Proceedings of the 36th
Conference on Decision & Control
San Diego, California USA • December 1997

Multi-Objective Hybrid Controller Synthesis: Least Restrictive Controls[1]

John Lygeros, Claire Tomlin and Shankar Sastry

Laboratory for Computer Science
Massachusetts Institute of Technology
545 Technology Square, Cambridge, MA 02139
lygeros@lcs.mit.edu

Intelligent Machines and Robotics Laboratory
University of California, Berkeley, CA 94720
clairet, sastry@eecs.berkeley.edu

Abstract

We present a methodology for synthesizing hybrid controllers that meet multiple control objectives. Our methodology uses game theoretic techniques to classify all controls that can be used to meet the high priority objectives. Lower priority objectives are then optimized within this class.

1 Introduction

Hybrid systems, that is systems that involve the interaction of discrete and continuous dynamics, have recently attracted considerable attention (for a discussion of research directions in this field see [1]). In this paper we address hybrid control problems, in particular ones where multiple requirements are imposed on the design. In such a multi-objective setting some of the requirements are usually assumed to be more important than others, either explicitly or implicitly. 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. Using optimal control tools we attempt to determine the *largest controlled invariant safe set*, i.e. the largest set of states for which there exists a control such that the safety objective can be met. In the process we also determine the *class of least restrictive safe controls*, i.e. all the controls that can be used to meet the safety objective from the safe states. The efficiency objective can then be optimized within this class. The resulting controller will typically be hybrid as it involves switching between the safe and efficient controllers.

Our analysis is based on the hybrid system modeling formalism introduced in [2]. The design algorithm (Section 2) is motivated by three examples. The first is purely discrete and involves the control of finite automata. The second is hybrid, a continuous process (the level of water in a boiler) is controlled using discrete controls (pumps being switched on and off). Finally, the third example is primarily continuous and is motivated by the design of a flight vehicle management system. The details of the calculations and the proofs have been omitted from the examples; the interested reader is referred to [1].

2 Multi-objective Controller Design

2.1 Modeling Framework

To introduce the necessary terminology and notation we briefly review the modeling framework of [2]. A ***hybrid dynamical system***, H, is a collection (X, U, Y, I, f, E, h), with $X = X_D \times X_C$, $U = U_D \times U_C$, $Y = Y_D \times Y_C$, $I \subset X$, $f : X \times U \rightarrow TX_C$, $E \subset X \times U \times X$, and $h : X \times U \rightarrow Y$. X_C, U_C, Y_C are respectively open subsets of $\mathbb{R}^n, \mathbb{R}^m, \mathbb{R}^p$, for some finite integers n, m, p. X_D, U_D, Y_D are countable sets and TX_C represents the tangent space of X_C. X, U and Y are referred to as the state, input and output variables respectively.

We consider the system evolution over a set of times of the form $T = [t_i, t_f] \subset \mathbb{R}$. Variables evolve either continuously as a function of time or in instantaneous jumps. Trajectories will therefore be defined over sets of the form $[\tau_0', \tau_1][\tau_1', \tau_2] \ldots [\tau_{n-1}', \tau_n]$ with $\tau_i \in T$ for all i, $\tau_0' = t_i, \tau_n = t_f$ and $\tau_i = \tau_i' \leq \tau_{i+1}$ for all $i = 1, 2, \ldots, n-1$. The implication is that τ_i are the times when discrete jumps occur. We will use $\mathcal{T}$ to denote the set of all such "super-dense" time trajectories over T and τ to denote an element of $\mathcal{T}$.

A *run* of the hybrid dynamical system H over an interval T is a collection (τ, q, x, y, u) with $\tau \in \mathcal{T}$, $q : \tau \rightarrow X_D$, $x : \tau \rightarrow X_C$, $y : \tau \rightarrow Y$, and $u : \tau \rightarrow U$, satisfying:

[1] Research supported by the ARO under grant DAAH 04-95-1-0588, by PATH, under MOU-238, and by NASA under grant NAG 2-1039.

1. $(q(\tau_0'), x(\tau_0')) \in I$.
2. For all i, either $(q(\tau_i), x(\tau_i), u(\tau_i), q(\tau_i'), x(\tau_i')) \in E$ and $(q(\tau_i), x(\tau_i)) \neq (q(\tau_i'), x(\tau_i'))$ or $u(\tau_i') \neq u(\tau_i)$.
3. For all i with $\tau_i' < \tau_{i+1}$ and for all $t \in [\tau_i', \tau_{i+1}]$, $\dot{x}(t) = f(q(t), x(t), u(t))$, $q(t) = q(\tau_i')$ and $(q(t), x(t), u(t), q(t), x(t)) \in E$.
4. For all $t \in \tau$, $y(t) = h(q(t), x(t), u(t))$.

