From: AAAI Technical Report SS-99-05. Compilation copyright © 1999, AAAI (www.aaai.org). All rights reserved.

A Case Study of Hybrid Controller Synthesis of a Heating System

Andrea Balluchi[§] Luca Benvenuti[§] Tiziano Villa[§]
Howard Wong-Toi[†] Alberto L. Sangiovanni-Vincentelli[§‡]

Abstract

A controller for a non-deterministic hybrid plant must ensure that the closed-loop system meets some requirement, regardless of what the plant does. When the plant is viewed as an adversary, controller synthesis becomes the task of solving a two-person game to find the system configurations from which the controller wins. For hybrid systems, the moves of each player can be either discrete or continuous. Thus winning strategies can involve a non-trivial mix of continuous and discrete actions, and are in general not easy to find. Fortunately, there is a systematic procedure to find all winning strategies in the case of safety properties (Tomlin, Lygeros, & Sastry 1998; Lygeros, Tomlin, & Sastry 1999).

To assess the applicability of the procedure, we study a hybrid model of a heating system. The system incorporates both discrete controls and disturbances, and continuous controls and disturbances. For this system, we detail how to compute the set of winning configurations, using a combination of case analysis in the discrete domain and solving min-max problems in the continuous domain. The steps of the synthesis procedure for our case study have been implemented in MATLAB, enabling us to experiment with different parameter settings. We also discuss preliminary lessons learned from this case study, and suggest areas for future research that will enable the synthesis procedure to be more applicable in practice.

Introduction

The task of synthesizing a controller can be viewed as solving a two-person game between a plant and its controller. In this paper, we restrict attention to *safety* games, where the closed-loop system meets its specification if the system configuration remains within some predetermined set of safe configurations. Each player has at its disposal two kinds of moves: discrete and continuous. It continuously monitors the full configuration of the system. At every point in time, it chooses either to make a discrete action, or to make no discrete action, but instead allow time to pass subject to its continuous action. If either player chooses a discrete action, then the discrete action takes place instantaneously. If both players choose a discrete action, then both actions take place simultaneously. If both players agree to let time pass, the system evolves according to its continuous dynamics, which are dependent on the continuous moves of both players, up until one player next chooses a discrete action. The controller wins the game if it can guarantee that the system configuration is always safe, regardless of what the adversarial plant does. Here, we study the systematic procedure for controller synthesis of hybrid systems presented in (Tomlin, Lygeros, & Sastry 1998; Lygeros, Tomlin, & Sastry 1998; 1999). It iteratively determines the configurations from which the controller will lose the game within a finite number of discrete actions. If the procedure terminates, the remaining configurations are precisely those from which the controller has a winning strategy. It is not however guaranteed to terminate, and the individual steps within the procedure may be impractical or even impossible to perform.

Here, we study the hybrid control of a mixed discrete-continuous heating system in order to assess the applicability of the synthesis techniques outlined above. We show how to apply the procedure for the heater example using min-max problems. The reported experiments were performed by implementing the synthesis procedure for our system in MATLAB. The maximal safe sets obtained vary depending on the settings of the parameters of the system. Finally we extract the maximal control strategy from the maximal safe set and analyze sample trajectories to highlight the operation of the controller. Our main conclusion is that the calculations involved in finding the sets of continuous-uncontrollable predecessors are prohibitive. One must reason about non-trivial shaped sets, and solve optimal control problems over them, taking into consideration that continuous trajectories must steer clear of sets where the opponent can escape to other parts of the state space via discrete jumps. In dimensions even as low as three, these calculations can become infeasible.

Background. In the discrete domain, Church's classical synthesis problem was first solved by reduction to a zero-sum, two-person game over infinite strings (Büchi & Landweber 1969). Numerous researchers have studied solutions to the synthesis problem via translations to tree automata; a tree is accepted iff it corresponds to a winning strategy, e.g., (Rabin 1972; Pnueli & Rosner 1989). For safety games, the game admits a particularly simple solution that consists of a fixpoint algorithm that successively eliminates states that lead to losing states within one discrete move, until the controller can always keep the state

[§] PARADES, Via di S.Pantaleo, 66, 00186 Roma, Italy. Email: {balluchi, lucab, villa, alberto}@parades.rm.cnr.it.

[†] Cadence Berkeley Labs, 2001 Addison St., Third Floor, Berkeley, CA 94704, USA. Email: howard@cadence.com.

[‡] Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720, USA. Email: alberto@eecs.berkeley.edu.

within those that remain (Ramadge & Wonham 1989; Maler, Pnueli, & Sifakis 1995).

In the continuous domain, there has been study of *differential games* between players who choose continuous values that affect a system's continuous evolution (Isaacs 1967). Casting controller synthesis for hybrid systems as differential games appears in (Nerode & Yakhnis 1992; Lygeros, Godbole, & Sastry 1996).

