15th Triennial World Congress, Barcelona, Spain

TARGET CONTROL FOR HYBRID SYSTEMS

Anastasios Trontis * **Hao Xia** * **Yan Pang** *
Michael P. Spathopoulos *

* *Department of Mechanical Engineering, University of Strathclyde, Glasgow, UK*
{*atrontis,hxia,ypang,mps*}*@mecheng.strath.ac.uk*

Abstract: This paper is concerned with the supervisory target control problem for hybrid systems modeled by hybrid automata. The problem is studied in a straightforward manner through reachability analysis. A switching controller is proposed such that all the trajectories of the controlled automaton, that initiate from a given initial set in the state space, reach a target set. At the same time a cost function, specified by weighting the discrete-event transitions of the automaton, is minimized. Emphasis is given to the control synthesis under event uncertainty modeled by uncontrollable transitions. *Copyright © 2002 IFAC*

Keywords: hybrid systems, target control, reachabilty, discrete event systems, supervisory control

1. INTRODUCTION

Hybrid dynamic systems arise when time-driven dynamics interacts with events. Such interaction is typical in complex systems that consist of a number of interconnected subsystems or operation modes. In a supervisory control framework transitions between modes are ordered by the controller (supervisor). In each mode the controller observes the continuous state of the system and detects certain events. After an event has been detected, the controller switches the system to the next mode. Our aim is to set the conditions that, when satisfied, the appropriate event is detected. Then the controller, utilizing this information, orders the switching such that the overall behavior of the system meets the specification.

Efficient modeling of hybrid systems requires a blending of continuous dynamics models (differential or difference equations) with discrete event models (automata, Petri-nets etc.). In this work hybrid systems are modeled by hybrid automata and the target control problem is cast as follows: *Given a hybrid automaton, an initial and a target set in the hybrid state space, do there exist switching conditions that "transfer" the system from the initial to the target set, while a performance criterion is minimized?*

In the hybrid systems community the target control problem has always been considered as the dual of the "safety" synthesis problem: *Given a set of forbidden states F, do there exist switching conditions that ensure that the system never reaches F?* Although a great amount of research work has been published on synthesizing controllers for safety specifications, we highlight the works of (Asarin *et al.*, 2000; Tomlin *et al.*, 1999; Wong-Toi, 1997), this is not the case for the target control problem which has attracted considerably little attention. However, the target control problem has its own unique features and the synthesis procedure cannot be seen simply as the dual of safety. The main result of this study is the algorithm that derives the switching conditions needed to drive the system from the initial to the target set at a minimum discrete transition cost. Also, emphasis is given to the control synthesis under uncontrollable transitions, which is an issue that previous works (e.g Asarin *et al.*, 2001; Tittus and Egardt, 1998; Trontis and Spathopoulos, 2001*a*; Trontis and Spathopoulos, 2001*b*) either do not cover or fail to present clearly.

The paper is organized as follows: Section 2 introduces the modelling framework and the control objective. Section 3 is concerned with the solution for hybrid automata with no uncontrollable transitions. Control synthesis for hybrid automata with uncontrollable transitions is studied in section 4. Finally, conclusions and directions for further research are given in section 5.

2. PROBLEM FORMULATION

2.1 *Hybrid automata*

The hybrid automaton is the basic entity of our analysis.

Definition 1. A **hybrid automaton** is a tuple $A = (X, Q, INV, \mathbf{f}, T, c, G)$ where:

- $X \subseteq \mathbb{R}^n$ is the continuous state space defining the continuous part of the plant state space.
- Q is the finite set of discrete control locations or control modes. The pair $Q \times X$ defines the hybrid state space of the system.
- The function $INV : Q \to 2^X$ assign to each control location $q \in Q$ an invariant $INV(q) \subseteq X$.
- $\mathbf{f} : Q \times X \to X$ is a vector field that assigns to each control location the dynamics of the continuous part of the plant.
- $T = T_u \cup T_c \subseteq Q \times Q$ is the set of discrete transitions between control locations. T_c is the set of controllable transitions while T_u is the set of uncontrollable transitions.
- $c : T \to \mathbb{R}^+$ is a transition cost function that assigns a positive real number to each transition.
- $G : T \to 2^X$ assigns to $(q, q') \in T$ a guard set $G(q, q')$ such that $G(q, q') \cap INV(q) \neq \emptyset$.