Like conventional control systems, hybrid dynamical systems can be composed by input-output interconnections. It can be shown that, under some mild technical assumptions, an interconnection of hybrid dynamical systems is another hybrid dynamical system. To simplify the notation we will assume $Y = X$ and $h(q, x, u) = (q, x)$.

2.2 Controller Synthesis

We assume that we are given a plant modeled in the above framework, whose inputs are subdivided into two classes, controls denoted by u and disturbances, denoted by d. The input space is accordingly split into two subspaces, $U \times D$. The interpretation is that the designer can influence the controls but not the disturbances. This implies that the controller design should be such that the desired performance is achieved despite the actions of the disturbances. Let PC denote the space of piecewise continuous functions of the reals and define the set of *admissible controls* by $\mathcal{U} = \{u \in PC | u(t) \in U, \forall t\}$ and the set of *admissible disturbances* by $\mathcal{D} = \{d \in PC | d(t) \in D, \forall t\}$.

For simplicity, we restrict our attention to the case where two requirements are imposed on the system performance; we refer to them as *safety* and *efficiency*. We assume that these requirements can be encoded by a pair of cost functions, J_1 and J_2 respectively, on the runs of the hybrid dynamical system. Here we restrict our attention to the case where each (u, d) generates a unique state trajectory for a given initial condition (q^0, x^0). We informally refer to hybrid dynamical systems that possess this property as *deterministic hybrid dynamical systems*. In this case the cost function can be thought of as a map:

$$J_i : I \times \mathcal{U} \times \mathcal{D} \longrightarrow \mathbb{R} \tag{1}$$

We assume that a threshold, C_i, is given for each cost function and say that a trajectory, $((q^0, x^0), u, d)$, *meets objective* i if $J_i((q^0, x^0), u, d) \leq C_i$.

To guarantee that the performance 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 the cost functions J_1 and J_2. As higher priority is given to safety, we solve the game for J_1 first. Assume that the game admits a saddle solution, i.e. there exist input and disturbance trajectories, u_1^* and d_1^* such that:

$$\begin{aligned} J_1^*(q^0, x^0) &= \max_{d \in \mathcal{D}} \min_{u \in \mathcal{U}} J_1(q^0, x^0, u, d) \\ &= \min_{u \in \mathcal{U}} \max_{d \in \mathcal{D}} J_1(q^0, x^0, u, d) \\ &= J_1(q^0, x^0, u_1^*, d_1^*) \end{aligned}$$

Then the set $V_1 = \{(q, x) \in X | J_1^*(q, x) \leq C_1\}$ contains all states for which there exists a control such that the safety objective is met for the worst possible admissible disturbance (and hence for any admissible disturbance). If u_1^* is used as the control it will guarantee that J_1 is minimized for the worst possible disturbance; moreover, if the initial state is in V_1 it will also guarantee that the safety objective is met.

u_1^* does not take into account J_2, however. To introduce efficiency let:

$$\mathcal{U}_1(q^0, x^0) = \{u \in \mathcal{U} | \max_{d \in \mathcal{D}} J_1(q^0, x^0, u, d) \leq C_1\} \tag{2}$$

$\mathcal{U}_1$ can be thought of as a feedback map $\mathcal{U}_1 : X \rightarrow 2^{\mathcal{U}}$, that to each state assigns the subset of admissible controls which guarantee that the safety objective will be met; the *least restrictive class of safe controls*. Within this class we would like to select the control that minimizes the cost function J_2. We again pose the problem as a two person zero sum game. Assume that a saddle solution, (u_2^*, d_2^*) exists. Then the set $V_2 = \{(q, x) \in X | J_2(q, x, (u_2^*, d_2^*)) \leq C_2\}$ contains the initial conditions for which there exists a control such that for any admissible disturbance both safety and efficiency objectives are met. The control law u_2^* and the set V_2 are such that for all $(q^0, x^0) \in I \cap V_2$ and for all $d \in \mathcal{D}$, $J_i(q^0, x^0, u_2^*, d) \leq C_i$ for $i = 1, 2$.

As $V_2 \subset V_1$ there may still be states for which the safety objective can be met whereas the efficiency objective can not. If the saddle solutions are in feedback form, the controller can be extended to these states using the simple switching scheme:

$$u^*(q, x) = \begin{cases} u_2^*(q, x) & (q, x) \in V_2 \\ u_1^*(q, x) & (q, x) \in X \setminus V_2 \end{cases} \tag{3}$$

This will make the operation of the controller hybrid, even when the plant is purely continuous.

