Perspectives in control*. Basel: Birkhäuser.
- Bryson, A. E., & Ho, Y.-C. (1975). *Applied optimal control*. Washington, DC: Hemisphere.
- Büchi, J. R., & Landweber, L. H. (1969). Solving sequential conditions by finite-state operators. *Proc. American Mathematical Society* (pp. 295–311).
- Daws, C., Olivero, A., & Yovine, S. (1994). Verifying ET-LOTOS programs with KRONOS. In D. Hogrefe, & S. Leue (Eds.), *Proceedings 7th. IFIP WG G.1 International Conference of Formal Description Techniques, FORTE'94* (pp. 227–242). Bern, Switzerland, October 1994. Formal Description Techniques VII, London: Chapman & Hall.
- Deshpande, A. (1994). *Control of hybrid systems*. Ph.D. thesis, Department of Electrical Engineering, University of California, Berkeley.
- Deshpande, A., Gollu, A., & Semenzato, L. (1997). The SHIFT programming language and run-time system for dynamic networks of hybrid automata. Technical Report UCB-ITS-PRR-97-7, Institute of Transportation Studies, University of California, Berkeley.
- Heitmeyer, C., & Lynch, N. (1994). The generalized railroad crossing: A case study in formal verification of real-time systems. *Proc. ICCR Real-Time Systems Symp.*, San Juan, Puerto Rico.
- Henzinger, T., Kopke, P., Puri, A., & Varaiya, P. (1995a). What's decidable about hybrid automata. In *27th Annual Symposium on the Theory of Computing, STOC'95* (pp. 373–382). New York: ACM Press.
- Henzinger, T. A., Ho, P. H., & Wong Toi, H. (1995b). A user guide to HYTECH. In E. Brinksma, W. Cleaveland, K. Larsen, T. Margaria & B. Steffen (Eds.), *TACAS 95: Tools and algorithms for the construction and analysis of systems* (Vol. 1019, pp. 41–71). Berlin: Springer.
- Henzinger, T. A., & Wong-Toi, H. (1996). Using HYTECH to synthesize control parameters for a steam boiler. In J.-R. Abrial, E. Börger, & H. Langmaack (Eds.), *Formal methods for industrial applications: Specifying and programming the steam boiler control*, Lecture Notes in Computer Science (Vol. 1165, pp. 265–282). Berlin: Springer.
- Heymann, M., Lin, F., & Meyer, G. (1997). Control synthesis for a class of hybrid systems subject to configuration-based safety constraints. In *Hybrid and real time systems*, Lecture Notes in Computer Science (Vol. 1201, pp. 376–391). Berlin: Springer.
- Hynes, C.S., & Sherry, L. (1996). Synthesis from design requirements of a hybrid system for transport aircraft longitudinal control. preprint, NASA Ames Research Center.
- Kurshan, R. P. (1994). *Computer-aided verification of coordinating processes; the automata-theoretic approach*. Princeton, NJ: Princeton University Press.
- Lemmon, M., Stiver, J. A., & Antsaklis, P. J. (1993). Event identification and intelligent hybrid control. In R. L. Grossman, A. Nerode, A. P. Ravn, & H. Rischel (Eds.), *Hybrid systems*, Lecture Notes in Computer Science (Vol. 736, pp. 268–296). New York: Springer.
- Lions, P. L. (1982). *Generalized solutions of Hamilton–Jacobi equations*. London: Pitman.
- Lygeros, J. (1996c). *Hierarchical hybrid control of large scale systems*. Ph.D. thesis, Department of Electrical Engineering, University of California, Berkeley.
- Lygeros, J., Godbole, D. N., & Sastry, S. (1996b). A game theoretic approach to hybrid system design. *Hybrid systems III*, Lecture Notes in Computer Science (Vol. 1066, pp. 1–12). Berlin: Springer.
- Lygeros, J., Godbole, D. N., & Sastry, S. (1996a). Multiagent hybrid system design using game theory and optimal control. In *IEEE Conf. on Decision and Control* (pp. 1190–1195). Kobe, Japan, 11–13 December.
- Lygeros, J., Godbole, D. N., & Sastry, S. (1996). Optimal control approach to multiagent, hierarchical system verification. In *IFAC World Congress* (pp. 389–394). San Francisco, CA, USA, 30 June–5 July.