The discrete synthesis procedure has been extended to automata modeling timed systems (Maler, Pnueli, & Sifakis 1995), and restricted forms of hybrid systems (Heymann, Lin, & Meyer 1997; Wong-Toi 1997), where the discrete moves are used to define target sets for the different opponents. The continuous dynamics of these works is simple, lacking even continuous controller input. The synthesis procedure assessed here is more general, and uses differential games to handle the continuous activity between potential discrete moves (Tomlin, Lygeros, & Sastry 1998).

Description of the Model

Our heating system has discrete and continuous components in its state, its control input, and its disturbance. The control objective is to maintain the temperature T_a of the air in a room within the range $[T_a^{min}, T_a^{max}]$, whatever the disturbances happen to be. The controller has at its disposal a boiler and a stove. It operates under full state feedback. The boiler can be viewed as a heating element that admits continuous settings: it receives a continuous input control variable $u_b \in [0, U_b]$ and outputs this power value instantaneously. The stove has only discrete settings. It is switched on or off by a two-valued input control variable $u_s \in \{0, 1\}$. When switched on, the stove delivers heat $w_s = w_s^{max}$; when switched off, it delivers heat $w_s = 0$.

The room is subject to non-deterministic disturbances that affect the temperature. First, the room contains electrical appliances whose operation generates heat as a side effect—modeled by a continuous input disturbance variable $d_e \in [0, D_e]$. Second, the room has a door that may be either closed or open. Its state is set by a two-valued input disturbance variable $d_d \in \{0, 1\}$. When the door is opened the air temperature of the room suddenly decreases. Its difference from the external temperature T_e is multiplied by a ratio $r < 1$, i.e., T_a is updated to $T_e + r(T_a - T_e)$. For physical reasons, we rule out the possibility of the door opening and closing infinitely often in zero time by assuming that at least Δ time passes between changes in the status of the door.

The continuous dynamics of the system are captured by two first-order differential equations whose unknowns are the room air temperature $T_a(t)$, and the door timer $t_d(t)$, with $\dot{t}_d(t) = 1$. For convenience, we translate the temperature variable to $T_{ae} = T_a - T_e$. We derive the following equation for T_{ae}:

$$\dot{T}_{ae}(t) = -\frac{1}{c_a}(\mu_{ae} + \mu_d(d_d))T_{ae}(t) + \frac{1}{c_a}(u_b(t) + d_e(t) + w_s(u_s)) \quad (1)$$

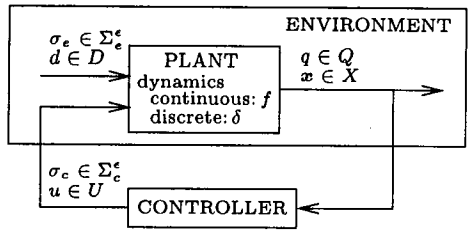


Figure 1: Generic control diagram

where $\mu_d(d_d) = \mu_{do}$ if $d_d = 1$ and $\mu_d(d_d) = \mu_{dc}$ if $d_d = 0$, and $w_s(u_s) = w_s^{max}$ if $u_s = 1$ and $w_s(u_s) = 0$ if $u_s = 0$, for the thermic conductance parameters μ_{ae}, (resp. μ_{dc}, μ_{do}) for the walls between the room and the environment (resp. the closed door, the open door), and c_a the air thermic capacitance.

Hybrid Automata

Syntax

A generic control diagram depicting the interaction between the controller and the environment appears in Figure 1. We model our system as a hybrid automaton. Intuitively, the hybrid automaton models the game board. This modeling formalism merges the game features (explicitly-defined independent moves) of (Asarin *et al.* 1998) into the hybrid automata model (input structure and hybrid dynamics) found in (Tomlin, Lygeros, & Sastry 1998; Lygeros, Tomlin, & Sastry 1998).