The above algorithm may run into technical difficulties, as there is no guarantee that the dynamic games will have a saddle solution, there is no straightforward way of computing $\mathcal{U}_1(q^0, x^0)$ and there is no guarantee that the sets V_1 (and consequently $\mathcal{U}_1(q^0, x^0)$) and V_2 will be non-empty. Fortunately, in the examples considered here (as well as the ones [3, 4]) solutions can be obtained analytically, or by using simple numerical calculations.

3 Reachability in Finite Automata

Consider a standard, deterministic finite automaton $G = (Q, \Sigma, \delta, Q_0)$ where Q is a finite set of states, Σ a finite set of events, $\delta : Q \times \Sigma \rightarrow Q$ a transition relation and $Q_0 \subset Q$ a set of initial states. Let $L(G)$ denote the strings of events (language) generated by G. Following [5] we assume that the set of events is partitioned into two subsets, $\Sigma = \Sigma_u \cup \Sigma_c$, where the events in Σ_c are controllable (in the sense that they can be disabled at will) while the events in Σ_u are uncontrollable. In this setting problems of safety are usually cast as questions of reachability: can the designer ensure that the automaton state will not enter a "bad" set $Q_B \subset Q$. Efficiency typically corresponds to questions of fairness or liveness.

We first write the finite automaton G as a hybrid dynamical system H. As there are no continuous variables, X_C, U_C, Y_C and f will be omitted. To ensure that H will not block for the saddle solutions (soon to be calculated) we add two new states, q_G and q_B, and define $X = Q \cup \{q_G, q_B\}$, $U = \Sigma_c \cup \{\epsilon\}$, $D = \Sigma_u \cup \{\epsilon\}$, $Q_B = Q_B \cup \{q_B\}$ and $I = I \cup \{q_G\}$. We then complete the transition relation by defining:

$$\begin{array}{rcll} E & = & \{(q_1,(d,u),q_2) \in X \times (D \times U) \times X | & \\ & & q_2 = \delta(q_1,d) & \text{if } q_1 \in Q, d \neq \epsilon, u = \epsilon, \delta(q_1,d)! \\ & & q_2 = q_G & \text{if } q_1 \in Q, d \neq \epsilon, u = \epsilon, \delta(q_1,d) \ !\!\!/ \\ & & q_2 = \delta(q_1,u) & \text{if } q_1 \in Q, d = \epsilon, u \neq \epsilon, \delta(q_1,u)! \\ & & q_2 = q_B & \text{if } q_1 \in Q, d = \epsilon, u \neq \epsilon, \delta(q_1,u) \ !\!\!/ \\ & & q_2 = q_G & \text{if } q_1 \in Q, d \neq \epsilon, u \neq \epsilon, \delta(q_1,d) \ !\!\!/ \\ & & q_2 = q_B & \text{if } q_1 \in Q, d \neq \epsilon, u \neq \epsilon, \delta(q_1,d)! \\ & & q_2 = q_1 & \text{if } q_1 \in Q, d = \epsilon, u = \epsilon \\ & & q_2 = q_B & \text{if } q_1 = q_B \\ & & q_2 = q_G & \text{if } q_1 = q_G\} \end{array}$$

$\delta(q,e)!$ denotes that the map δ is defined for the pair (q,e) and $\delta(q,e)\ !\!\!/$that it is not. It can be shown that the runs of H that do not involve q_B and q_G are in 1-1 correspondence with $L(G)$, modulo (ϵ, ϵ) transitions.

To cast the problem in the setting of Section 2 consider a discrete metric, $m : Q \times Q \rightarrow \mathbb{R}$, defined by $m(q_1,q_2) = 0$ if $q_1 = q_2$ and 1 if $q_1 \neq q_2$. The metric induces a map on pairs of subsets of Q by $M(Q_1,Q_2) = \min_{(q_1,q_2) \in Q_1 \times Q_2} m(q_1,q_2)$. Let $d = \{d_1, d_2, \ldots\} \in D^*$ denote a sequence in D and $u = \{u_1, u_2, \ldots\} \in U^*$ denote a sequence in U and define their interleaving as $(d,u) = \{(d_1,u_1),(d_2,u_2),\ldots\} \in (D \times U)^*$. As G is assumed to be deterministic, the above transition structure defines a unique state trajectory $x = \{q_0, q_1, \ldots\} \in X^*$ for every $q_0 \in I$ and every $(d,u) \in (D \times U)^*$. The defining relationship is $(q_i,(d_{i+1},u_{i+1}),q_{i+1}) \in E$. The metric can be used to assign a cost to this run by $J_1(q_0,(d,u)) = -\min_{q \in x} M(\{q\},Q_B)$. The reachability problem can now be thought of as a game between u and d over the cost function J_1. Consider "feedback" maps $\hat{D} : X \rightarrow 2^D$ and $\hat{U} : X \rightarrow 2^U$. The following algorithm provides the safe states and controls.