All the sets involved in the above definition are considered closed and compact. Without loss of generality it is assumed that there are no discontinuities in the state $\mathbf{x}$ of the system during transitions. The set $INV(q)$ is the subset of the continuous state space in which the continuous state must be contained as long as the system resides in location q. The vector field $\mathbf{f}$ is assumed to be globally Lipschitz in X. The guard set $G(q, q')$ is the subset of the state space where the system can switch from location q to q'. A transition can take place as long as the corresponding guard condition is satisfied. In the case of controllable transitions the moment that the transition takes place is a design variable. An external system (controller) orders the transition when a certain condition, subject to design, is satisfied. If there is no controller imposed on the automaton the transition can take place anytime while $\mathbf{x} \in G(q, q')$. In the case of uncontrollable transitions, the switching between two locations q and q' can occur anytime the guard $G(q, q')$ is satisfied and there is no control over them. An uncontrollable transition is forced when the continuous state exits the corresponding guard. That is, unlike the model of (Dang, 2000; Wong-Toi, 1997), the system will definitely take the uncontrollable transition if the state enters its guard. Finally, both controllable and uncontrollable transitions are forced when the continuous state, while it satisfies the guard condition, reaches the boundary of the invariant.

Definition 2. A **hybrid time trajectory** τ is defined as a finite or infinite sequence of intervals of the non-negative real line, $\tau = \{I_i\}$, $i \in \mathbb{N} \setminus \{0\}$, which satisfy the following properties:

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

The set of all hybrid time trajectories is denoted by $\mathbb{T}$. A trajectory of a hybrid automaton is defined as follows:

Definition 3. A **trajectory of a hybrid automaton** A, initiating from a state $(q_0, \mathbf{x_0}) \in Q \times X$ is a collection $(\tau, q, \mathbf{x})$ with $\tau \in \mathbb{T}$, $q : \tau \to Q$ and $\mathbf{x} : \tau \to X$ which satisfies:

- Initial condition: $(q_0, \mathbf{x_0}) = (q(\tau_1), \mathbf{x}(\tau_1))$, $\mathbf{x}(\tau_1) \in INV(q_0)$.
- Discrete transition: for all i there exists a τ'_i and a q' with $(q, q') \in T$ such that $q(\tau'_i) = q$ and $\mathbf{x}(\tau'_i) \in G(q, q')$. If $t_e = \sup\{t \,|\, \mathbf{x}(t) \in G(q, q')\}$ then $q(\tau_{i+1}) = q'$, $\tau_{i+1} \leq t_e$.
- Continuous transition: for all $t \in [\tau_i, \tau'_i]$: $q(t) = q \in Q$ and $\dot{\mathbf{x}}(t) = \mathbf{f}_q(\mathbf{x}(t))$, $\mathbf{x}(t) \in INV(q)$.

The set of hybrid trajectories initiating from $(q, \mathbf{x})$ is denoted by $L(A, (q, \mathbf{x}))$, where $L(A)$ denotes the set of all hybrid trajectories generated by A. $L(A)$ is called the *language* of A. From the above definition it is inferred that there is more than one possible trajectory initiating from $(q_0, \mathbf{x_0})$. However, uniqueness of solutions is not a prerequisite for the control synthesis problem.

Nonetheless, some abnormalities may occur under these definitions. For instance, a trajectory $\xi \in L(A)$ may reach the boundary of $INV(q)$ without entering any guard set $G(q, q')$. Such trajectory becomes *blocked* and the automaton is called *blocking*. Furthermore, a trajectory can switch infinitely often between discrete locations in finite time. This phenomenon may occur when there exists a sequence of discrete states $q_1, \ldots, q_s$ such that $G(q_1, q_2) \cap G(q_2, q_3) \cap \cdots \cap G(q_s, q_1) \neq \emptyset$. For more details see (Asarin *et al.*, 2000). An automaton that accepts this trajectory is called *Zeno*. These issues must be taken into consideration during the control design process. One requires a controller that prevents blocking and does not allow finitely many transitions in finite time.

