

# A Partial Order Approach for Low Complexity Control of Block Triangular Hybrid Automata

Domitilla Del Vecchio  
University of Michigan, USA

**Abstract**—This paper addresses the safety control problem for a class of block triangular order preserving hybrid automata. In particular, the partial order structure of the system is exploited to obtain two main contributions. First, conditions are provided for the termination of the proposed algorithm. Second, the control algorithm has linear complexity in the number of variables.

## I. INTRODUCTION

The problem addressed in this paper is the control of the parallel composition of a class of hybrid automata (triangular order preserving hybrid automata) under safety specifications. Motivating applications both for the model and for the problem considered include multi-agent hybrid systems such as intelligent transportation systems and railway control systems. In these systems, each agent (a vehicle) can be modeled as a hybrid automaton, in which the continuous state dynamics has triangular structure and models the physical motion of the agent. The discrete state can model a control mode in which the agent can be (turning, accelerating, run-out, etc.) or it can model input and state constraints. The entire system is given as the parallel composition of the component systems modeling the agents. In particular, one problem for which automated solutions are sought ([1], [2]) is the collision prediction and avoidance at traffic intersections and at railway mergings.

The control problem under safety specifications can be addressed by computing the set of states that lead to an unsafe configuration independently of an input choice (called the backward reachable set [3], [4] or the uncontrollable predecessor [5] of an unsafe set). Then, a feedback is computed that guarantees that the state never enters such a set [6], [7]. As it appears from these previous works, there are two main difficulties in solving this problem: Complexity and lack of termination guarantees. There is a large body of literature about safety control design and the list here provided is not exhaustive. The reachability and backward reachability problem is undecidable even for simple subclasses of hybrid automata [8]. For classes of hybrid automata for which the continuous dynamics reachable set can be computed, computational constraints usually limit the system to four or five continuous variables and to two or three discrete states. Furthermore, the proposed algorithms are not guaranteed to terminate [3], [6]. To reduce the computational load, approximate algorithms have been proposed to compute an over-approximation of the backward reachable set of the

unsafe set [9]–[11]. However, the obtained algorithms only provide semi-decision procedures as they are not guaranteed to terminate.

In this paper, we propose a solution to the safety control problem, which exploits the triangular order preserving structure of the system dynamics to address complexity and termination issues. In particular, our solution also relies on the approximated computation of the set of states that lead to a bad state independently of the input, here called “the escape set”. In contrast to previous work, however, our algorithm is guaranteed to terminate while being provably correct. Furthermore its complexity is linear in the number of continuous and discrete variables as opposed to being exponential as it occurs with algorithms that consider general classes of hybrid systems [4]. An explicit expression of the resulting feedback controller is provided. Two simple application examples involving conflict avoidance at a traffic intersection and at a railway merging are proposed. These examples serve two main purposes. On the one hand, they illustrate that practically relevant situations may be treated with this approach. On the other hand, they show that the control law is not conservative, even if it has been computed on the basis of an over-approximation of the escape set.

This paper is organized as follows. In Section II, we introduce basic notions. In section III, we introduce the block triangular order preserving hybrid automaton model. For this class of systems, we compute in Section IV the escape set approximation and the resulting control algorithm. Finally, we show two application examples in Section V.

## II. TRANSITION SYSTEMS, PARTIAL ORDERS, AND ESCAPE SETS

A partial order [12] is a set  $P$  with a partial order relation “ $\leq$ ”, and we denote it by the pair  $(P, \leq)$ . For all  $x, w \in P$ , the  $\sup\{x, w\}$ , denoted  $x \vee w$ , is the smallest element that is larger than both  $x$  and  $w$ . The  $\inf\{x, w\}$ , denoted  $x \wedge w$ , is the largest element that is smaller than both  $x$  and  $w$ . If  $S \subseteq P$ ,  $\bigvee S := \sup S$  and  $\bigwedge S := \inf S$ . If  $x \wedge w \in X$  and  $x \vee w \in X$  for all  $x, w \in X$ , then  $(X, \leq)$  is a *lattice*. Any interval sublattice of  $(P, \leq)$  is given by  $[L, U] = \{w \in P \mid L \leq w \leq U\}$  for  $L, U \in P$ . That is, this special sublattice can be represented by only two elements. Let  $(P, \leq)$  and  $(Q, \leq)$  be partially ordered sets. A map  $f : P \rightarrow Q$  is (i) an *order preserving map* (*order reversing map*) if  $x \leq w \implies f(x) \leq f(w)$  ( $x \leq w \implies f(x) \geq f(w)$ ); (ii) an *order isomorphism* if  $x \leq w \iff f(x) \leq f(w)$  and it maps  $P$  onto  $Q$ ; (iii) *order continuous* if  $f(\bigvee S) = \bigvee f(S)$  and  $f(\bigwedge S) = \bigwedge f(S)$  for

This work was in part supported by the Crosby Award at University of Michigan and by the NSF CAREER award number CNS-0642719.

$S \subseteq P$ . An order isomorphism is always order continuous. A particular partial order that we will consider in the sequel, is the power set of a set  $S$ , that is, the set of all subsets of  $S$ , denoted  $2^S$ , ordered according to inclusion relation. This partial order will be denoted by  $(2^S, \subseteq)$ . In this partial order, the supremum operators ( $\vee$  and  $\bigvee$ ) are given by set union  $\cup$  and the infimum operators ( $\wedge$  and  $\bigwedge$ ) are given by set intersection  $\cap$ . For a map  $f : P \rightarrow P$  with  $(P, \leq)$  a partial order, we call *fix-point* an element  $x \in P$  such that  $f(x) = x$ . The least fix-point of  $f$  is denoted by  $\text{lfp}(f)$  and the greatest fix-point is denoted by  $\text{gfp}(f)$ .

We introduce the *escape set* for the general modeling formalism of transition systems (see [5], for example) as the notion of escape set is independent on whether the system has continuous or discrete variables. We denote a *transition system* by the tuple  $\Sigma = (S, \mathcal{I}, \tau)$ , in which  $S$  is a (possibly infinite) set of states,  $\mathcal{I}$  is a set of inputs, and  $\tau : S \times \mathcal{I} \rightarrow S$  is a transition map. We denote a state by  $s \in S$  and an input by  $u \in \mathcal{I}$ . An execution of  $\Sigma$  is an infinite sequence  $\{s^k\}_{k \in \mathbb{N}}$  such that  $s^{k+1} = \tau(s^k, u^k)$  for  $u^k \in \mathcal{I}$ . An infinite input sequence is denoted by  $\{u^k\}_{k \in \mathbb{N}}$ . Let  $B \subseteq S$  be a set of bad states and let us assume that once a state is in  $B$ , it cannot recover from  $B$ , that is,  $s \in B \Rightarrow \tau(s, u) = s, \forall u \in \mathcal{I}$ . For all  $n \geq 0$  we define  $\tau^n(s, \{u^k\}_{k \leq n})$  by the following relations  $\tau^0(s, u^0) := s \vee u^0 \in \mathcal{I}$ ,  $\tau^n(s, \{u^k\}_{k \leq n}) := \tau(\tau^{n-1}(s, \{u^k\}_{k \leq n-1}), u^n)$ .

**Definition 1:** We call *escape set* the set of states for which all possible input sequences will lead to a bad set  $B \subseteq S$  in finite time. It is characterized by  $E = \{s \in S \mid \forall \{u^k\}_{k \in \mathbb{N}} \exists N \text{ such that } \tau^N(s, \{u^k\}_{k \leq N}) \in B\}$ .

**Problem 1:** The safety control problem for system  $\Sigma = (S, \mathcal{I}, \tau)$  with bad set  $B$ , is the one of designing a feedback law  $u = g(s)$  such that for all executions  $\{s^k\}_{k \in \mathbb{N}}$  starting with  $s^0 \notin E$ , we have that  $s^k \notin E$  for all  $k$ .