Algorithm: Safe States and Controls

Step 0: Set $i = 1$ and define $Q'_B = Q_B$, $\hat{D}(q) = \{\epsilon\}$ and $\hat{U}(q) = U$ for all $q \in Q'_B$.

Step i: Define:

$$\text{NewQ}_B = \{q \in Q \setminus Q'_B \quad | \quad \exists d_i \in D, q' \in Q'_B \text{ with } (q,(d_i,\epsilon),q') \in E\}$$

If $\text{NewQ}_B \neq \emptyset$ increment i and for all $q \in \text{NewQ}_B$ define $\hat{U}(q) = U$ and $\hat{D}(q) = \{d_i \in D | \exists q' \in Q'_B \text{ with } (q,(d_i,\epsilon),q') \in E\}$. Redefine $Q'_B = Q'_B \cup \text{NewQ}_B$ and return to step i.

If $\text{NewQ}_B = \emptyset$, then for all $q \in X \setminus Q'_B$ define $\hat{D}(q) = D$ and $\hat{U}(q) = \{u_i \in U | (q,(\epsilon,u_i),q') \in E \Rightarrow q' \notin Q'_B\}$. Define the safe set as $V_1 = Q \setminus Q'_B$. Terminate. ■

Lemma 1 *The algorithm terminates in at most $|Q|$ steps. The system is guaranteed to be safe if and only if $I \subset V_1$ and along the trajectory $u \in \hat{U}(q)$.*

Note that $J_1^*(q_0) = -1$ if $q_0 \in V_1$ and $J_1^*(q_0) = 0$ otherwise. As J_1 can take on only two values, any pair (u^*, d^*) that satisfies $d_i^* \in \hat{D}(q_{i-1})$ and $u_i^* \in \hat{U}(q_{i-1})$ for the corresponding run $x = \{q_0, q_1, \ldots\}$ is a min-max solution.

4 The Steam Boiler

Our analysis of the steam boiler problem is based on the description of [6]. 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 $\dot{p}_1(t)$ and $\dot{p}_2(t)$ respectively. Pump i can either be on ($\dot{p}_i(t) = P_i$) or off ($\dot{p}_i(t) = 0$). There is a delay T_{p_i} between the time pump i is ordered to switch on and the time $\dot{p}_i$ switches to P_i. There is no delay when the pumps are switched off. The objective is to keep the water level between two values M_1 and M_2.

The boiler is modeled by a hybrid dynamical system, $H_B = \{X_B, U_B, Y_B, I_B, f_B, E_B, h_B\}$, with a single discrete state (suppressed to simplify the notation) and two continuous states, the water level w and the rate at which steam escapes, r. We assume that both states are available for measurement, i.e. $Y_B = X_B$ and $h_B(x_B, u_B) = x_B$. The system evolution is influenced by two discrete inputs, $\dot{p}_1$ and $\dot{p}_2$ and one continuous input, the derivative of the steam rate, d. The physical properties of the boiler impose bounds on the states and inputs: $x_B = (w,r) \in X_B = \mathbb{R} \times [0,W]$ and $u_B = (\dot{p}_1, \dot{p}_2, d) \in U_B = \{0, P_1\} \times \{0, P_2\} \times [-U_2, U_1]$, where W, U_1, U_2, P_1 and P_2 are positive constants. Following [6], the dynamics are given by:

$$f_B(x_B, u_B) \quad = \quad \left[\begin{array}{c} \dot{p}_1 + \dot{p}_2 - r \\ d \end{array} \right]$$

The set E_B does not allow any discrete jumps.