A *hybrid automaton* is a tuple $H = ((Q, X), (U, \Sigma_c), (M_c^{cts}, M_c^{disc}), (D, \Sigma_e), (M_e^{cts}, M_e^{disc}), (f, \delta))$. Elements of $\mathcal{C} = Q \times X$ are called *configurations*, where Q is the finite set of *modes* and $X = \mathbb{R}^n$ is the set of (continuous) *states*. The controller input comes from the domain $U \times \Sigma_c^\epsilon$, where $U \subseteq \mathbb{R}^m$ is a set of *continuous control values*, Σ_c is a finite set of *discrete control events* and $\Sigma_c^\epsilon = \Sigma_c \cup \{\epsilon\}$ is the set of *discrete control moves*, with the special ϵ move being the *silent* move. The *discrete (resp. continuous) controller move function* $M_c^{disc} : \mathcal{C} \to 2^{\Sigma_c^\epsilon} \setminus \{\}$ (resp. $M_c^{cts} : \mathcal{C} \to 2^U \setminus \{\}$) defines a subset of allowable discrete (resp. continuous) input values for every configuration. The environment input (or disturbance) comes from the domain $D \times \Sigma_e^\epsilon$, where $D \subseteq \mathbb{R}^p$ is a set of *continuous environment (or disturbance) values*, Σ_e is a finite set of *discrete environment events* and $\Sigma_e^\epsilon = \Sigma_e \cup \{\epsilon\}$ is the set of *discrete environment moves*. The *discrete* and *continuous environment move functions* $M_e^{disc} : \mathcal{C} \to 2^{\Sigma_e^\epsilon} \setminus \{\}$ and $M_e^{cts} : \mathcal{C} \to 2^D \setminus \{\}$ are analogous to the controller move functions. The continuous dynamics are modeled by the mode-dependent function $f : \mathcal{C} \times U \times D \to \mathbb{R}^n$; the discrete dynamics, modeled by the *transition function* $\delta : \mathcal{C} \times \Sigma_c^\epsilon \times \Sigma_e^\epsilon \to 2^{\mathcal{C}} \setminus \{\}$, are subject to the restriction that for all $(q, x) \in \mathcal{C}$, $\delta(q, x, \epsilon, \epsilon) = \{(q, x)\}$. As usual we assume existence and uniqueness of solutions for f etc.

Intuitive semantics

Both players have perfect information, and make their moves simultaneously. Thus the controller is a full-state feedback controller. At the configuration (q, x), the controller chooses a pair $(\sigma_c, u) \in M_c^{disc}(q, x) \times M_c^{cts}(q, x)$, and the environment chooses a pair $(\sigma_e, d) \in M_e^{disc}(q, x) \times M_e^{cts}(q, x)$. If either of the players chooses a non-silent discrete move, then a non-trivial discrete step takes place, with label (σ_c, σ_e), to any configuration in $\delta(q, x, \sigma_c, \sigma_e)$. As long as both players choose ϵ as their discrete move, then a continuous step takes place as time progresses. In this case, the discrete mode remains fixed, and the continuous variables evolve according to the continuous control u chosen by the controller, the continuous disturbance d chosen by the environment, and the continuous dynamics specified by the function f for the discrete mode. One may think of the interaction between the players as a continuous game with occasional discrete interruptions.

A *safety property* asserts that nothing bad happens along system trajectories. It can be characterized by the set *Good* of configurations that do not violate the property. The hybrid automaton with initial configurations $\mathcal{C}_0 \subseteq \mathcal{C}$ satisfies the safety property *Good* if all its trajectories that start in $\mathcal{C}_0$ remain within *Good*.

Heating system

To avoid nonZeno controllers (which enforce safety properties only by causing time to stop), we introduce a timer t_c to enforce that no discrete control move can occur until at least Δ time has passed in a mode. However, to reduce complexity we replace the two timers t_c and t_d with only one, i.e., t_c, weakening the expressiveness of the model.

The hybrid automaton for the heating system is depicted in Figure 2, and sketched below. For modes, we have $Q = \{q_1 = (off, closed), q_2 = (on, closed), q_3 = (on, open), q_4 = (off, open)\}$. The first component of each tuple refers to the status of the stove, and the second to the door. We set $X = \{(t_c, T_{ae}) \mid (t_c, T_{ae}) \in \mathbb{R}^2\}$. The continuous controller input set is $U = \{u_b \mid u_b \in [0, U_b]\}$, and its discrete input events are $\Sigma_c = \{stove_on, stove_off\}$. The event *stove_off* appears in the discrete controller move function whenever the mode is $(on, open)$ or $(on, closed)$, and $t_c \geq 0$. In addition, *stove_on* is allowed whenever the mode is $(off, open)$ or $(off, closed)$, and $t_c \geq 0$. For all (q, x), $M_c^{cts}(q, x) = U$. For the environment, we have continuous disturbance input $D = \{d_e \mid d_e \in [0, D_e]\}$, and discrete input $\Sigma_e = \{door_close, door_open\}$. We have $\epsilon \in M_c^{disc}(q, x)$ and $\epsilon \in M_e^{disc}(q, x)$ for all (q, x), i.e., neither the controller nor the environment is ever forced to make a discrete action.

The first derivative for T_{ae} associated with each mode q_i is determined according to Equation 1 (e.g., for $q_2 = (on, closed)$, $\dot{T}_{ae}(t)$ is $-\frac{1}{c_a}(\mu_{ae} + \mu_{dc})T_{ae}(t) + \frac{1}{c_a}(u_b(t) + d_e(t) + w_s^{max}))$. In all modes, the dynamics of the door timer are specified as $\dot{t}_c(t) = 1$.