- Lygeros, J., Godbole, D. N., & Sastry, S. (1998). Verified hybrid controllers for automated vehicles. *IEEE Trans. Automat. Control*, 43(4), 522–539.
- Lynch, N., Segala, R., Vaandrager, F., & Weinberg, H. B. (1996). Hybrid I/O automata. In *Hybrid systems III*, Lecture Notes in Computer Science (Vol. 1066, pp. 496–510). Berlin: Springer.
- Maler, O., Pnueli, A., & Sifakis, J. (1995). On the synthesis of discrete controllers for timed systems. In *Theoretical aspects of computer science*, Lecture Notes in Computer Science (Vol. 900, pp. 229–242). Berlin: Springer.
- Manna, Z., & Pnueli, A. (1992). *The temporal logic of reactive and concurrent systems: Specification*. Berlin: Springer.
- Manna, Z., & Pnueli, A. (1995). *Temporal verification of reactive systems: Safety*. New York: Springer.
- Nerode, A., & Kohn, W. (1993a). Models for hybrid systems: Automata, topologies, controllability, observability. In R. L. Grossman, A. P. Ravn & H. Rischel (Eds.), *Hybrid systems*, Lecture Notes in Computer Science (Vol. 736, pp. 317–356). New York: Springer.
- Nerode, A., & Kohn, W. (1993b). Multiple agent hybrid control architecture. In R. L. Grossman, A. Nerode, A. P. Ravn & H. Rischel (Eds.), *Hybrid systems*, Lecture Notes in Computer Science (Vol. 736, pp. 297–316). New York: Springer.
- Nicollin, X., Olivero, A., Sifakis, J., & Yovine, S. (1993). An approach to the description and analysis of hybrid systems. In R. L. Grossman, A. Nerode, A. P. Ravn & H. Rischel (Eds.), *Hybrid systems* Lecture Notes in Computer Science (Vol. 736, pp. 149–178). New York: Springer.
- Puri, A. (1995). *Theory of hybrid systems and discrete event systems*. Ph.D. thesis, Department of Electrical Engineering, University of California, Berkeley.
- Puri, A., & Varaiya, P. (1995). Verification of hybrid systems using abstractions. In *Hybrid Systems II* Lecture Notes in Computer Science (Vol. 999). Berlin: Springer.
- Ramadge, P. J. G., & Wonham, W. M. (1989). The control of discrete event systems. *Proc. IEEE*, 77(1), 81–98.
- Schwartz, A. L. (1996). *Theory and implementation of numerical methods based on Runge–Kutta integration for solving optimal control problems*. Ph.D. thesis, Department of Electrical Engineering, University of California, Berkeley.
- Sethian, J. A. (1996). *Level set methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*. New York: Cambridge University Press.
- Seto, D., Krogh, B., Sha, L., & Chutinan, A. (1998). Dynamic control system upgrade using the Simplex architecture. In *American Control Conf.* (pp. 3504–3508). Philadelphia, Pennsylvania, USA, 24–26 June.
- Tavernini, L. (1987). Differential automata and their simulators. *Nonlinear Anal. Theory Methods Appl.*, 11(6), 665–683.
- Thomas, W. (1995). On the synthesis of strategies in infinite games. In E. W. Mayr & C. Puech (Eds.), *Proc. STACS 95*. Lecture Notes in Computer Science (Vol. 900, pp. 1–13). Munich: Springer.
- Tomlin, C., Lygeros, J., & Sastry, S. (1998). Synthesizing controllers for nonlinear hybrid systems. In S. Sastry & T. A. Henzinger (Eds.), *Hybrid systems: Computation and control*, Lecture Notes in Computer Science (Vol. 1386, pp. 360–373). Berlin: Springer.
- Tomlin, C., Pappas, G., & Sastry, S. (1998). Conflict resolution for air traffic management: A case study in multi-agent hybrid systems. *IEEE Trans. Automat. Control*, 43(4), 509–521.
- Wong-Toi, H. (1997). The synthesis of controllers for linear hybrid automata. In *IEEE Conference on Decision and Control*, (pp. 4607–4613). San Diego, California, USA, 10–12 December.
- Ye, H., Michel, A., & Hou, L. (1998). Stability theory for hybrid dynamical systems. *IEEE Trans. Automat. Control*, 43(4), 461–474.