Each pump can also be modeled by a hybrid dynamical system, $H_{p_i} = \{X_{p_i}, U_{p_i}, Y_{p_i}, I_{p_i}, f_{p_i}, E_{p_i}, h_{p_i}\}$, with two discrete states $q_i = 0$ and $q_i = P_i$ that reflect if the

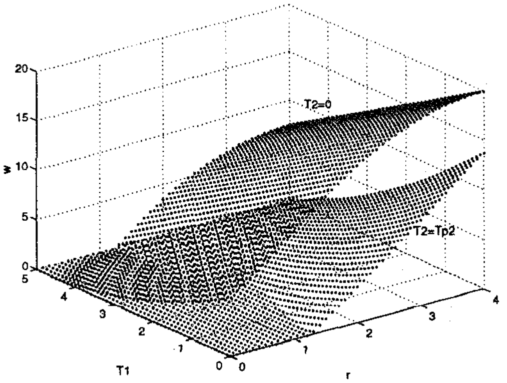

Figure 1: Lower limit on w to avoid draining

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. The evolution of the state is affected by a discrete input, $u_i \in \{0,1\}$ that takes the value 0 if the pump is commanded to switch off and 1 if the pump is commanded to switch on. We assume that the pump state is available for measurement, i.e. $h_{p_i}(x_{p_i}, u_i) = x_{p_i}$. The combined system can be obtained as an interconnection of H_B, H_{p_1} and H_{p_2}. The resulting system will have two discrete and four continuous states. We will use $x = ((q_1, q_2), (w, r, T_1, T_2))$ to denote the overall state.

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 two cost functions $J_1(x^0, u_1, u_2, d) = -\inf_{t \geq 0} w(t)$ and $J_1'(x^0, u_1, u_2, d) = \sup_{t \geq 0} w(t)$. For a given run the safety objective is met if and only if $J_1 \leq -M_1$ and $J_1' \leq M_2$.

For the game with cost J_1 consider the candidate saddle solution $u_i^* \equiv 1$ and $d^*(t) = U_1$ if $r < W$ or $d^*(t) = 0$ if $r = W$. Likewise, for cost J_1', consider the candidate saddle solution $u_i'^* \equiv 0$ and $d'^*(t) = -U_2$ if $r > 0$ or $d'^*(t) = 0$ if $r = 0$. It can be shown that:

Lemma 2 *(u_1^*, u_2^*, d^*) and $(u_1'^*, u_2'^*, d'^*)$ are saddle solutions for the game between (u_1, u_2) and d over J_1 and J_1' respectively.*

The saddle solutions allow us to determine the set of states for which there exists inputs for the pumps such that the water level is guaranteed to remain between the specified limits for any steam rate. 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^0, T_1^0, T_2^0) to the minimum water level required for safety. An example of level sets of $\hat{w}$ for $T_2^0 = 0$ and for $T_2^0 \geq T_{p_2}$ is shown in Figure 1. As expected the higher the value of T_2 the more states are safe (the surface moves down). Safety ($w(t) \geq M_1$) can be maintained as long as the water level is on or above the corresponding surface.

As $J'^*(x^0) = w^0$ any state with $w^0 \leq M_2$ is safe with respect to J'. However, the u_i^* are not the unique minimizers of J', as any controls such that $\dot{w} \leq 0$ whenever $w = w^0$ achieve the same value of J'. As $\dot{w} = q_1 + q_2 - r$, the boundary between safe and unsafe states is such that $w^0 = M_2$ and $r^0 = \hat{r}(T_1^0, T_2^0)$ where:

$$\hat{r}(T_1^0, T_2^0) = \begin{cases} 0 & \text{if } T_1^0 < T_{p_1} \text{ and } T_2^0 < T_{p_2} \\ P_1 & \text{if } T_1^0 \geq T_{p_1} \text{ and } T_2^0 < T_{p_2} \\ P_2 & \text{if } T_1^0 < T_{p_1} \text{ and } T_2^0 \geq T_{p_2} \\ P_1 + P_2 & \text{if } T_1^0 \geq T_{p_1} \text{ and } T_2^0 \geq T_{p_2} \end{cases}$$

Any initial condition such that either $w^0 < M_2$ or $w^0 = M_2$ and $r^0 \geq \hat{r}(T_1^0, T_2^0)$ is safe with respect to J'.

The calculation of the safe set also allows us to classify the controls that can keep the system safe (water level between M_1 and M_2) provided it starts safe (w^0 and r^0 in the ranges discussed above). The class of safe controls is given in a state feedback form.

Lemma 3 *A control law for (u_1, u_2) is safe with respect to M_1 if and only if $u_1 = 1$ whenever $\hat{w}(r, 0, 0) \geq w > \hat{w}(r, T_1, 0)$, $u_2 = 1$ whenever $\hat{w}(r, 0, 0) \geq w > \hat{w}(r, 0, T_2)$ and $u_1 = u_2 = 1$ whenever $w \leq \hat{w}(r, T_1, T_2)$.*

Note that, as $\hat{w}$ is monotone in T_1 and T_2, 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. The controls that are safe with respect to J_1' can similarly be calculated.