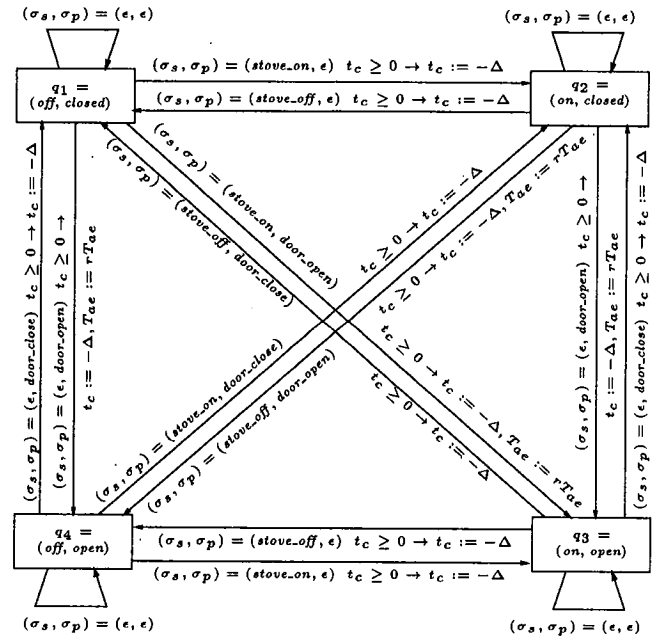


Figure 2: Hybrid model of the room.

Synthesis of Hybrid Controllers

We review the synthesis methodology introduced in (Tomlin, Lygeros, & Sastry 1998).

Controllers

At configuration (q, x), a controller decides whether to (1) take some discrete control action in Σ_c, or to (2) let time pass under a continuous input u. Formally, a controller for a hybrid automaton is a pair (T^{disc}, T^{cts}), where $T^{disc} : \mathcal{C} \rightarrow 2^{\Sigma_c^e} \setminus \{\}$ and $T^{cts} : \mathcal{C} \rightarrow 2^U \setminus \{\}$ model the values allowed by the controller. The controller can only offer values permitted by the move functions, and hence for all $(q, x) \in \mathcal{C}$, we require $T^{disc}(q, x) \subseteq M_c^{disc}(q, x)$ and $T^{cts}(q, x) \subseteq M_c^{cts}(q, x)$. The coupling of the hybrid automaton H with the controller (T^{cts}, T^{disc}) is the hybrid automaton by replacing the controller move functions M_c^{disc} with T^{disc} and M_c^{cts} with T^{cts}. A set of configurations is a *safe* set if all its configurations satisfy the specification, and from all its configurations there exists a controller strategy to remain in the set.

Synthesis procedure

The procedure to synthesize the maximal controller first computes the maximal safe set. This maximal set is obtained by first overapproximating it with all the safe configurations. Then one eliminates all configurations from which the environment can drive the system into an unsafe configuration via either one discrete jump, or one continuous flow. The controller must avoid these configurations, since from them the environment can win within one "step". By iteration, one eliminates the

configurations from which the environment wins within i steps. If the procedure terminates, we have determined the maximal safe set.

For discrete steps, we define the *discrete uncontrollable predecessors* operator $Pre_e : 2^{\mathcal{C}} \rightarrow 2^{\mathcal{C}}$ that characterizes the configurations where the environment wins (enters the complement of its argument set) within one discrete step (Maler, Pnueli, & Sifakis 1995; Tomlin, Lygeros, & Sastry 1998). It is defined by $Pre_e(K) = \{(q,x) \in \mathcal{C} : \forall \sigma_c \in M_c^{disc}(q,x).\exists \sigma_e \in M_e^{disc}(q,x).(\sigma_c,\sigma_e) \neq (\epsilon,\epsilon) \wedge \delta((q,x),(\sigma_c,\sigma_e)) \not\subseteq K\}$.

For continuous steps, we define the operator *Unavoid_Pre*, which captures the configurations for which no matter what continuous input function u the controller chooses there is a continuous disturbance function such that the resulting continuous flow reaches a bad state, avoiding along the way all configurations where the controller could "escape" by causing a jump to a good state. The escaping configurations (Wong-Toi 1997; Lygeros, Tomlin, & Sastry 1998) are characterized by the *discrete controllable predecessors* operator $Pre_c : 2^{\mathcal{C}} \rightarrow 2^{\mathcal{C}}$ defined by $Pre_c(K) = \{(q,x) \in \mathcal{C} : \exists \sigma_c \in M_c^{disc}(q,x).\forall \sigma_e \in M_e^{disc}(q,x).(\sigma_c,\sigma_e) \neq (\epsilon,\epsilon) \wedge \delta((q,x),(\sigma_c,\sigma_e)) \subseteq K\}$. The set $Pre_c(K)$ is the set of configurations where the controller can guarantee that the system enters K in one discrete step, regardless of the environment's action.