**John Lygeros** received his BEng degree in Electrical and Electronic Engineering in 1990 and his MSc degree in Control and Systems in 1991, both from Imperial College of Science Technology & Medicine, London, UK. In May 1996 he completed his PhD degree at the Electrical Engineering and Computer Sciences Department of the University of California, Berkeley. From June to October 1996 he was a visiting postdoctoral researcher with the National Automated Highway Systems Consortium, at the Institute of Transportation Studies, University of California, Berkeley. Between November 1996 and September 1997 he was a Postdoctoral Research Associate with the Laboratory for Computer Science at Massachusetts Institute of Technology. He is currently a postdoctoral researcher at the Electrical Engineering and Computer Sciences Department of University of California, Berkeley, and holds a part time Research Engineer position at SRI International. His research interests include hierarchical and hybrid systems, nonlinear control theory and their applications to Highway Systems and Air Traffic Management. Dr Lygeros is the recipient of the 1997 Eliahu Jury award “for excellence in systems research”, awarded by the Electrical Engineering and Computer Sciences Department of the University of California, Berkeley.

He is currently a postdoctoral researcher at the Electrical Engineering and Computer Sciences Department of University of California, Berkeley, and holds a part time Research Engineer position at SRI International. His research interests include hierarchical and hybrid systems, nonlinear control theory and their applications to Highway Systems and Air Traffic Management. Dr Lygeros is the recipient of the 1997 Eliahu Jury award “for excellence in systems research”, awarded by the Electrical Engineering and Computer Sciences Department of the University of California, Berkeley.



**S. Shankar Sastry** received his PhD degree in 1981 from the University of California, Berkeley. He was on the faculty of MIT from 1980–82 and Harvard University as a Gordon Mc Kay professor in 1994. He is currently a Professor of Electrical Engineering and Computer Sciences and Director of the Electronics Research Laboratory at Berkeley. He has held visiting appointments at the Australian National University, Canberra the University of

Rome, Scuola Normale and University of Pisa, the CNRS laboratory LAAS in Toulouse, and as a Vinton Hayes Visiting fellow at the Center for Intelligent Control Systems at MIT. His areas of research are nonlinear and adaptive control, robotic telesurgery, control of hybrid systems, and biological motor control. He is a coauthor (with M. Bodson) of “Adaptive Control: Stability, Convergence and Robustness,

Prentice Hall, 1989.” and (with R. Murray and Z. Li) of “A Mathematical Introduction to Robotic Manipulation, CRC Press, 1994”. His book “Nonlinear Control: Analysis, Stability and Control” is to be published by Springer Verlag in 1999. He has co-edited “Hybrid Control II” and “Hybrid Control IV” with P. Antsaklis, A. Nerode, and W. Kohn, and “Hybrid Systems: Computation and Control” with T. Henzinger Springer Lecture Notes in Computer Science, 1995, 1997 and 1998. Dr. Sastry was an Associate Editor of the *IEEE Transactions on Automatic Control*, *IEEE Control Magazine*, *IEEE Transactions on Circuits and Systems*, and the *Journal of Mathematical Systems, Estimation and Control* and is an Associate Editor of the *IMA Journal of Control and Information*, the *International Journal of Adaptive Control and Signal Processing* and the *Journal of Biomimetic Systems and Materials*.



**Claire Tomlin** received the BAsC degree in Electrical Engineering from the University of Waterloo, Canada, in 1992, the MSc degree in Electrical Engineering from Imperial College, University of London, in 1993, and the PhD degree in Electrical Engineering from the University of California, Berkeley, in 1998. Since September 1998 she has been an Assistant Professor in the Department of Aeronautics and Astronautics at Stanford University, with

a courtesy appointment in Electrical Engineering.

She was a graduate fellow in the Division of Applied Sciences at Harvard University in 1994, and she has been a visiting researcher at NASA Ames Research Center during 1994–1998, 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 in Paris, and at Bell-Northern Research and Gandalf Data Limited, both in Ottawa.

Claire 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 (1996–98), 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). Her research interests are in hybrid systems, nonlinear control systems, air traffic management, and flight vehicle dynamics and control.