2.2 *Control scheme and specification*

Given the plant modeled by a hybrid automaton and a specification for the desired behavior, the objective is to derive a controller that guarantees the evolution of the system dynamics according to the specification. Assuming full observability of the hybrid state, the controller is considered to be a map:

$$C : L(A, (q_0, \mathbf{x_0})) \rightarrow 2^Q \tag{1}$$

The controller evaluates the current state against pre-specified conditions and makes the decision over ordering a transition or not. These conditions are expressed through the *control sequence* $l_i = (\pi_i, G_i^*(\pi_i, \pi_{i+1}))$. π_i denotes a sequence (path) of control locations $q \in Q$, while G_i^* is the corresponding sequence of control guards. The control guards $G_i^*(\cdot,\cdot) \subseteq G(\cdot,\cdot)$ are the design parameters of the controller. Having loaded the sequence l_i in the controller's memory, the control strategy is applied according to the scheme:

$$C(\xi(q_0, \mathbf{x_0})) = \begin{cases} \{q, q'\} & \begin{array}{l}\text{if } \mathbf{x} \in G_i^*(q,q') \wedge \\ (q,q') \in T_c\end{array} \\ q' & \begin{array}{l}\text{if } \mathbf{x}(t_e^-) \in G_i^*(q,q') \wedge \\ \mathbf{x}(t_e^+) \notin G_i^*(q,q') \wedge \\ (q,q') \in T_c\end{array} \\ \emptyset & \begin{array}{l}\text{if } \mathbf{x} \in G_i^*(q,q') \wedge \\ (q,q') \in T_u\end{array} \\ q & \text{otherwise} \end{cases} \tag{2}$$

with $q = \pi_i$ and $q' = \pi_{i+1}$. Clearly, the proposed controller is *non-deterministic*. If the first condition is satisfied, the controller can immediately order the transition to the next location q' on π or "idle" for some time and order it later. In either case the idling period stops when the second condition is satisfied. The controller must order the transition just before the expression $\mathbf{x} \in G_i^*(q,q')$ becomes *false* at time $t_e = \sup\{t \,|\, \mathbf{x}(t) \in G_i^*(q,q')\}$. The idling period reflects design margins and is directly connected to the fact that the guard conditions $G(q,q')$ are in general full dimensional sets rather than switching hyper-surfaces. In our framework no restrictions over the duration of idling are considered. Naturally, when the continuous state satisfies the guard condition $G_i^*(q,q')$ of an uncontrollable transition $(q,q') \in T_u$ the controller can take no action. Last but not least, there may be the case where $\mathbf{x} \in G(q,q'')$ with $q = \pi_i$ and $q'' \neq \pi_{i+1}$. In this case the controller prevents the transition (q,q'') since $q'' \neq \pi_{i+1}$. If this transition is uncontrollable, then the design must guarantee that the guard $G(q,q'')$ is not reachable. In other words, the control scheme (2) restricts the evolution of the system along a single path π.

Let us now consider a hybrid automaton A, an *initial* set $I = (q_0, X_0)$, a *target* set $F = (q_F, X_F)$ and the path $\pi = q_0, \ldots, q_F$. The *cost* of π is defined as :

$$J(\pi) = \sum_{i=1}^{M} c\,(\pi_i, \pi_{i+1}) \tag{3}$$

with $M \in \mathbb{N}$. This function represents the transition cost along π from an initial state $(q(\tau_1), \mathbf{x}(\tau_1)) \in I$ to a final $(q(t_f), \mathbf{x}(t_f)) \in F$, with $t_f \geq \tau_{M+1}$. Note that by applying the control scheme (2) all the trajectories $\xi \in L(A, (q(\tau_1), \mathbf{x}(\tau_1)))$ evolve exclusively on π and therefore the cost function (3) is well defined even when uncontrollable transitions are considered. Finally, the target control problem is cast as follows:

Given a hybrid automaton A, an initial set $I = (q_0, X_0)$ and a target set $F = (q_F, X_F)$, design the control sequence l_i such that all trajectories initiating from I reach F with the least overall transition cost.