The *Unavoid_Pre* operator takes two arguments. The first is the environment's target set, and the second the set it must avoid. The set $Unavoid_Pre(B,E)$ contains all configurations (q,x) such that for all legal controller input functions $u(\cdot)$, there exists a disturbance function $d(\cdot)$ for which the trajectory $x : \mathbb{R}_{\geq 0} \rightarrow X$ reaches the set B at some time $\bar{t}$ with $(q,x(t)) \in Inv \cap \overline{E}$ for all $t \in [0,\bar{t})$. The set Inv denotes the set of configurations $\{(q,x) \mid \epsilon \in M_c^{disc}(q,x) \text{ and } \epsilon \in M_e^{disc}(q,x)\}$ in which both players may choose not to play a discrete move, but instead wait for time to pass.

Given an overapproximation W^i of the winning states, the procedure prunes away the losing configurations $Pre_e(W^i)$ (for discrete steps) and $(Unavoid_Pre(Pre_e(W^i) \cup \overline{W^i}, Pre_c(W^i))$ (for the continuous steps) (Tomlin, Lygeros, & Sastry 1998). It is not guaranteed to stop in a finite number of steps.

Extracting the maximal control strategy from the maximal safe set W amounts to determining for every configuration in W, which control choices (discrete and continuous) will keep the system in W.

Synthesis for the heating system

We show how to compute the Pre_c, Pre_e, and *Unavoid_Pre* operators at each iteration for the hybrid automaton shown in Fig. 2.

The ensuing computations are largely independent of the specific parameters chosen. However, for illustrative purposes, we use particular parameter values in order to demonstrate the procedure in practice.

The temperature must be maintained within values $T_{ae}^{min} = 18$ and $T_{ae}^{max} = 20$. The reset ratio is $r = 0.95$. We normalize $c_a = 1$. The continuous input domains are $U = [0, U_b = 0.5]$ for control and $D = [0, D_e = 0.01]$ for disturbance. The maximum power of the stove is $w_s^{max} = 0.2$. The conductances are such that $\mu_{ae} + \mu_{dc} = 0.001$ and $\mu_{ae} + \mu_{do} = 0.002$.

The iterations of the synthesis procedure for these parameters appear in Figure 3. When the controller input is restricted to $[0, 0.2]$ instead of $[0, 0.5]$, there is no valid controller.

Discrete predecessor operators

We first define three useful auxiliary operators. Let $W \subseteq \mathcal{C}$ be a set of configurations, and $q \in Q$ be a mode. Let $W|_q = \{x \in X \mid (q,x) \in W\}$ denote the projection of elements of W onto the continuous state only. Let $W|_q^{-\Delta} = \{(t_c, T_{ae}) \mid (-\Delta, T_{ae}) \in W|_q\}$ denote the set of points for which resetting t_c to $-\Delta$ results in a point in $W|_q$. Let $W|_q^{-\Delta r} = \{(t_c, T_{ae}) \mid (-\Delta, r\ T_{ae}) \in W|_q\}$ denote the set of points for which resetting t_c to $-\Delta$ and multiplying T_{ae} by r results in a point in $W|_q$. We also define the set $\mathcal{T}_{\geq 0} = \{(t_c, T_{ae}) \mid t_c \in [0, \infty)\}$.

We show how to compute $Pre_c(W)$ mode by mode. Consider q_1. The controller has two choices of discrete actions (*stove_on* and ϵ) to force the system into the set W. Consider first the *stove_on* action. It is only enabled when $t_c \geq 0$. If the environment simultaneously chooses $\sigma_e = \epsilon$, there will be a jump to q_2. Since the timer is reset to $t_c = -\Delta$ and the temperature unchanged, the states (t_c, T_{ae}) land in $W|_{q_2}$ iff $(-\Delta, T_{ae}) \in W|_{q_2}$ and $t_c \geq 0$ iff $(t_c, T_{ae}) \in W|_{q_2}^{-\Delta} \cap \mathcal{T}_{\geq 0}$. If the environment chooses $\sigma_e = door_open$, there is a jump to q_3. Since the timer is reset to $t_c = -\Delta$ and the temperature T_{ae} is reset to rT_{ae}, the states (t_c, T_{ae}) land in $W|_{q_3}$ iff $(-\Delta, r\ T_{ae}) \in W|_{q_3}$ and $t_c \geq 0$ iff $(t_c, T_{ae}) \in W|_{q_3}^{-\Delta r} \cap \mathcal{T}_{\geq 0}$. Thus the discrete action *stove_on* witnesses the inclusion of (q_1, x) in $Pre_c(W)$ iff $x = (t_c, T_{ae})$ meets both the conditions above for the choice of environment action iff $(t_c, T_{ae}) \in W|_{q_2}^{-\Delta} \cap W|_{q_3}^{-\Delta r} \cap \mathcal{T}_{\geq 0}$.