5 Flight Vehicle Management Systems

The flight vehicle management system (FVMS) example is based on the dynamic aircraft equations and the design specification of [7]. 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 θ, accessed through the elevators. The outputs we wish to control are the speed V and the flight path angle γ. There are three primary modes of operation. In **Mode 1**, the thrust T is between its specified operating limits ($T_{min} < T < T_{max}$), the control inputs are T and θ, and both V and γ are controlled outputs. In **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 θ, and the only controlled output is V. Finally, in **Mode 3**, the thrust saturates ($T = T_{min} \vee T = T_{max}$); the input is again θ, and the controlled output is γ. Within Modes 2 and 3 there are two submodes depending on whether $T = T_{min}$ (idle thrust) or $T = T_{max}$ (maximum thrust).

Let $x = (V, \gamma) \in \mathbb{R} \times S^1$. The flight path angle dynamics can be modeled by:

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

where m is the mass of the aircraft, g is gravitational acceleration, a_L and a_D are the lift and drag coefficients and c is a small positive constant. Physical considerations impose constraints on the inputs.

$$U = [T_{min}, T_{max}] \times [\theta_{min}, \theta_{max}] \quad (4)$$

Safety regulations for the aircraft dictate that V and γ must remain within specified limits: for ease of presentation we simplify the *safety envelope*, S, of [7] to:

$$S = \{(V,\gamma)|(V_{min} \leq V \leq V_{max}) \cap (\gamma_{min} \leq \gamma \leq \gamma_{max})\}$$

where $V_{min}, V_{max}, \gamma_{min}, \gamma_{max}$ are constant values. In addition, constraints are imposed on the linear and angular accelerations for passenger comfort:

$$|\dot{x}_1(t)| \leq 0.1g, \quad |x_1(t)\dot{x}_2(t)| \leq 0.1g \quad (5)$$

5.1 Optimal Controls and Set of Safe States

Safety is maintained by operating within the largest subset, V_1, of S which can be rendered invariant by control inputs $u \in \mathcal{U}$. Let ∂S denote the boundary of S, ∂V_1 denote the boundary of V_1. We calculate the set V_1 by solving an optimal control problem over a time interval $[t, t_f]$. We define t_f to be the first time at which the state leaves S and let t be free. If t_f exists, we set $t_f = 0$ and consider negative initial times t (without loss of generality, as the dynamics are time invariant). The cost function $J_1(x, t, u(\cdot))$ depends only on the state at the terminal time:

$$J_1(x, t, u(\cdot)) = l(x(0)) \quad (6)$$

with $l(x)$ such that $l(x) > 0$ if $x \in S \backslash \partial S$ [Safe], $l(x) = 0$ if $x \in \partial S$ [Boundary] and $l(x) < 0$ if $x \in \mathbb{R}^n \backslash S$ [Unsafe]. The Hamiltonian is then simply $H_1(x,p,u) = pf(x,u)$, where $p \in T^*\mathbb{R}^2$ is the costate. Let $H_1^*(x,p)$ denote the optimal Hamiltonian, i.e.:

$$H_1^*(x,p) \quad = \quad \max_{u \in U} H_1(x,p,u) = pf(x,u^*)$$

If $J_1^*(x,t)$ is a smooth function of x and t then it satisfies the Hamilton-Jacobi equation:

$$\frac{\partial J_1^*(x,t)}{\partial t} = -H_1^*(x, \frac{\partial J_1^*(x,t)}{\partial x}) \quad (7)$$

with boundary condition $J_1^*(x,0) = l(x)$.

For a given initial time t, the safe set of states is $V_1(t) = \{x \in S | J_1^*(x,t) \geq 0\}$. If we let $t \to -\infty$, the set $V_1(t)$ becomes the "steady state" safe set: $V_1 \equiv V_1(-\infty) = \{x \in S | J_1^*(x,-\infty) \geq 0\}$, with boundary $\partial V_1 = \{x \in S | J_1^*(x,-\infty) = 0\}$. In order to compute the steady state solution $J_1^*(x,-\infty)$ of (7), we assume that no shocks exist, and set the left hand side to zero. Then, $\frac{\partial J_1^*(x,-\infty)}{\partial x}$ is normal to the vector field $f(x,u^*)$.