Let us denote the set valued map  $\bar{\tau} : S \rightarrow 2^S$  by  $\bar{\tau}(s) := \tau(s, \mathcal{I}) = \bigcup_{u \in \mathcal{I}} \tau(s, u)$  and by  $\bar{\tau}^n(s) := \tau^n(s, \mathcal{I}) = \tau(\tau^{n-1}(s, \mathcal{I}), \mathcal{I})$  with  $\bar{\tau}^0(s) := s$  for all  $s \in S$ . Since once a state is in  $B$  it cannot recover from  $B$ , it is the case that  $s \in E$  if and only if there is a finite  $N$  such that  $\bar{\tau}^N(s) \subseteq B$  (if such  $N$  did not exist, we would have had one infinite input sequence  $\{u^k\}_{k \in \mathbb{N}}$  such that  $\tau^n(s, \{u^k\}_{k \leq n}) \notin B$  for all  $n$ , which means by Definition 1 that  $s$  cannot be in  $E$ ). We can thus re-define the set  $E$  as

$$E = \{s \in S \mid \exists N < \infty \text{ such that } \bar{\tau}^N(s) \subseteq B\}, \quad (1)$$

and we introduce the notation

$$\text{pre}(\bar{\tau}^n)(B) := \{s \in S \mid \bar{\tau}^n(s) \subseteq B\} \quad (2)$$

to denote the set of states  $s$  that reach the bad set  $B$  in at most  $n$  steps. The following theorem provides a mathematical characterization of the escape set and a tool to compute it.

**Theorem 1:** The map  $\text{pre}(\bar{\tau})$  that attaches to the bad set  $B \subseteq S$  the set  $\text{pre}(\bar{\tau})(B) \subseteq S$  as defined in equation (2) with  $n = 1$  is order preserving with respect to partial order  $(2^S, \subseteq)$ . Furthermore, the set  $E$  of equation (1) is characterized by

$$E = \bigvee_{n \geq 0} \text{pre}(\bar{\tau}^n)(B) = \text{lfp}(F), \quad (3)$$

in which  $F(a) := B \vee \text{pre}(\bar{\tau})(a)$  for  $a \subseteq S$ .

This theorem can be proved similarly to Theorem 10-7 in [13]. One can also verify that

$$E = \bigcup_{k=0}^{\infty} E^k, \text{ with } E^k = \text{pre}(\bar{\tau})\left(\bigcup_{i=0}^{k-1} E^i\right), \text{ and } E^0 = B. \quad (4)$$

There are two major difficulties in the computation of the set  $E$  for infinite state systems: (a) the representation of  $E^k$  (if computable) may grow in complexity even exponentially with  $k$ ; (b) iteration (4) even if converging by virtue of Theorem 1, might not converge in a finite number of steps. If the sets  $E^k$  are not computable, the problem is said to be undecidable. If instead the sets  $E^k$  can be algorithmically computed, but the above iteration is not guaranteed to terminate, the problem is said to be semi-decidable.

In this paper, we provide a solution to Problem 1 for a class of block triangular hybrid automata, which also uses an over-approximation  $\bar{E}$  in place of the escape set  $E$ . In contrast to previous work, our algorithm is *guaranteed to terminate* while being provably correct. The computational complexity of the proposed decision procedure is *linear in the number of variables*. The main structural conditions on the system are: (1) it is the parallel composition of hybrid automata with upper triangular structure and with order preserving dynamics; (2) the continuous input and the mode of each composing system enters only the last one of the continuous state variable updates; (3) the discrete mode update map has no memory of the previous mode.

### III. HYBRID AUTOMATON MODEL

We start by defining the discrete time hybrid automaton in a way analogous to the continuous time counterpart [3].

**Definition 2:** A *discrete time hybrid automaton* is a tuple  $H = (Q, X, \mathcal{I}, \iota, f, \text{Dom}, R)$ , in which  $Q = \{q_1, \dots, q_m\}$  is a set of discrete states (or modes);  $X = \mathbb{R}^p$  is the set of continuous states;  $\mathcal{I} = \mathcal{I}_D \times \mathcal{I}_C$ , is the set of discrete and continuous inputs, respectively;  $\iota : Q \rightarrow 2^{\mathcal{I}}$  is a function that attached to each discrete state the set of enabled inputs;  $f : Q \times X \times \mathcal{I}_C \rightarrow X$  is the continuous state update function;  $\text{Dom} : Q \rightarrow 2^X$  is a map that for each mode establishes the domain in  $X$  in which such mode holds;  $R : Q \times X \times \mathcal{I}_D \rightarrow Q$  is the discrete state update map, which for any current discrete state, continuous state, and input determines the new discrete state.

We denote by  $q \in Q$  the mode, by  $x \in X$  the continuous state, by  $u \in \mathcal{I}_C$  the continuous inputs, and by  $\sigma \in \mathcal{I}_D$  the discrete input. We assume that  $R$  is static, that is, it does not contain memory of previous discrete states. Thus, we have that  $q = R(x, \sigma)$ . We make an explicit distinction between two types of modes: the modes  $q$  such that  $\text{Dom}(q) = \mathbb{R}^p$  and the modes  $q$  such that  $\text{Dom}(q) \neq \mathbb{R}^p$ . In particular, we assume that a transition to a mode with  $\text{Dom}(q) = \mathbb{R}^p$  can happen *only* by a suitable choice of discrete input  $\sigma \in \mathcal{I}_D$ , while a transition to a mode with  $\text{Dom}(q) \neq \mathbb{R}^p$  can happen *only* autonomously and thus cannot be controlled. This is

formalized by the following structure of  $R$ :

$$R(x, \sigma) := \begin{cases} R(\sigma) & \text{if } \sigma \neq \emptyset \\ R(x) & \text{if } \sigma = \emptyset, \end{cases} \quad (5)$$

in which we define  $R(x) := q$  if  $x \in \text{Dom}(q)$ . One can verify that this update is *deterministic* if  $\text{Dom}(q_1) \cap \text{Dom}(q_2) = \emptyset$  whenever  $\text{Dom}(q_1) \neq \mathbb{R}^p$  and  $\text{Dom}(q_2) \neq \mathbb{R}^p$ . Also, we assume that for any mode with  $\text{Dom}(q) = \mathbb{R}^p$ , there exists a discrete input  $\sigma \in \mathcal{I}_D$  such that  $q = R(\sigma)$ . The *non-blocking* condition can be guaranteed if  $\bigcup_{q \mid \text{Dom}(q) \neq \mathbb{R}^p} \text{Dom}(q) = \mathbb{R}^p$ . In the sequel, we use the notation  $\mathcal{Q} := \{q \in \mathcal{Q} \mid \text{Dom}(q) \neq \mathbb{R}^p\}$ .

Hybrid automaton  $H$  corresponds to the transition system  $\Sigma_H = (S, \mathcal{I}, \tau)$ , in which  $S = X$ ,  $\mathcal{I} = \mathcal{I}_D \times \mathcal{I}_C$ . An input  $\bar{u} \in \mathcal{I}$  is a pair  $\bar{u} = (u, \sigma)$ , in which  $u \in \mathcal{I}_C$  and  $\sigma \in \mathcal{I}_D$ . The transition map is given by  $\tau(x, \bar{u}) := f(R(x, \sigma), x, u)$  with  $q = R(x, \sigma)$  and  $u \in \iota(q)$ . Thus, given a set of bad states  $B \subseteq X$ , the safety control problem (Problem 1) is the one of establishing a pair of feedback maps  $u = g_C(x, q)$  and  $\sigma = g_D(x)$  such that the state  $x^k$  along any execution  $\{x^k\}_{k \in \mathbb{N}}$  of  $\Sigma$  is outside the escape set  $E$  in equation (4) if  $x^0 \notin E$ . From expression (4), it appears that we need to compute the set  $\text{pre}(\bar{\tau})(P)$  for a set  $P \subseteq X$  to obtain the escape set. We thus explicitly write the form of  $\text{pre}(\bar{\tau})(P)$  for the transition system  $\Sigma_H$ . For a set  $P \subseteq X$  the set  $\text{pre}(\bar{\tau})(P)$  defined in (2) with  $n = 1$  takes the form  $\text{pre}(\bar{\tau})(P) = \{x \in \mathbb{R}^p \mid f(q, x, u) \in P \text{ with } q = R(x, \sigma), \forall \sigma \in \mathcal{I}_D, \forall u \in \iota(q)\}$ , which is equivalent to

$$\begin{aligned} \text{pre}(\bar{\tau})(P) &= \bigcup_{\bar{q}} (\{x \mid f(q, x, u) \in P, \forall u \in \iota(q)\} \\ &\cap \text{Dom}(q)) \cap \bigcap_{q \mid \text{Dom}(q) = \mathbb{R}^p} \{x \mid f(q, x, u) \in P, \forall u \in \iota(q)\}. \end{aligned} \quad (6)$$

A guaranteed over-approximation of this set can be efficiently computed if the function  $f$  is triangular and order preserving as we show in the next section.