Consider next the case of the ϵ action. The action is always enabled in the controller. Since ϵ is always enabled in the environment, the condition $(\sigma_c, \sigma_e) \neq (\epsilon, \epsilon) \wedge \delta((q,x),(\sigma_c,\sigma_e)) \subseteq W$ inside the quantifications in the definition of Pre_c is FALSE because of the first conjunct. Therefore $\sigma_c = \epsilon$ cannot be an existential witness for any (q,x).

Thus we conclude that $Pre_c(W)|_{q_1} = W|_{q_2}^{-\Delta} \cap W|_{q_3}^{-\Delta r} \cap \mathcal{T}_{\geq 0}$. Analogous reasoning yields similar formulae for the other modes.

Computation of the discrete uncontrollable predecessors is similar.

Continuous uncontrollable predecessors

The set $Unavoid_Pre(Pre_e(W^i) \cup \overline{W^i}, Pre_c(W^i))$ is also computed mode by mode, i.e., by computing for each mode q, the set $Unavoid_Pre(B,E)|_q$ for $B = Pre_e(W^i) \cup \overline{W^i}$ and $E = Pre_c(W^i)$. We restrict the

analysis to $X = [-\Delta, \infty) \times \mathbb{R}$, since t_c is always reset to $-\Delta$ and $\dot{t}_c = 1 > 0$. For a fixed q, the continuous–time dynamics are

$$\dot{t}_c = 1 \quad (2)$$

$$\dot{T}_{ae} = aT_{ae} + b(u_b + d_e) + b_0 \quad (3)$$

Since the objective of the game is to find the configurations not already in $\overline{W^i}|_q$ that can be steered to $B|_q$ without passing through $E|_q$, we restrict attention to the set $R|_q = W^i|_q \setminus (E|_q \cup B|_q)$. The boundary $\partial R|_q$ of $R|_q$ is made by arcs of $\partial E|_q$ and arcs of $\partial B|_q$, boundaries of $E|_q$ and $B|_q$ respectively, and segments that lie on $t_c = -\Delta$. Since $\dot{t}_c = 1$, trajectories starting inside $R|_q$ cannot exit $R|_q$ through the boundary of $R|_q$ that lies on $t_c = -\Delta$. That is, for any $\hat{x} \in R|_q$, under any $u_b \in \mathcal{U}_b$ and $d_e \in \mathcal{D}_e$, the trajectory from $\hat{x}$ either remains in $R|_q$ for all $t > 0$ or intersects, at some time $t = \bar{t}$, either $\partial E|_q \cap \partial R|_q$ or $\partial B|_q \cap \partial R|_q$.

The set *Unavoid_Pre*$(B, E)|_q$ corresponds to the playable set for the disturbance d_e in a two–player differential game defined as follows (see (Isaacs 1967)). Given an initial state $x_0 \in R|_q$, the disturbance d_e wants to steer x_0 to $\partial B|_q \cap \partial R|_q$, while the control u_b opposes it (u_b wants to steer x_0 to $\partial E|_q \cap \partial R|_q$). The playable set for d_e in the two–player differential game is the subset of $R|_q$ from which the player d_e can guarantee to drive the initial state to the target set $\partial B|_q \cap \partial R|_q$, regardless of the adversarial control actions of u_b.

One can show that a family of curves sufficient for the description of the boundary of the playable set for d_e can be derived from the solution of a min–max problem (see (Vincent & Grantham 1997)). Introduce the adjoint variables λ_1, λ_2 and the Hamiltonian associated to the dynamics (2),(3)

$$H(t_c, T_{ae}, \lambda_1, \lambda_2, d_e, u_b) = \lambda_1 \dot{t}_c + \lambda_2 \dot{T}_{ae} = \lambda_1 + \lambda_2(aT_{ae} + b(u_b + d_e) + b_0) \ . \quad (4)$$

If $d_e^\star(t), u_b^\star(t)$ generate a trajectory $[t_c^\star(t), T_{ae}^\star(t)]^T$ on the boundary of the playable set, then there exists a nonzero continuous trajectory $[\lambda_1(t), \lambda_2(t)]^T$, satisfying

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial t_c} = 0 \text{ and } \dot{\lambda}_2 = -\frac{\partial H}{\partial T_{ae}} = -a\lambda_2 \ , \quad (5)$$

such that $[\lambda_1(t), \lambda_2(t)]^T$ is an outward normal to the boundary of the playable set and

$$\min_{d_e \in \mathcal{D}_e} \max_{u_b \in \mathcal{U}_b} H(t_c^\star, T_{ae}^\star, \lambda_1, \lambda_2, d_e, u_b) = H(t_c^\star, T_{ae}^\star, \lambda_1, \lambda_2, d_e^\star, u_b^\star) = 0 \ . \quad (6)$$