From the previous, two early conclusions are derived. First, if there exists a solution to the target control problem, then all the trajectories of the closed system are non-Zeno. Indeed, since we require the number of switches $M \in \mathbb{N}$, a solution that involves infinite number of switches does not satisfy the specification. Second, given that there exists a solution, all trajectories of the closed system initiating from I and reaching F are non-blocking, under the assumption that their behavior is not considered after they have reached the target set.

3. CONTROL SYNTHESIS WITH NO UNCONTROLLABLE TRANSITIONS

As it was shown in the previous section, the solution to the synthesis problem is the control sequence l_i. To obtain l_i, backward reachability analysis is applied on separate paths π between the initial and the target location q_0 and q_F, in ascending order of cost. Starting from the target set F, one asks to compute the set of states W_π from which F is reachable by the evolution of the continuous and discrete dynamics along π. This computation is iterated untill the initial location q_0 is reached. Using the information provided by the reachability analysis, one derives the solution, if any, to the synthesis problem. To perform reachability analysis, the following operators are employed:

Definition 4. Given a hybrid automaton A and a set $Y \subseteq 2^{Q\times X}$, the **continuous predecessor operator** $pre_c : 2^{Q\times X} \to 2^{Q\times X}$ is defined as:

$$\begin{aligned} pre_c(Y) = \{(q,\mathbf{x}) \mid \exists t \in \mathbb{R}^+ :& \\ \mathbf{x}' = \mathbf{x} + \int_0^t \mathbf{f}_q(\tau)\,\mathrm{d}\tau \wedge (q,\mathbf{x}'(t)) \in Y \wedge& \\ \mathbf{x}'(\tau) \in INV(q)\,\forall \tau \in [0,t]\}& \end{aligned} \tag{4}$$

Definition 5. Given a hybrid automaton A, a set $Y \subseteq 2^{Q\times X}$ and a path $\pi = q_0, \cdots, q_F$, the **discrete predecessor operator** $pre_d : 2^{Q\times X} \times \mathbb{N} \to 2^{Q\times X}$ is defined as:

$$\begin{aligned} pre_d^i(Y) = \{(q,\mathbf{x}) \mid \exists q' \in Q : (q',\mathbf{x}) \in Y \wedge \\ q' = \pi_i \wedge q = \pi_{i-1} \wedge \mathbf{x} \in INV(q) \cap G(q,q')\} \end{aligned} \tag{5}$$

Intuitively, the continuous predecessor operator defines all the states that can reach a set Y by the evolution of the continuous dynamics only. Similarly, the discrete predecessor operator defines all the states that can reach a set Y by a discrete transition from the previous location q to the current location q' on π. With these definitions in place, the solution to the target control problem is obtained from algorithm 1.

Algorithm 1: Target control synthesis with no uncontrollable transitions

> $\{\pi^1, \cdots, \pi^K\} := K\_shortest\_paths(A)$
> **repeat** $j = 1, 2, \ldots$
> $\quad m^j := length(\pi^j) \quad \%\pi^j = q_0, \cdots, q_F$
> $\quad W_{\pi^j_{m^j}} := pre_c(F)$
> $\quad$ **for** $i = m^j - 1$ **downto** 1
> $\quad\quad W_{\pi^j_i} := pre_c(pre_d^{i+1}(W_{\pi^j_{i+1}}))$
> $\quad$ **end**
> $\quad I^j := W_{\pi^j_i} \cap I, \quad I := I \setminus I^j$
> **until** $I = \emptyset$ **or** $j = K$
> **if** $I = \emptyset$ **then** $\exists$ *feasible solution*
> $I := (q_0, X_0)$