#### A. Block triangular order preserving hybrid automata

**Definition 3:** A *triangular order preserving hybrid automaton* is a hybrid automaton  $H = (\mathcal{Q}, X, \mathcal{I}, \iota, f, \text{Dom}, R)$ , in which

(i) The update map  $f(q, x, u)$  for every mode  $q$  and  $x = (x_1, \dots, x_n) \in \mathbb{R}^n$  has the following triangular structure  $f(q, x, u) = (f_1(x_1, \dots, x_n), \dots, f_i(x_i, \dots, x_n), \dots, f_n(x_n, q, u))$ , in which  $f_i : \mathbb{R}^{n-(i-1)} \rightarrow \mathbb{R}$  for  $i \in \{1, \dots, n-1\}$ ,  $f_n : \mathbb{R} \times \mathcal{Q} \times \mathcal{I}_C \rightarrow \mathbb{R}$  with  $\mathcal{I}_C = \mathbb{R}$ , and  $\text{Dom}(q) \subseteq \mathbb{R}^n$ .

(ii) We consider the component-wise partial ordering on  $\mathbb{R}^n$ , and the usual order on  $\mathbb{R}$ . We assume that the set of discrete states with  $\text{Dom}(q) = \mathbb{R}^n$  is a lattice with minimum  $\alpha$  and with maximum  $\beta$ , that is,  $\{q \in \mathcal{Q} \mid \text{Dom}(q) = \mathbb{R}^n\} = [\alpha, \beta]$ . For all  $q \in \mathcal{Q}$ , we assume that  $\iota(q)$  is an interval in  $\mathbb{R}$ , that is,  $\iota(q) = [u_L(q), u_U(q)]$ . Also, the functions  $u_L(\cdot)$  and  $u_U(\cdot)$  are order preserving for  $q$  with  $\text{Dom}(q) = \mathbb{R}^n$ .

(iii) We assume that  $f_i$  is order preserving in all its arguments, that is if  $(x_i^a, \dots, x_n^a) \leq (x_i^b, \dots, x_n^b)$  then  $f_i(x_i^a, \dots, x_n^a) \leq f_i(x_i^b, \dots, x_n^b)$  for  $i < n$ , and  $f_n(x_n^a, q, u) \leq f_n(x_n^b, q, u)$ . Also,  $f_n : \mathcal{Q} \mid_{\{q \in \mathcal{Q} \mid \text{Dom}(q) = \mathbb{R}^n\}} \times \mathbb{R} \times \mathcal{I}_C \rightarrow \mathbb{R}$  is order preserving in

all its arguments. Additionally,  $f_i$  is one-one and onto in  $x_i$ , that is, fixed  $x_{i+1}, \dots, x_n, q, u$ , for any  $x_i'$  there is one and only one  $x_i$  such that  $f_i(x_i, \dots, x_n) = x_i'$  if  $i < n$  or  $f_i(x_i, q, u) = x_i'$  if  $i = n$ . We denote the first one by  $f_i^{-1}(x_i', x_{i+1}, \dots, x_n)$  and the second one by  $f_i^{-1}(x_i', q, u)$ .

(iv) The maps  $f_i$  are non-decreasing:  $f_i(x_i, \dots, x_n) \geq x_i$ , for  $i < n$  and  $f_n(x_n, q, u_U(q)) > x_n$  for all  $q$ .

A partial order preserving update map corresponds to a monotone continuous time dynamical system [14]. The extensive studies on monotone systems have been in part motivated by the fact that they can model competitive and cooperative systems. The continuous input is allowed to enter only the update map for  $x_n$  as it occurs in feedback linearized systems. The parallel composition of a number of triangular order preserving hybrid automata generates a block-triangular order preserving hybrid automaton. This is made more precise by defining the parallel composition of hybrid automata in a way similar to [15].

**Definition 4:** Let  $H_1 = (\mathcal{Q}_1, X_1, \mathcal{I}_1, \iota_1, f_1, \text{Dom}_1, R_1)$  and  $H_2 = (\mathcal{Q}_2, X_2, \mathcal{I}_2, \iota_2, f_2, \text{Dom}_2, R_2)$  be two hybrid automata. The parallel composition, denoted  $H = H_1 \parallel H_2$ , is given by  $H = (\mathcal{Q}, X, \mathcal{I}, \iota, f, \text{Dom}, R)$ , in which  $\mathcal{Q} = \mathcal{Q}_1 \times \mathcal{Q}_2$ ,  $X = X_1 \times X_2$ ,  $\mathcal{I} = \mathcal{I}_C \times \mathcal{I}_D$  with  $\mathcal{I}_C = \mathcal{I}_{C,1} \times \mathcal{I}_{C,2}$  and  $\mathcal{I}_D = \mathcal{I}_{D,1} \times \mathcal{I}_{D,2}$ ;  $\iota : \mathcal{Q} \rightarrow \mathcal{I}_C$  is given by  $\iota = (\iota_1, \iota_2)$ ;  $f : \mathcal{Q} \times X \times \mathcal{I}_C \rightarrow X$  is given by  $f = (f_1, f_2)$ ;  $\text{Dom}(q) = \text{Dom}_1(q_1) \times \text{Dom}_2(q_2)$ ;  $R(x, \sigma) = (R_1(x_1, \sigma_1), R_2(x_2, \sigma_2))$ .

**Definition 5:** A *block triangular order preserving hybrid automaton* is the parallel composition of  $N$  triangular order preserving hybrid automata  $H_1, \dots, H_N$ .

Let  $x_i = (x_{1,i}, \dots, x_{n,i}) \in \mathbb{R}^n$ ,  $q_i \in \mathcal{Q}_i$ ,  $u_i \in \iota(q_i)$ ,  $\sigma_i \in \mathcal{I}_{D,i}$  represent the continuous state, the discrete state, the continuous input, and the discrete input of the triangular hybrid automaton  $H_i$ , respectively. Then, in each mode  $q = (q_1, \dots, q_N)$  of the hybrid automaton  $H = H_1 \parallel \dots \parallel H_N$ , the continuous state update map has the following form

$$\begin{aligned} x'_{j,i} &= f_{j,i}(x_{j,i}, \dots, x_{n,i}), \quad j < n \quad i \in \{1, \dots, N\} \\ x'_{n,i} &= f_{n,i}(x_{n,i}, q_i, u_i), \quad i \in \{1, \dots, N\}, \end{aligned} \quad (7)$$

in which primed variables denote updated variables.

For this system, we model the safety requirement by requesting that the state  $x$  never enter the bad set

$$\begin{aligned} B &= \{(x_{1,1}, \dots, x_{n,1}, \dots, x_{1,N}, \dots, x_{n,N}) \mid (x_{1,1}, \dots, x_{1,N}) \in \bar{B}\}, \\ \bar{B} &= [L_1, U_1] \times \dots \times [L_N, U_N], \text{ with } L_i, U_i \in \mathbb{R}. \end{aligned} \quad (8)$$

This choice of the bad set to involve only the variables  $(x_{1,1}, \dots, x_{1,N})$  is motivated by the applications that we are targeting (see Section V).

## IV. CONTROL DESIGN

In this section, we construct the control map by computing an approximation  $\bar{E}$  of the escape set  $E$ . We show that  $E \subseteq \bar{E}$  by showing that if the state is in  $\bar{E}$  then a control input exists that maps the state outside  $\bar{E}$ . Let  $B$  be as given in equations (8). Denote  $F_i(x_{2,i}, \dots, x_{n,i}, q_i, u_i) := (f_{2,i}(x_{2,i}, \dots, x_{n,i}), \dots, f_{n,i}(x_{n,i}, q_i, u_i))$  and  $F_i^k(\bar{x}_i, q_i, u_i) := F(F^{k-1}(\bar{x}_i, q_i, u_i), q_i, u_i)$ , with

$\bar{x}_i = (x_{2,i}, \dots, x_{n,i})$ . Then we have that  $\bar{E} = \{(x_{1,1}, \dots, x_{n,1}, \dots, x_{1,N}, \dots, x_{n,N}) \mid (x_{1,1}, \dots, x_{1,N}) \in \bar{E}^*(x)\}$ , in which  $\bar{E}^*(x)$  is given by the following algorithm.