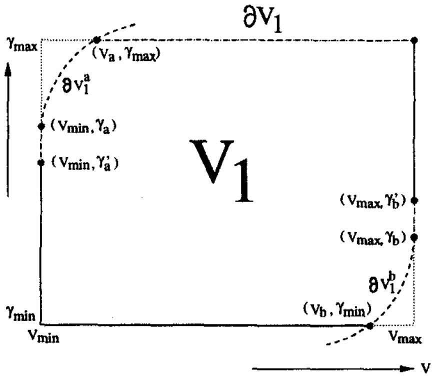

Figure 2: The safe set of states, V_1, and its boundary ∂V_1

We construct ∂V_1 one edge at a time. Define each edge of ∂S separately, by $l_1^1(x) = x_1 - V_{min}$, $l_1^2(x) = -x_2 + \gamma_{max}$, $l_1^3(x) = -x_1 + V_{max}$ and $l_1^4(x) = x_2 - \gamma_{min}$ and let $J_1^i(x,t,u(\cdot)) = l_1^i(x(0))$ be the cost function for edge i, $H_1^i(x,p,u)$ be the corresponding Hamiltonian, and $p_i = \partial l_1^i(x)/\partial x$ be the inward pointing normal to $l_1^i(x) = 0$. Start with $l_1^1(x)$. Define $(V_{min}, \gamma_a) = \{x \in S | l_1^1(x) = 0 \cap H_1^*(x) = 0\}$. γ_a is given by:

$$\gamma_a = \sin^{-1}(\frac{T_{max}}{mg} - \frac{a_D V_{min}^2}{mg}) \quad (8)$$

Integrate the system dynamics backwards from $x(0) = (V_{min}, \gamma_a)$ at $t = 0$ until the solution intersects $\{x \in S | l_1^2(x) = 0\}$. Denote the point of intersection by (V_a, γ_{max}), and the solution between (V_{min}, γ_a) and (V_a, γ_{max}) by ∂V_1^a (Figure 2).

The optimal control u^* is required for this calculation. $p_1 = [1,0]^T$, so along the l_1 boundary, $u_1^* = T_{max}$ but u_2 is indeterminate. Because of the loss of dependency of the optimal Hamiltonian on u_2, the points in $\{x \in S | l_1^1(x) = 0\}$ are *abnormal extremals.* At the abnormal extremal (V_{min}, γ_a), any $u_2 \in [\theta_{min}, \theta_{max}]$ may be used. However, as we integrate, we instantaneously leave the abnormal extremal regardless of the choice of u_2. From then on 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, $u_2^* = \theta_{min}$. In this example, the abnormal extremal was not complicated enough to cause difficulties in the construction; the general situation is considered in [8].

The calculation can be repeated for the remaining three boundaries. Only $\{x \in S | l_1^3(x) = 0\}$ contains a point at which $H_1^*(x)$ vanishes. We denote this point by (V_{max}, γ_b) where:

$$\gamma_b = \sin^{-1}(\frac{T_{min}}{mg} - \frac{a_D V_{max}^2}{mg}) \quad (9)$$

and calculate ∂V_1^b and V_b similarly.

Lemma 4 *The safe set is enclosed by:*

$$\begin{aligned}\partial V_1 = \{(V,\gamma) |\ & (V = V_{min}) \wedge (\gamma_{min} \leq \gamma \leq \gamma_a) & \vee \\ & \partial V_1^a & \vee \\ & (\gamma = \gamma_{max}) \wedge (V_a \leq V \leq V_{max}) & \vee \\ & (V = V_{max}) \wedge (\gamma_b \leq \gamma \leq \gamma_{max}) & \vee \\ & \partial V_1^b & \vee \\ & (\gamma = \gamma_{min}) \wedge (V_{min} \leq V \leq V_b)\} & \end{aligned}$$

The safe controls U_1 can now be characterized as a set-valued feedback map $U_1 : S \rightarrow 2^U$:

Lemma 5 $U_1(x) = U \cap \hat{U}_1(x)$, *with:*

$$\begin{aligned}\hat{U}_1(x) = \{\ & \emptyset,\ \mathit{if}\ x \in S \backslash V_1 \\ & T \geq T_a(\gamma),\ \mathit{if}\ (V = V_{min}) \wedge (\gamma \leq \gamma_a) \\ & \theta = \theta_{min} \wedge T = T_{max},\ \mathit{if}\ x \in \partial V_1^a \\ & \theta \leq \theta_c(V),\ \mathit{if}\ (\gamma = \gamma_{max}) \wedge (V_a \leq V) \\ & T \leq T_b(\gamma),\ \mathit{if}\ (V = V_{max}) \wedge (\gamma_b \leq \gamma) \\ & \theta = \theta_{max} \wedge T = T_{min},\ \mathit{if}\ x \in \partial V_1^b \\ & \theta \geq \theta_d(V),\ \mathit{if}\ (\gamma = \gamma_{min}) \wedge (V \leq V_b) \\ & U,\ \mathit{else}\}\end{aligned}$$