The algorithm begins with ranking the K shortest paths from the initial location q_0 to the target location q_F in ascending order of cost. For that reason it utilizes a generalization of Dijkstra's shortest path algorithm on weighted graphs (e.g. Martins *et al.*, 2000; Martins *et al.*, 1998). Then it performs backward reachability analysis along the first identified path π^1 from the Dijkstra's algorithm. Namely, it computes the set of states W_{π^1} along π^1, from which the target set F is reachable. When the initial location q_0 is reached, it generates the set $I^1 := W_{\pi^1_1} \cap I$. This set represents all the states $(q_0, \mathbf{x_0}) \in I$ that can reach F following the optimal path π^1. If $I \setminus I^1 \neq \emptyset$ then the states $(q_0, \mathbf{x_0}) \in I \setminus I^1$ either need another path to reach F or, in the worst case, cannot reach F at all. In this case the same procedure is repeated on the next identified path π^2, considering initial set I^1. The algorithm terminates when either there are no more initial states, for which a path has not been found ($I = \emptyset$), or the K^{th} path has been tested ($j = K$). After the termination, the initial set is re-assigned the original value $I = (q_0, X_0)$.

From a theoretical point of view, if the algorithm terminates on the second condition ($j = K$) it cannot be claimed that there is no feasible solution to the problem, unless the maximum number of paths in the automaton equals K. There may be the case where a feasible path with rank greater than K does exist but the K-shortest-paths algorithm fails to identify. This indicates that the target control problem is semi-decidable, i.e. if the problem is feasible algorithm 1 may provide the solution. If not then in general it gives no answer.

Proposition 1. If algorithm 1 terminates on the first condition then the target control problem is feasible and the control sequence for $(q_0, \mathbf{x_0}) \in I^j \subseteq I = (q_0, X_0)$ is defined as:

$$l_i^j = (\pi_i^j, G_i^*(\pi_i^j, \pi_{i+1}^j)) \tag{6}$$

with $G_i^*(\pi_i^j, \pi_{i+1}^j) = proj_X(W_{\pi_i^j}) \cap G(\pi_i^j, \pi_{i+1}^j)$, while for the transition cost it holds:

$$J^*(\pi^j) = \sum_{i=1}^{m^j-1} c\left(\pi_i^j, \pi_{i+1}^j\right) \tag{7}$$

The operator $proj_X(\cdot)$ projects a set of the hybrid state space $Q \times X$ to the continuous state space X. The proof of the proposition is straightforward. Note that in general the control sequence l_i is not unique for all $(q_0, \mathbf{x_0}) \in I$. Indeed, during the execution of the algorithm the initial set I is split up into the disjoint sets I^j. States that belong to different I^js follow different paths to the target. As a result, the control sequence that must be loaded to the controller depends on the initial state $(q_0, \mathbf{x_0}) \in I$.

4. CONTROL SYNTHESIS WITH UNCONTROLLABLE TRANSITIONS

When uncontrollable transitions are introduced into the model the control design needs to be modified. The reason behind is that the controller has no control over them. It is the system itself or the environment that decide when to make a transition and to which location, no matter if the system is driven off target or gets blocked. In order to prevent these undesirable situations, the existence of uncontrollable transitions has to be taken into account through the controllable transitions of the automaton.

More specifically, let us apply algorithm 1 to an automaton A with uncontrollable transitions to obtain the control sequence $l_i = (\pi_i, G_i^*(\pi_i, \pi_{i+1}))$.

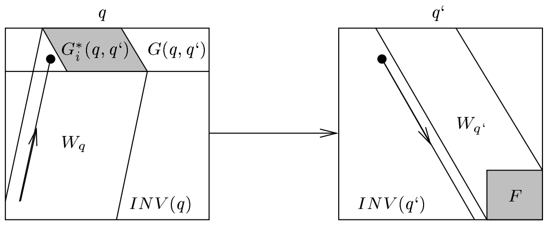

Fig. 1. Blocking induced by the uncontrollable transition (q, q')

Also let us assume that the path π contains an uncontrollable transition (q, q') with $q = \pi_i$ and $q' = \pi_{i+1}$. Necessary condition for the transition (q, q') to drive the system to the target is to occur when $\mathbf{x} \in G_i^*(q, q')$. Recall now that for every control guard $G_i^*(\cdot, \cdot)$ it holds $G_i^*(\cdot, \cdot) \subseteq G(\cdot, \cdot)$. This inclusion implies that for the continuous state, as it evolves along π, it may hold $\mathbf{x} \in G(q, q') \setminus G_i^*(q, q')$. Given that the transition (q, q') is uncontrollable, it may occur anytime the state lies in $G(q, q')$. So it may happen at $\mathbf{x} \notin G_i^*(q, q')$. Consequently, the system never reaches F and probably, as shown in figure 1, it gets blocked.