**Algorithm 1.**

$\bar{E}^*(x) = \bigcup_{k=0}^{k^*} [\bar{L}^k, \bar{U}^k]$ ,  $\bar{L}^0 = L$ ,  $\bar{U}^0 = U$ ,  $\bar{L}^k = (\bar{L}_1^k, \dots, \bar{L}_N^k)$ ,  $\bar{U}^k = (\bar{U}_1^k, \dots, \bar{U}_N^k)$  with

$$\bar{L}_i^1(\bar{x}_i) = f_{1,i}^{-1}(L_i^0, \bar{x}_i)$$

$$\bar{U}_i^1(\bar{x}_i) = f_{1,i}^{-1}(U_i^0, \bar{x}_i),$$

while for  $k > 1$ , we have

$$L_i^k(\bar{x}_i) = L_i^{k,a}(\bar{x}_i) \vee L_i^{k,b}(\bar{x}_i) \quad (9)$$

$$L_i^{k,a}(\bar{x}_i) = \bigwedge_{q_i \in \bar{Q}_i} f_{1,i}^{-1}(\bar{L}_i^{k-1}(F_i(\bar{x}_i, q_i, u_L(q_i))), \bar{x}_i) \quad (10)$$

$$L_i^{k,b}(\bar{x}_i) = f_{1,i}^{-1}(\bar{L}_i^{k-1}(F_i(\bar{x}_i, \alpha_i, u_L(\alpha_i))), \bar{x}_i) \quad (11)$$

$$U_i^k(\bar{x}_i) = U_i^{k,a}(\bar{x}_i) \wedge U_i^{k,b}(\bar{x}_i) \quad (12)$$

$$U_i^{k,a}(\bar{x}_i) = \bigvee_{q_i \in \bar{Q}_i} f_{1,i}^{-1}(\bar{U}_i^{k-1}(F_i(\bar{x}_i, q_i, u_U(q_i))), \bar{x}_i) \quad (13)$$

$$U_i^{k,b}(\bar{x}_i) = f_{1,i}^{-1}(\bar{U}_i^{k-1}(F_i(\bar{x}_i, \beta_i, u_U(\beta_i))), \bar{x}_i) \quad (14)$$

with (removing the dependence on  $\bar{x}_i$  for shortness of notation)

$$\bar{L}_i^k = \inf(L_i^k, \bar{L}_i^{k-1}) \quad (15)$$

$$\bar{U}_i^k = \begin{cases} \sup(U_i^k, \bar{L}_i^{k-1}), & \text{if } \exists j \text{ such that } U_j^k > \bar{L}_j^{k-1} \\ U_i^k, & \text{if } U_j^k \leq \bar{L}_j^{k-1} \forall j, \end{cases} \quad (16)$$

with  $k^*$  the smallest  $k$  such that

$$U_i^k \leq \bar{L}_i^{k-1} \forall i \text{ and } \exists j \text{ such that } \bar{U}_j^{k+1} < \bar{L}_j^{k+1}.$$

For a fixed  $x$ , the set  $\bar{E}^*(x)$  is the union of  $k^*$  rectangles in  $\mathbb{R}^N$ . The expressions (9) and (12) of the extremes of such rectangles depend on the values of the variables  $(x_{2,i}, \dots, x_{n,i})$ . For computation, one can off-line symbolically compute the iterative expressions (9) and (12) and evaluate them only when the value of  $(x_{2,i}, \dots, x_{n,i})$  becomes available on-line. The set  $\bar{E}$  is obtained by computing at each iteration  $(n-1)N$  computations for computing  $f_{j,i}$  for  $j > 1$  and for  $i \in [2, N]$ . This procedure has thus linear complexity with the number of continuous variables. To prove termination, we make the following assumptions, which can be statically checked. First, define the following notation for  $y \in \mathbb{R}$  and  $z_j \in \mathbb{R}^{n-1}$  for  $j \in \mathbb{N}$

$$\phi_i(y, z_0) := f_{1,i}^{-1}(y, z_0)$$

$$\phi_i^k(y, z_{k-1}, \dots, z_0) := f_{1,i}^{-1}(\phi_i^{k-1}(y, z_{k-1}, \dots, z_1), z_0). \quad (17)$$

**Assumption 1:** Let  $y_1, y_2, y_3, y_4 \in \mathbb{R}^n$  with  $y_1 - y_2 < y_3 - y_4$ . Then, we have that  $f_1^{-1}(y_1, x_{2,i}^A, \dots, x_{n,i}^A) - f_1^{-1}(y_2, x_{2,i}^B, \dots, x_{n,i}^B) < f_1^{-1}(y_3, x_{2,i}^A, \dots, x_{n,i}^A) - f_1^{-1}(y_4, x_{2,i}^B, \dots, x_{n,i}^B)$ , for all  $(x_{2,i}^A, \dots, x_{n,i}^A), (x_{2,i}^B, \dots, x_{n,i}^B) \in \mathbb{R}^{n-1}$

**Assumption 2:** Let  $y_1, y_2 \in \mathbb{R}^n$  with  $y_2 \geq y_1$ , and let  $(x_{2,i}^A, \dots, x_{n,i}^A) > (x_{2,i}^B, \dots, x_{n,i}^B)$ . Then  $f_1^{-1}(y_2, x_{2,i}^A, \dots, x_{n,i}^A) - f_1^{-1}(y_1, x_{2,i}^B, \dots, x_{n,i}^B) < y_2 - y_1$ .

**Assumption 3:** For all  $i$ ,  $\bigwedge_{q_i \in \bar{Q}_i} f_{n,i}(x_{n,i}, q_i, u_U(q_i)) > \bigvee_{q_i \in \bar{Q}_i} f_{n,i}(x_{n,i}, q_i, u_L(q_i))$ .

**Assumption 4:** Let

$$z_{k+l_1-1} = F_i^{l_1-1}(F_i^k(\bar{x}_i, \beta_i, u_U(\beta_i)), \alpha_i, u_L(\alpha_i))$$

$$z_{k+l_1-2} = F_i^{l_1-2}(F_i^k(\bar{x}_i, \beta_i, u_U(\beta_i)), \alpha_i, u_L(\alpha_i))$$

$\vdots$

$$z_{k-1} = F_i^k(\bar{x}_i, \beta_i, u_U(\beta_i))$$

$\vdots$

$$z_2 = F_i(\bar{x}_i, \beta_i, u_U(\beta_i))$$

$$z_0 = \bar{x}_i,$$

$$w_{k+l_2-1} = F_i^{k+l_2-1}(\bar{x}_i, \alpha_i, u_L(\alpha_i))$$

$$w_{k+l_2-2} = F_i^{k+l_2-2}(\bar{x}_i, \alpha_i, u_L(\alpha_i))$$

$\vdots$

$$w_0 = \bar{x}_i,$$

then we assume that for all  $y \in \mathbb{R}$  the sequence  $\{\phi_i^{k+l_1}(y, z_{k+l_1-1}, \dots, z_0) - \phi_i^{k+l_2}(y, w_{k+l_2-1}, \dots, w_0)\}_{k>1}$  is strictly decreasing for all  $l_1, l_2 > 0$ . Let now re-define the  $z_j$  and the  $w_j$  by replacing  $\beta_i$  by  $q_i^A$  and  $\alpha_i$  by  $q_i^B$ . Then, we assume that the sequence  $\{\bigvee_{q_i^A \in \bar{Q}_i} \bigwedge_{q_i^B \in \bar{Q}_i} \phi_i^{k+l_1}(y, z_{k+l_1-1}, \dots, z_0) - \bigwedge_{q_i^B \in \bar{Q}_i} \phi_i^{k+l_2+1}(y, w_{k+l_2-1}, \dots, w_0)\}_{k>1}$  is also strictly decreasing for all  $l_1, l_2 > 0$ .

The meaning of Assumption 3 and 4 is basically that the difference between the maximum and minimum control actions applicable is enough to shrink (through backward iteration) an initial set in the state space. Since Algorithm 1 terminates when  $U_i^k \leq \bar{L}_i^{k-1}$  for all  $i$  and there is a  $j$  such that  $\bar{U}_j^{k+1} < \bar{L}_j^{k+1}$ , we will show that Assumptions 1, 2, and 3 guarantee that for all  $i$  there is a  $k$  such that  $U_i^k \leq \bar{L}_i^{k-1}$ . Then, Assumption 4 will be used to show that when  $U_i^k \leq \bar{L}_i^{k-1}$  for all  $i$ , there is a step  $l$  at which  $\bar{U}_i^{l+1} < \bar{L}_i^{l+1}$  for some  $j$ .

**Proposition 1:** Under Assumptions 1, 2, 3, and assuming  $\bar{U}_i^k(\bar{x}_i) = U_i^k(\bar{x}_i)$  for all  $k$ , then there is  $\bar{k}$  such that  $U_i^{\bar{k}}(\bar{x}_i) < \bar{L}_i^{\bar{k}}(\bar{x}_i)$  for all  $\bar{x}_i$  and all  $i$ .