By (4), the signals $d_e^\star(t)$, $u_b^\star(t)$ that satisfy (6) are

$$d_e^\star(t) = \begin{cases} 0, & \text{if } b\lambda_2(t) > 0 \\ D_e, & \text{if } b\lambda_2(t) < 0 \end{cases} \quad (7)$$

$$u_b^\star(t) = \begin{cases} U_b, & \text{if } b\lambda_2(t) > 0 \\ 0, & \text{if } b\lambda_2(t) < 0 \end{cases} \ . \quad (8)$$

Since by (5) $\lambda_2(t) = e^{-at}\lambda_{20}$ where $\lambda_{20} = \lambda_2(0)$, if $\lambda_{20} \neq 0$, then d_e and u_b are constant along the boundary of the playable set, because $b\lambda_2(t)$ never changes in sign. Moreover, by (5), λ_1 is also constant.

If $\lambda_{20} = 0$ then $\lambda_2(t) = 0$ for all t and a singular control may occur. However, singular controls cannot take place; in fact, by (6),(4) $\lambda_1(t)$ has to be zero if $\lambda_2(t) = 0$, which is against the request of $[\lambda_1(t), \lambda_2(t)]^T$ being nonzero.

Then, a trajectory $[t_c(t)\ T_{ae}(t)]^T =$

$$\begin{bmatrix} t_c(0) + t \\ e^{at}T_{ae}(0) + (1 - e^{at})\left[-a^{-1}b\,(u_b + d_e) - a^{-1}b_0\right] \end{bmatrix} \quad (9)$$

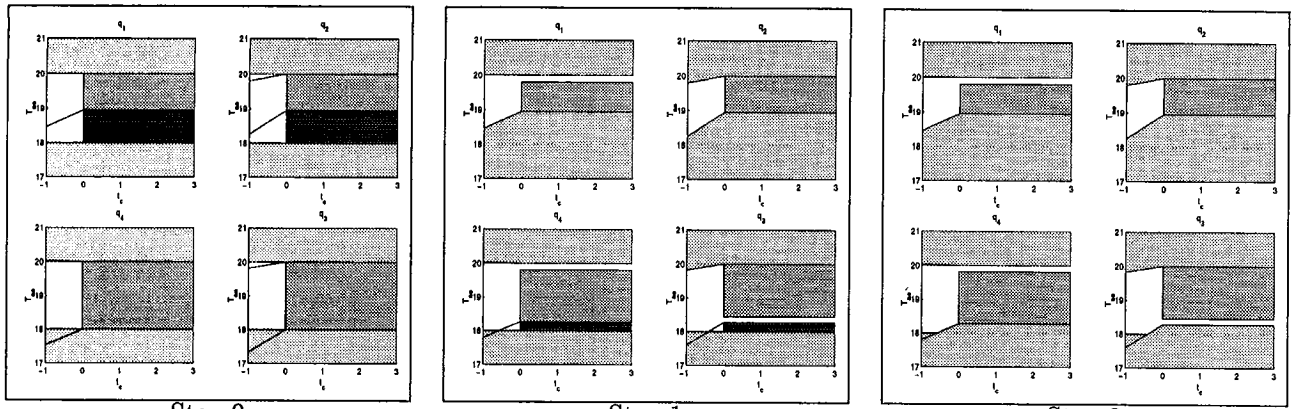
solution to (2),(3) with constant inputs $u_b = u_b^\star$ and $d_e = d_e^\star$ chosen according to (7),(8) satisfies the min–max necessary condition to belong to the boundary of the playable set.

It is clear that along a trajectory of type (9), T_{ae} is monotonic with respect to t_c. Hence, if an arc of trajectory (9) lies on the boundary of the playable set then the playable set is either below or above it, and an outward normal $[\lambda_1(t), \lambda_2(t)]^T$ of the playable set has either $\lambda_2 > 0$ in the former case (the arc is an upper boundary) or $\lambda_2 < 0$ in the latter case. According to (8), if such trajectory defines an upper boundary, then (since $\lambda_2 > 0$) necessarily $d_e = 0$ and $u_b = U_b$; else if it defines a lower boundary, then (since $\lambda_2 < 0$) $d_e = D_e$ and $u_b = 0$. So, a family of curves $\varphi^{upper}_{(\hat{t}_c, \hat{T}_{ae})}$ ($\varphi^{lower}_{(\hat{t}_c, \hat{T}_{ae})}$) are found whose arcs belong respectively to the upper (lower) boundary of the playable set for the disturbance d_e in the game. The set *Unavoid_Pre* is bounded by curves of these families.

Conclusions

The synthesis procedure of (Lygeros, Tomlin, & Sastry 1998) provided a consistent framework for synthesizing controllers for our heater system. On reflection, we remark that the modeling formalism helps us provide a careful and precise model of the discrete and continuous interaction between the controller and the heater system. The iterative procedure offers a structure with well-defined steps to perform, guaranteeing that we do not overlook any potential winning strategies.