where $T_a(\gamma) = a_D V_{min}^2 + mg \sin\gamma$, $T_b(\gamma) = a_D V_{max}^2 + mg \sin\gamma$, $\theta_c(V) = \frac{m}{a_L V c}\left(\frac{g \cos\gamma_{max}}{V} - \frac{a_L V(1 - c\gamma_{max})}{m}\right)$ *and* $\theta_d(V) = \frac{m}{a_L V c}\left(\frac{g \cos\gamma_{min}}{V} - \frac{a_L V(1 - c\gamma_{min})}{m}\right)$.

In Figure 2, the portions of ∂V_1 for which all control inputs are safe ($U_1(x) = U(x)$) are indicated with solid lines; those for which only a subset are safe ($U_1(x) \subset U(x)$) are indicated with dashed lines. The map defines the least restrictive safe control scheme and determines the mode switching logic. On ∂V_1^a and ∂V_1^b, the system must be in **Mode 2** or **Mode 3**. Anywhere else in V_1, any of the three modes is valid as long as the input constraints of Lemma 5 are satisfied. In the regions $S \backslash V_1$ (the upper left and lower right corners of S), no control inputs are safe ($U_1(x) = \emptyset$).

5.2 Passenger Comfort Constraints

Cost functions involving the linear and angular accelerations can be used to encode passenger comfort:

$$J_2(x, u(\cdot)) = \max_{t \geq 0} |\dot{x}_1(t)|,\ J_2'(x, u(\cdot)) = \max_{t \geq 0} |x_1(t)\dot{x}_2(t)|$$

The requirement that the linear and angular acceleration remain within the limits determined for comfortable travel are encoded by thresholds $J_2(x, u(\cdot)) \leq 0.1g$ and $J_2'(x, u(\cdot)) \leq 0.1g$. Within the class of safe controls, a control scheme which meets the passenger comfort (efficiency) objective can be constructed. The sets of comfortable states and controls can be easily calculated by substituting the bounds on the accelerations into the system dynamics, to get:

$$|T - a_D V^2 - mg \sin\gamma| \leq 0.1mg$$
$$\left|\theta + \frac{1 - c\gamma}{c} - \frac{mg \cos\gamma}{a_L V^2 c}\right| \leq \frac{0.1mg}{a_L V^2 c}$$

6 Conclusions

We presented a methodology for synthesizing controllers for hybrid systems to meet multiple control objectives. We restricted our attention to two objectives, safety and efficiency; the methodology easily extends to an arbitrary number. The notions of "maximal safe set" and "least restrictive safe controller" are central to our formulation. They 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 a hierarchical context, as they can provide sufficient conditions for a supervisor that switches between controllers to be safe [4].

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 techniques used in the last example (FVMS) seem to be the most promising in this respect. We are currently working on formalizing these techniques in the context of semi-permeable surface calculation in pursuit evasion games. Semi-permeable surfaces form the boundary of the maximal safe set and define regions where there are limitations on the allowable controls.

References

[1] John Lygeros, Claire Tomlin, and Shankar Sastry, "Mulit-objective hybrid controller synthesis", Tech. Rep. UCB/ERL M96/59, Electronic Research Laboratory, University of California Berkeley, 1997.

[2] John Lygeros, *Hierarchical Hybrid Control of Large Scale Systems*, PhD thesis, Electrical Engineering, University of California, Berkeley, 1996.

[3] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A case study in multi-agent hybrid systems", Tech. Rep., UCB/ERL M97/33, Electronics Research Laboratory, University of California, Berkeley, 1997.

[4] John Lygeros, Datta N. Godbole, and Shankar Sastry, "A verified hybrid controller for automated vehicles", in *IEEE Control and Decision Conference*, 1996, pp. 2289–2294.

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

[6] T. A. Henzinger and H. Wong-Toi, "Using HYTECH to synthesize control parameters for a steam boiler", in *Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control*, number 1165 in LNCS, pp. 265–282. Springer Verlag, 1996.

[7] Charles S. Hynes and Lance Sherry, "Synthesis from design requirements of a hybrid system for transport aircraft longitudinal control", preprint, NASA Ames Research Center, 1996.

[8] Claire Tomlin, Shankar Sastry, and Richard Montgomery, "Computing safe sets using the Hamilton-Jacobi equation", (preprint), 1997.