*Proof:* If both controlled and autonomous switches are present, we have that  $[\bar{L}_i^k(\bar{x}_i), U_i^k(\bar{x}_i)] \subseteq [\bar{L}_i^{k,a}(\bar{x}_i), U_i^{k,a}(\bar{x}_i)]$  in which  $L_i^{k,a}$  and  $U_i^{k,a}$  are computed by Algorithm 1 assuming that only autonomous switches are present ( $\{q_i \mid \text{Dom}_i(q_i) = \mathbb{R}^n\} = \emptyset$ ), that is,  $L_i^{k,b} := -\infty$  and  $U_i^{k,b} := \infty$  for all  $k$ . This is true because, the sets  $[\bar{L}_i^k, U_i^k]$  are computed by intersecting  $[\bar{L}_i^{k,a}, U_i^{k,a}]$  with  $[\bar{L}_i^{k,b}, U_i^{k,b}]$  for all  $k$ . If only controlled switches are present ( $\bar{Q}_i = \emptyset$ ), then we have that  $[\bar{L}_i^k(\bar{x}_i), U_i^k(\bar{x}_i)] = [\bar{L}_i^{k,b}(\bar{x}_i), U_i^{k,b}(\bar{x}_i)]$ . Hence, to show that there is a  $k$  such that  $U_i^k(\bar{x}_i) < \bar{L}_i^k(\bar{x}_i)$ , it is enough to show that  $U_i^{k+1} - \bar{L}_i^{k+1} < U_i^k - \bar{L}_i^k$  when either only autonomous switches are present (setting  $L_i^{k,b} := -\infty$  and  $U_i^{k,b} := \infty$  for all  $k$  in Algorithm 1) or when only controlled switches are present (setting  $L_i^{k,b} := -\infty$  and  $U_i^{k,b} := \infty$  for all  $k$  in Algorithm 1). We consider here the case in which  $L_i^{k,b} := -\infty$  and  $U_i^{k,b} := \infty$  for all  $k$  in Algorithm 1, as the other case can be treated in a similar way.

If  $\bar{L}_i^{k+1}(\bar{x}_i) = \bar{L}_i^k(\bar{x}_i)$  (see equation (15)), we immediately have that  $U_i^{k+1}(\bar{x}_i) - \bar{L}_i^{k+1}(\bar{x}_i) < U_i^k(\bar{x}_i) - \bar{L}_i^k(\bar{x}_i)$  because the sequence  $\{U_i^k\}_{k>0}$  is strictly decreasing. This can be

shown by showing that  $\bigvee_{q_i \in \bar{Q}_i} f_{1,i}^{-1}(U_i^k(F_i(\bar{x}_i, q_i, u_U(q_i))), \bar{x}_i) < U_i^k(\bar{x}_i)$ . By the non-decreasing property of  $f_{n,i}$ , we obtain that  $f_{n,i}(x_{n,i}, q_i, u_U(q_i)) > x_{n,i}$  for all  $q_i$ . As a consequence, one can infer that  $U_i^k(F_i(\bar{x}_i, q_i, u_U(q_i))) < U_i^k(\bar{x}_i)$  for all  $q_i$  (this derives from the fact that the functions  $U_i^k(\cdot)$  are order reversing function of their arguments). By the fact that  $f_{1,i}(x_{1,i}, \dots, x_{n,i}) \geq x_{1,i}$  and that  $f_{1,i}$  is an order isomorphism in the first argument, we have the following set of inequalities:  $\bigvee_{q_i \in \bar{Q}_i} f_{1,i}^{-1}(U_i^k(F_i(\bar{x}_i, q_i, u_U(q_i))), \bar{x}_i) \leq \bigvee_{q_i \in \bar{Q}_i} U_i^k(F_i(\bar{x}_i, q_i, u_U(q_i))) \leq U_i^k(\bar{x}_i)$ , which give the desired result.

If  $\bar{L}_i^{k+1}(\bar{x}_i) = L_i^{k+1}(\bar{x}_i)$  (see equation (15)), it is enough to show that  $U_i^{k+1}(\bar{x}_i) - L_i^{k+1}(\bar{x}_i) < U_i^k(\bar{x}_i) - L_i^k(\bar{x}_i)$  because  $U_i^k(\bar{x}_i) - L_i^k(\bar{x}_i) \leq U_i^k(\bar{x}_i) - \bar{L}_i^k(\bar{x}_i)$  by the fact that  $\bar{L}_i^k(\bar{x}_i) \leq L_i^k(\bar{x}_i)$  (see equation (15)). We thus need to show that

$$\begin{aligned} & \bigvee_{q_i \in \bar{Q}_i} f_{1,i}^{-1}(\bar{U}_i^k(F_i(\bar{x}_i, q_i, u_U(q_i))), \bar{x}_i) - \\ & \bigwedge_{q_i \in \bar{Q}_i} f_{1,i}^{-1}(\bar{L}_i^k(F_i(\bar{x}_i, q_i, u_L(q_i))), \bar{x}_i) < \\ & \bigvee_{q_i \in \bar{Q}_i} f_{1,i}^{-1}(\bar{U}_i^{k-1}(F_i(\bar{x}_i, q_i, u_U(q_i))), \bar{x}_i) - \\ & \bigwedge_{q_i \in \bar{Q}_i} f_{1,i}^{-1}(\bar{L}_i^{k-1}(F_i(\bar{x}_i, q_i, u_L(q_i))), \bar{x}_i). \end{aligned} \quad (18)$$

Since  $f_{1,i}^{-1}$  is an order isomorphism in its first argument, then we also have that  $f_{1,i}^{-1}(\bigvee S, \bar{x}_i) = \bigvee f_{1,i}^{-1}(S, \bar{x}_i)$  and  $f_{1,i}^{-1}(\bigwedge S, \bar{x}_i) = \bigwedge f_{1,i}^{-1}(S, \bar{x}_i)$  for all  $S \subseteq \mathbb{R}$ . As a consequence, we can use Assumption 1 to infer that relation (18) holds if

$$\begin{aligned} & \bigvee_{q_i \in \bar{Q}_i} U_i^k(F_i(\bar{x}_i, q_i, u_U(q_i))) - \bigwedge_{q_i \in \bar{Q}_i} \bar{L}_i^k(F_i(\bar{x}_i, q_i, u_L(q_i))) < \\ & \bigvee_{q_i \in \bar{Q}_i} U_i^{k-1}(F_i(\bar{x}_i, q_i, u_U(q_i))) - \bigwedge_{q_i \in \bar{Q}_i} \bar{L}_i^{k-1}(F_i(\bar{x}_i, q_i, u_L(q_i))). \end{aligned} \quad (19)$$

Then, we have that relation (19) is holding if (proceeding iteratively)

$$\begin{aligned} & f_{1,i}^{-1}(U_i^0, \bigwedge_{q_i \in \bar{Q}_i} F_i^k(\bar{x}_i, q_i, u_U(q_i))) - \\ & f_{1,i}^{-1}(L_i^0, \bigvee_{q_i \in \bar{Q}_i} F_i^k(\bar{x}_i, q_i, u_L(q_i))) < U_i^0 - L_i^0. \end{aligned} \quad (20)$$

By virtue of Assumption 3 and by virtue of the fact that  $F_i^k$  is continuous and a composition of order preserving functions, we obtain that  $\bigwedge_{q_i \in \bar{Q}_i} F_i^k(\bar{x}_i, q_i, u_U(q_i)) > \bigvee_{q_i \in \bar{Q}_i} F_i^k(\bar{x}_i, q_i, u_L(q_i))$ . As a consequence, relation (20) holds by virtue of Assumption 2. ■

**Theorem 2:** (Termination) Under Assumptions 1, 2, 3, and 4, Algorithm 1 terminates, that is,  $k^*$  is finite.

*Proof:* We omit the dependencies on  $(x_{2,i}, \dots, x_{n,i})$  to simplify notation. The proof is composed of two main steps: (1) we show that there is a  $\bar{k} < \infty$  such that  $U_i^k \leq \bar{L}_i^{k-1}$ ,  $U_i^k \leq \bar{L}_i^{k-1}$  for all  $k > \bar{k}$  and for all  $i$  (thus the update rule becomes the second one of (16)); (2) we show that if  $U_i^k \leq \bar{L}_i^{k-1}$  for all  $i$  and all  $k > \bar{k}$ , then there is a finite  $k^* > \bar{k}$  such that  $\bar{U}_i^{k^*} < \bar{L}_i^{k^*}$  for some  $i$ .