However, in the course of this case study, we encountered very practical obstacles in performing the steps required for the continuous aspects of the procedure. It was our original goal to analyze a system with two continuous temperatures (the air temperature and the temperature of the boiler) and two timers in the state vector. However, the continuous calculations for this model required complex geometric reasoning that quickly became extremely difficult, even for early iterations. The effort was temporarily abandoned. Attempts to handle the system with one temperature but two timers—one each for the controller and the plant—met with the same difficulty. The implementation in MATLAB required careful off-line analysis of the problem. It appears that automating such a procedure, even



Step 0 Step 1 Step 2

Figure 3: The procedure converges in three steps and returns the maximal safe set. The sets $\overline{W^i}|_q$, $(Pre_c(W^i) \cap W^i)|_q$ and $(Pre_e(W^i) \cap W^i)|_q$ are represented respectively in light gray, dark gray, and black.

for low-dimensional systems will involve a fair amount of manual reasoning to encode the necessary computations in MATLAB.

We conclude that the usefulness of the procedure is severely hampered by the computations of the *Unavoid_Pre* operator, even for our simple continuous dynamics. One approach is to attempt to develop efficient computational methods for finding *Unavoid_Pre* sets (Lygeros, Tomlin, & Sastry 1998). A second possibility is to simplify the computations by recasting the *Unavoid_Pre* operator in lower dimensions. Our initial efforts in this direction have met with some success.

Alternatively, one could forsake exact calculation of *Unavoid_Pre* as being too inefficient. The original model could be approximated using a specialized subclass for which the computations are feasible. Since automated computation for piecewise-constant differential inclusions without even continuous input control (Wong-Toi 1997) is already challenging, likely the abstraction needs to be an entirely discrete system, or perhaps a timed system. Perhaps a more promising option is to develop a suitable method for finding dynamic approximations of the *Unavoid_Pre* operator that are accurate, conservative, and efficient to compute.

References

Asarin, E.; Maler, O.; Pnueli, A.; and Sifakis, J. 1998. Controller synthesis for timed automata. In *Proc. of System Structure and Control.* IFAC.

Büchi, J. R., and Landweber, L. H. 1969. Solving sequential conditions by finite-state strategies. *Trans. of the American Mathematical Society* 138:295–311.

Heymann, M.; Lin, F.; and Meyer, G. 1997. Control synthesis for a class of hybrid systems subject to configuration-based safety constraints. In Maler, O., ed., *HART 97: Hybrid and Real-Time Systems*, LNCS 1201, 376–390. Springer-Verlag.

Isaacs, R. 1967. *Differential Games.* John Wiley.

Lygeros, J.; Godbole, D.; and Sastry, S. 1996. A game-theoretic approach to hybrid systems design. In Alur, R.; Henzinger, T. A.; and Sontag, E. D., eds., *Proc. of Hybrid Systems III: Verification and Control*, LNCS 1066, 1–12. Springer-Verlag.

Lygeros, J.; Tomlin, C.; and Sastry, S. 1998. On controller synthesis for nonlinear hybrid systems. In *Proc. of 37th Conf. on Decision and Control*, 2101–2106.

Lygeros, J.; Tomlin, C.; and Sastry, S. 1999. Controllers for reachability specifications for hybrid systems. *Automatica* 35(3). To appear.

Maler, O.; Pnueli, A.; and Sifakis, J. 1995. On the synthesis of discrete controllers for timed systems. In Mayr, E., and Puech, C., eds., *Proc. of 12th Annual Symposium on Theoretical Aspects of Computer Science*, LNCS 900, 229–242. Springer-Verlag.

Nerode, A., and Yakhnis, A. 1992. Modelling hybrid systems as games. In *Proc. of 31st Conf. on Decision and Control*, 2947–2952.

Pnueli, A., and Rosner, R. 1989. On the synthesis of a reactive module. In *ACM Symposium on Principles of Programming Languages*, 179–190.

Rabin, M. 1972. *Automata on Infinite Objects and Church's Problem.* Nr 13 in Regional Conference Series in Mathematics. American Mathematical Society.

Ramadge, P., and Wonham, W. 1989. The control of discrete event systems. *Proc. of the IEEE* 77(1):81–98.

Tomlin, C.; Lygeros, J.; and Sastry, S. 1998. Synthesizing controllers for nonlinear hybrid systems. In Henzinger, T., and Sastry, S., eds., *Hybrid Systems: Computation and Control*, LNCS 1386, 360–373.

Vincent, T. L., and Grantham, W. J. 1997. *Nonlinear and Optimal Control Systems.* John Wiley.

Wong-Toi, H. 1997. The synthesis of controllers for linear hybrid automata. In *Proc. of 36th Conf. on Decision and Control*, 4607–4612.