(1) We show that there is a  $\bar{k} < \infty$  such that  $U_i^k \leq \bar{L}_i^{k-1}$  for some arbitrary  $i$ . Assume that  $U_i^k > \bar{L}_i^{k-1}$ , then  $\bar{U}_i^k = U_i^k$ . By Proposition 1, we have that there must be a  $k_i$  such that  $\bar{U}_i^{k_i} \leq \bar{L}_i^{k_i-1}$  because this must be the case at least for that  $k_i$  such that  $U_i^{k_i} < \bar{L}_i^{k_i}$ . If  $U_i^{k_i} < \bar{L}_i^{k_i}$ , since  $\bar{L}_i^{k_i} \leq L_i^{k_i-1}$  by virtue of equation (9), we also have that  $U_i^{k_i} < \bar{L}_i^{k_i-1}$ . Thus, we have shown that for all  $i$  there is a  $k_i$  such that  $U_i^{k_i} < \bar{L}_i^{k_i-1}$ . Next, we show that if it is the case that  $U_i^{k_i} \leq \bar{L}_i^{k_i-1}$ , then it is also the case that  $U_i^k \leq \bar{L}_i^{k-1}$  for all  $k > k_i$ . This follows directly from the order isomorphism property of  $f_{1,i}$  and from the fact that (by equations (16) we have  $\bar{U}_i^k = \bar{L}_i^{k-1}$ . Since this is true for all  $i$ , we can say that there is a  $\bar{k} = \max_i(k_i)$  such that for all  $k > \bar{k}$  we have that  $U_i^k \leq \bar{L}_i^{k-1}$  for all  $i$ .

(2) Let then  $\bar{k}$  be the smallest  $k$  for which  $U_i^k \leq \bar{L}_i^{k-1}$  for all  $i$ . Set  $k = \bar{k} + j$ , then we have  $\bar{L}_i^k(\bar{x}_i) = \phi^{\bar{k}+j-l_1}(L_1^0, F_i^{\bar{k}-l_1-1}(x_i, \alpha_i, u_L(\alpha_i)), \dots, F(\bar{x}_i, \alpha_i, u_L(\alpha_i)), \bar{x}_i)$ , in which  $l_1 \geq 0$ . We have that  $l_1$  can be larger than zero by virtue of equations (15). Similarly, we have that

$$\begin{aligned} U_i^k(\bar{x}_i) = \phi^{\bar{k}+j-l-1}(L_1^0, F_i^{\bar{k}-l-2}(F_i^j(\bar{x}_i, \beta_i, u_U(\beta_i)), \alpha_i, u_L(\alpha_i)), \\ \dots, F(\bar{x}_i, \beta_i, u_U(\beta_i)), \bar{x}_i), \end{aligned}$$

by virtue of the first of (16), in which  $l \leq l_1$ . If instead, the first of (16) was never verified for  $k < \bar{k}$ , we could use again Proposition 1 for showing that after some  $k > \bar{k}$  we have that  $U_i^k(\bar{x}_i) < L_i^k(\bar{x}_i)$ . By virtue of Assumption 4, the sequence  $\{U_i^{\bar{k}+j} - \bar{L}_i^{\bar{k}+j}\}_{j>0}$  is strictly decreasing. Thus, for all  $i$  there is a finite  $k_i^*$  such that  $\bar{U}_i^{k_i^*} < L_i^{k_i^*}$ . Let  $k^* = \min_i(k_i^*)$ . ■

We next show that  $\bar{E} \supseteq E$  by showing that for all  $x \notin \bar{E}$ , there is always an input such that  $x$  is mapped outside  $\bar{E}$ . We show this in two parts. First, we demonstrate that whenever  $x \notin \bar{E}$  (and thus  $(x_{1,1}, \dots, x_{1,N}) \notin \bar{E}^*(x) \subseteq \mathbb{R}^N$ ) there is a two-dimensional projection of  $\bar{E}^*(x) \subseteq \mathbb{R}^N$  and of  $(x_{1,1}, \dots, x_{1,N})$  along coordinate axis  $(i, j)$  in  $\mathbb{R}^N$ , such that  $(x_{1,i}, x_{1,j})$  is not contained in  $\bigcup_{k=0}^{k^*} [\bar{L}_i^k(\bar{x}_i), \bar{U}_i^k(\bar{x}_i)] \times [\bar{L}_j^k(\bar{x}_j), \bar{U}_j^k(\bar{x}_j)]$ . Secondly, we consider the two-dimensional projection of  $\bar{E}^*(x)$  and of  $(x_{1,1}, \dots, x_{1,N})$  to compute an input that maps the two-dimensional projection of  $(x_{1,1}, \dots, x_{1,N})$  outside the two-dimensional projection of  $\bar{E}^*(x)$ .

**Proposition 2:** If  $(x_{1,1}, \dots, x_{1,N}) \notin \bar{E}^*(x)$ , then there is a pair of coordinates  $(i, j)$  such that  $(x_{1,i}, x_{1,j}) \notin [\bar{L}_i^k(\bar{x}_i), \bar{U}_i^k(\bar{x}_i)] \times [\bar{L}_j^k(\bar{x}_j), \bar{U}_j^k(\bar{x}_j)]$  for all  $k$ .

*Proof:* We omit here the dependence of  $\bar{E}^*$ , of  $\bar{L}^k$ , and of  $\bar{U}^k$  on  $x$ . If  $(x_{1,1}, \dots, x_{1,N}) \notin \bar{E}^*$ , then it means that  $(x_{1,1}, \dots, x_{1,N})$  is not in any of the component rectangles of  $\bar{E}^*$ , that is,  $(x_{1,1}, \dots, x_{1,N}) \notin [\bar{L}_1^k, \bar{U}_1^k] \times \dots \times [\bar{L}_N^k, \bar{U}_N^k]$  for all  $k$ . Thus, for all  $k$  there is at least one  $i_k$  such that either (a)  $x_{1,i_k} < \bar{L}_{i_k}^k$  or (b)  $x_{1,i_k} > \bar{U}_{i_k}^k$ . Let  $k$  be the smallest integer less or equal to  $k^*$  such that there is a  $i_k$  with  $x_{1,i_k} > \bar{U}_{i_k}^k$ . If it does not exist, it means that there is  $i_{k^*}$  such that  $x_{1,i_{k^*}} < \bar{L}_{i_{k^*}}^{k^*}$ . Therefore, independently of the component  $i$  we will have that  $(x_{1,i}, x_{1,i_{k^*}}) \notin [\bar{L}_i^k, \bar{U}_i^k] \times [\bar{L}_{i_{k^*}}^k, \bar{U}_{i_{k^*}}^k]$  for all  $k$  because  $\bar{L}_{i_{k^*}}^{k^*} < \bar{L}_{i_{k^*}}^k$  for all  $k < k^*$  by construction. If it exists, it means that  $x_{1,i_k} > \bar{U}_{i_k}^k$  and that there is a  $i_{k-1}$  such that  $x_{1,i_{k-1}} < \bar{L}_{i_{k-1}}^{k-1}$ . As a consequence, we have that

$x_{1,ik} > \bar{U}_i^k \geq \bar{U}_i^{k+1} \geq \dots \geq \bar{U}_i^{k^*}$  and therefore  $x_{1,ik} \notin [\bar{L}_i^j, \bar{U}_i^j]$  for all  $j \in \{k, \dots, k^*\}$ . Also, we have that  $x_{1,ik-1} < \bar{L}_i^{k-1} \leq \bar{L}_i^{k-2} \leq \dots \leq \bar{L}_i^0$  and therefore  $x_{1,ik-1} \notin [\bar{L}_i^j, \bar{U}_i^j]$  for all  $j \in \{0, \dots, k-1\}$ . Thus, one can conclude that the pair of coordinates  $(x_{1,ik}, x_{1,ik-1}) \notin [\bar{L}_i^j, \bar{U}_i^j] \times [\bar{L}_i^j, \bar{U}_i^j]$  for all  $j$ . ■

**Proposition 3:** Let  $\bar{L}_i^k(\bar{x}_i)$  and  $\bar{U}_i^k(\bar{x}_i)$  be as in Algorithm 1. If  $x_{1,i} < \bar{L}_i^k(\bar{x}_i)$ ,  $(x_{1,i} > \bar{U}_i^k(\bar{x}_i))$  then there exists a continuous/discrete control law  $(\sigma_i, u_i)$  such that  $x'_{1,i} < \bar{L}_i^{k-1}(\bar{x}'_i)$  ( $x'_{1,i} > \bar{U}_i^{k-1}(\bar{x}'_i)$ ). In particular, such control laws are as follows:

if  $x_{1,i} < \bar{L}_i^k(\bar{x}_i)$ , then

$$\begin{cases} R_i(\sigma_i) = \alpha_i, u_i = u_L(\alpha_i) & \text{if } L_i^{k,a}(\bar{x}_i) < L_i^{k,b}(\bar{x}_i) \\ R_i(x_i) = q_i, u_i = u_L(q_i) & \text{if } L_i^{k,a}(\bar{x}_i) \geq L_i^{k,b}(\bar{x}_i) \end{cases} \quad (21)$$

if  $x_{1,i} > \bar{U}_i^k(\bar{x}_i)$ ,

$$\begin{cases} R_i(\sigma_i) = \beta_i, u_i = u_U(\beta_i) & \text{if } U_i^{k,a}(\bar{x}_i) > U_i^{k,b}(\bar{x}_i) \\ R_i(x_i) = q_i, u_i = u_U(q_i) & \text{if } U_i^{k,a}(\bar{x}_i) \leq U_i^{k,b}(\bar{x}_i). \end{cases} \quad (22)$$

*Proof:* In the case in which  $x_{1,i} < \bar{L}_i^k(\bar{x}_i)$  and  $L_i^{k,a}(\bar{x}_i) > L_i^{k,b}(\bar{x}_i)$ , we will have that  $x_{1,i} < L_i^{k,a}(\bar{x}_i)$ . Applying  $f_{1,i}$  both sides and taking into account that  $f_{1,i}$  preserves the ordering, we obtain that  $f_{1,i}(x_{1,i}, \dots, x_{n,i}) < f_{1,i}(L_i^{k,a}(\bar{x}_i), \bar{x}_i)$ . By equation (10) and by the order isomorphism property of  $f_{1,i}$  in its first argument, we have that  $f_{1,i}(L_i^{k,a}(\bar{x}_i), \bar{x}_i) = \bigwedge_{q_i \in \bar{Q}_i} \bar{L}_i^{k-1}(F_i(\bar{x}_i, q_i, u_L(q_i)))$ . Also, we have that  $\bigwedge_{q_i \in \bar{Q}_i} \bar{L}_i^{k-1}(F_i(\bar{x}_i, q_i, u_L(q_i))) \leq \bar{L}_i^{k-1}(F_i(\bar{x}_i, q_i, u_L(q_i)))$ . As a consequence, if we choose the control action such that  $q_i = R_i(x_i)$  and  $u_i = u_L(q_i)$ , we obtain that  $F_i(\bar{x}_i, q_i, u_L(q_i)) = (x'_{2,i}, \dots, x'_{n,i}) = \bar{x}'_i$  and therefore that  $x'_{1,i} = f_{1,i}(x_{1,i}, \dots, x_{n,i}) < \bar{L}_i^{k-1}(\bar{x}'_i)$ . If  $L_i^{k,a}(\bar{x}_i) \leq L_i^{k,b}(\bar{x}_i)$ . We can proceed similarly to obtain that  $x_{1,i} < L_i^{k,b}(\bar{x}_i)$  implies by the order preserving property of  $f_{1,i}$  that  $f_{1,i}(x_{1,i}, \dots, x_{n,i}) < f_{1,i}(L_i^{k,b}(\bar{x}_i), \bar{x}_i)$ . By equation (11), we also have that  $f_{1,i}(L_i^{k,b}(\bar{x}_i), \bar{x}_i) = \bar{L}_i^{k-1}(F_i(\bar{x}_i, \alpha_i, u_L(\alpha_i)))$ , which by choosing  $R_i(\sigma_i) = \alpha_i$  and  $u_i = u_L(\alpha_i)$  is equal to  $\bar{L}_i^{k-1}(x'_{2,i}, \dots, x'_{n,i}) = \bar{L}_i^{k-1}(\bar{x}'_i)$ . As a consequence, we have again that  $x'_{1,i} = f_{1,i}(x_{1,i}, \dots, x_{n,i}) < \bar{L}_i^{k-1}(x'_{2,i}, \dots, x'_{n,i}) = \bar{L}_i^{k-1}(\bar{x}'_i)$ . If  $x_{1,i} > \bar{U}_i^k(\bar{x}_i)$ , the proof proceeds in a similar way. ■

When the measurement  $x$  becomes available, the extremes  $\bar{L}_i^k(\bar{x}_i)$  and  $\bar{U}_i^k(\bar{x}_i)$  can be evaluated. Then, one checks whether maintaining the current input will cause that  $(x'_{1,1}, \dots, x'_{1,N})$  will enter any of the intervals  $[\bar{L}_1^k(\bar{x}'_1), \bar{U}_1^k(\bar{x}'_1)] \times \dots \times [\bar{L}_N^k(\bar{x}'_N), \bar{U}_N^k(\bar{x}'_N)]$  for all  $k$ . If not, the input is maintained constant. Otherwise, the input is changed according to the following algorithm.

**Algorithm 2.**

- (i) If there is a  $k \in [0, k^*]$  and a pair of coordinates  $(i, j)$  such that  $x_{1,i} > \bar{U}_i^{k+1}$  and  $x_{1,j} < \bar{L}_j^k$  then set  $(R_i(x_i, \sigma_i), u_i)$  as in equation (22) with  $k+1$  in place of  $k$ , and set  $(R_j(x_j, \sigma_j), u_j)$  as in equation (21);
- (ii) If instead  $(x_{1,1}, \dots, x_{1,N}) < (L_1^{k^*}(\bar{x}_1), \dots, L_N^{k^*}(\bar{x}_N))$ , select  $(i, j)$  such that  $x_{1,i} < \bar{L}_i^{k^*}(\bar{x}_i)$  and  $x_{1,j} < \bar{L}_j^{k^*}(\bar{x}_j)$  with  $U_j^{k^*+1}(\bar{x}_j) < L_j^{k^*+1}(\bar{x}_j)$ . If  $x_{1,j} > L_j^{k^*+1}(\bar{x}_j)$  then  $(x_{1,j} > U_j^{k^*+1}(\bar{x}_j))$  set  $(R_j(x_j, \sigma_j), u_j)$  as in equation (22) and set  $(R_i(x_i, \sigma_i), u_i)$  as in equation (21). If  $x_{1,j} \leq L_j^{k^*+1}(\bar{x}_j)$ , set

$(R_j(x_j, \sigma_j), u_j)$  as in equation (21) and set  $(R_i(x_i, \sigma_i), u_i)$  arbitrarily (if  $R_i(x_i, \sigma_i) = R_i(x_i)$ , then set  $u_i \in u_i(q_i)$ ).

**Theorem 3:** (Correctness) There exists a continuous control law  $u = g_C(x, q)$  and a switching law  $\sigma = g_D(x)$  with  $q = R(x, \sigma)$  such that if  $x \notin \bar{E}$  with  $\bar{E} = \{(x_{1,1}, \dots, x_{n,1}, \dots, x_{1,N}, \dots, x_{n,N}) \mid (x_{1,1}, \dots, x_{1,N}) \in \bar{E}^*(x)\}$  and  $\bar{E}^*(x)$  as computed by Algorithm 1, then  $x' \notin \bar{E}$ . In particular, Algorithm 2 provides one such control law.

The proof of this theorem is a direct consequence of Proposition 2 and of Proposition 3.

V. EXAMPLES

**Example 1: Vehicles at a traffic intersection.** Let us consider two vehicles converging to a traffic intersection (represented in Figure 1). The vehicle's physical motion can

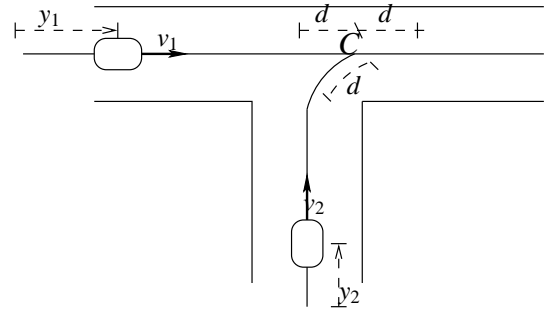


Fig. 1. Two vehicles converging at a traffic intersection. The bad set is defined to be the set of all vehicle 1/vehicle 2 configurations in which the vehicles are both closer than some distance  $d$  from the intersection  $C$  of their paths.

be modeled by considering its longitudinal dynamics along its geometric path (determined by the geometry of the lanes) following a similar modeling framework as performed in [3]. Let then  $y_1$  and  $y_2$  denote the position of the two vehicles along their path with respect to some fixed reference point. Let  $v_1$  and  $v_2$  be the velocities of the two cars along their paths. We assume that each vehicle dynamics along its path can be modeled by a second order system, which in discrete time takes the form:

$$y'_i = y_i + v_i(\Delta T), v'_i = v_i + u_i(\Delta T), i \in \{1, 2\}, \quad (23)$$

in which  $\Delta T$  is the time interval. The controller  $u_i$  can directly affect the acceleration by acting on the throttle pedal or on the brake. When a vehicle is inside the intersection, it cannot stop as it has to free the intersection as soon as possible, while it can stop before entering the intersection. In addition, a vehicle cannot move backwards in its lane. These constraints can be modeled by requiring that (for a suitable  $y_i^A$ ) for  $y_i \leq y_i^A$  then  $v_i \geq 0$ , while for  $y_i > y_i^A$  we must have  $v_i \geq v_m$  with  $v_m > 0$ . Let  $u_m < 0 < u_M$ . Thus, each vehicle can be described by a hybrid automaton with two modes:  $q_i = q_{1,i}$  if  $(y_i \leq y_i^A$  and  $v_i \leq 0)$  or  $(y_i > y_i^A$  and  $v_i \leq v_m)$ ;  $q_i = q_{2,i}$  if  $(y_i \leq y_i^A$  and  $v_i > 0)$  or  $(y_i > y_i^A$  and  $v_i > v_m)$ . In each one of these modes, the update map  $f$  is given by equations (23), in which  $u(q_{1,i}) = [0, u_M]$ ,  $u(q_{2,i}) = [u_m, u_M]$ .

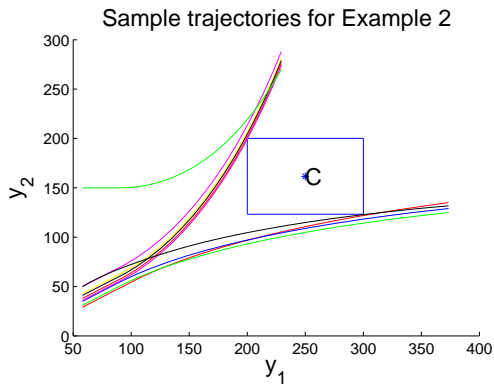


Fig. 2. The rectangle represents the set  $\bar{B}$ . The plot shows the  $y_1, y_2$  trajectories of trains at a railway merging. Each trajectory corresponds to a different choice of initial values for  $(y_1, y_2, v_1, v_2, u_1, u_2)$ .

Since  $\mathcal{I}_D = \emptyset$ , the hybrid automaton admits only autonomous mode transitions.

**Example 2: Trains at a railway merging.** Consider two trains in the proximity of a railway merging. Assuming a second order dynamics along their rail, each train can be modeled again as in equations (23). However, now the input sets will be different from the previous example. In digital control mode [16], the input  $u_i$  can take four values corresponding to a “hard-brake” mode, a “run-out” mode, a “constant-speed” mode, and an “acceleration” mode. Let these 4 values be denoted by respectively  $\alpha, \gamma, \delta, \beta$  so that  $\alpha < \gamma < \delta < \beta$ . Each vehicle dynamics can thus be modeled by a hybrid automaton with four modes such that  $q_i = q_{1,i}$  iff  $u_i = \alpha$ ,  $q_i = q_{2,i}$  iff  $u_i = \gamma$ ,  $q_i = q_{3,i}$  iff  $u_i = \delta$ , and  $q_i = q_{4,i}$  iff  $u_i = \beta$ . There are no autonomous switches in this system, so that for each train  $R_i(y_i, v_i, \sigma_i) = R_i(\sigma_i)$  where  $\sigma_i$  is the discrete input.

One can verify that the above models for each vehicle are triangular order preserving hybrid automata. The safety requirement is that the two vehicles never are in a ball of radius  $d$  around the conflict point  $C$  at the same time. This is encoded by a bad set  $B = \{(y_1, v_1, y_2, v_2) \mid (y_1, y_2) \in \bar{B}\}$ , in which  $\bar{B} = [L_1, U_1] \times [L_2, U_2]$  for suitable  $L_1, U_1, L_2, U_2 \in \mathbb{R}$ . Algorithms 1 and 2 were implemented for the examples introduced for the case  $\Delta T = 1$ . The results are shown in Figure 2 and in Figure 3. The over-approximation of the escape set is tight as the trajectories of the controlled system are very close to the bad set  $\bar{B}$  (Figure 2). This means that the control law is not conservative. An instance of escape set for fixed values of the velocities is depicted in Figure 3.

## VI. CONCLUSIONS

We have presented a linear complexity algorithm for the computation of safety controllers for which a termination condition is provided. This result is obtained by directly exploiting the triangular structure of the system and the order preserving property of the dynamics. Simulation results have shown that the proposed control law is not conservative. In our future work, we will relax the non-decreasing assumption of the update functions, we will consider bad sets that are the union of intervals as opposed to an interval itself and that

Escape set for fixed  $v_1$  and  $v_2$  for Example 1

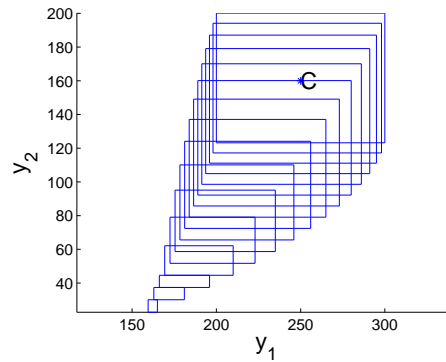


Fig. 3. The escape set for a combination of speeds for Example 1.

incorporate all the state variables. We will consider discrete state update maps with memory and we will use interval abstraction techniques to embrace class of systems that are not order preserving. Finally, the case of imperfect and partial observation will be addressed and the extension to games considered.

## REFERENCES

- [1] O. of Safety Analysis, “Accidents/incidents counts,” *Federal Railroad Administration*, <http://safetydata.fra.dot.gov/officeofsafety>, 2005.
- [2] K. Laberteaux, L. Caminiti, D. Caveney, and H. Hada, “Pervasive vehicular networks for safety,” *IEEE Pervasive Computing, Spotlight*, pp. 60–62, 2006.
- [3] C. J. Tomlin, J. Lygeros, and S. Sastry, “A game theoretic approach to controller design for hybrid systems,” *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–970, 2000.
- [4] C. Tomlin, I. Mitchell, and R. Ghosh, “Safety verification of conflict resolution maneuvers,” *IEEE Trans. Intelligent Transportation*, vol. 2, pp. 110–120, 2001.
- [5] T. A. Henzinger and P. W. Kopke, “Discrete-time control for rectangular hybrid automata,” *Theoretical Computer Science*, vol. 221, pp. 369–392, 1999.
- [6] O. Shakhmura, G. J. Pappas, and S. Sastry, “Semi-decidable synthesis for triangular hybrid systems,” in *Lecture Notes in Computer Science, volume 2034*, 2001.
- [7] E. Asarin, O. Maler, and A. Pnueli, “Symbolic controller synthesis for discrete and timed systems,” in *Lecture Notes in Computer Science, volume 999*, 1995, pp. 1–20.
- [8] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, “What’s decidable about hybrid automata,” *Journal of Computer and System Sciences*, vol. 57, pp. 94–124, 1998.
- [9] I. Mitchell and C. J. Tomlin, “Overapproximating reachable sets by hamilton-jacobi projections,” *Journal of Scientific Computation*, vol. 19, no. 1, pp. 323–346, 2003.
- [10] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, “Hytech: A model checker for hybrid systems,” *Software Tools for Technology Transfer*, vol. 1, pp. 110–122, 1997.
- [11] T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi, “Beyond hytech: Hybrid systems analysis using interval numerical methods,” in *Lecture Notes in Computer Science 1790*, Springer-Verlag, 2000, pp. 130–144.
- [12] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*. Cambridge University Press, 2002.
- [13] P. Cousot, “Semantic foundations of program analysis: Theory and applications,” In *S. S. Muchnick and N. D. Jones (Eds.), Program Flow Analysis: Theory and Applications*. Prentice-Hall, pp. 303–345, 1981.
- [14] H. L. Smith, *Monotone Dynamical Systems*. American Mathematical Society, 1995.
- [15] T. A. Henzinger, “The theory of hybrid automata,” in *Proceedings of the 11th Annual Symposium on Logic in Computer Science*. IEEE press, 1996, pp. 278–292.
- [16] J. Pachl, *Railway operation and control*. VTD Rail Publishing, 2002.