Uncontrollable transitions that are not contained in π can cause undesired behaviors as well. Let us consider the uncontrollable transition (q, q'') with $q \in \pi$ and $q'' \notin \pi$. As it is shown in figure 2, the continuous state, as it evolves towards the control guard $G_i^*(q, q')$, may reach first the guard $G(q, q'')$. But since (q, q'') is uncontrollable, it cannot be prevented and as a result the system is driven again off target.

It is clear that there exist certain states which must not be reached at any time during the evolution along a path π towards the target set F. Let us denote this set of states in each control location π_i by $\bar{F}_{\pi_i}$. From the previous, states $(q, \mathbf{x}) \in \bar{F}_{\pi_i}$ drive the system off target and may cause blocking. Therefore the set of forbidden states is defined as $\bar{F}_{\pi_i} = \bar{F}^{on}_{\pi_i} \cup \bar{F}^{off}_{\pi_i}$ where:

$$\bar{F}^{on}_{\pi_i} = \{(q, G(q, q')) \setminus pre_d^{i+1}(W_{q'}) \mid (q, q') \in T_u \wedge q = \pi_i \wedge q' = \pi_{i+1}\} \tag{8}$$

and

$$\bar{F}^{off}_{\pi_i} = \{(q, G(q, q'')) \mid (q, q'') \in T_u \wedge q = \pi_i \wedge q'' \neq \pi_{i+1}\} \tag{9}$$

The sets $\bar{F}^{on}_{\pi_i}$ contain the forbidden states related to uncontrollable transitions on the current path π, while $\bar{F}^{off}_{\pi_i}$ represent the forbidden states related to uncontrollable transitions that lead out of the path. Having defined the set $\bar{F}_\pi = \bigcup_i \bar{F}_{\pi_i}$, the next step is to compute the set $W'_\pi \subseteq W_\pi$ from which all the trajectories reach the target F avoiding at the same time $\bar{F}_\pi$. This can be seen as a *safety* synthesis problem along π considering $\bar{F}_\pi$ as the set of forbidden states. The computation of W'_π requires the definition of the *reach-avoid* operator:

Definition 6. Given a hybrid automaton A and the sets Y, $Z \subseteq 2^{Q \times X}$, the **reach-avoid operator** $reach : 2^{Q \times X} \times 2^{Q \times X} \to 2^{Q \times X}$ is defined as:

$$\begin{aligned} reach(Y, Z) = \{(q, \mathbf{x}) \mid \exists t \in \mathbb{R}^+ : \\ \mathbf{x}' = \mathbf{x} + \int_0^t \mathbf{f}_q(\tau)\, \mathrm{d}\tau \wedge (q, \mathbf{x}'(t)) \in Y \wedge \\ \mathbf{x}'(\tau) \in INV(q) \setminus Z \,\forall \tau \in [0, t]\} \end{aligned} \tag{10}$$

The reach-avoid operator defines all the states that can reach a set Y by the evolution of the continuous dynamics only, avoiding a forbidden states set Z. Incorporating this safety synthesis procedure in algorithm 1, one obtains algorithm 2 for target control synthesis with uncontrollable transitions.

Algorithm 2: Target control synthesis with uncontrollable transitions

$\{\pi^1, \cdots, \pi^K\} := K\_shortest\_paths(A)$
repeat $j = 1, 2, \ldots$
  $m^j := length(\pi^j) \quad \% \pi^j = q_0, \cdots, q_F$
  $W'_{\pi^j_{m^j}} := reach(F, \bar{F}^{off}_{\pi^j_{m^j}})$
  for $i = m^j - 1$ **downto** 1
    if $(\pi^j_i, \pi^j_{i+1}) \in T_c$
      $W'_{\pi^j_i} := reach(pre_d^{i+1}(W'_{\pi^j_{i+1}}), \bar{F}^{off}_{\pi^j_i})$
    else
      $W'_{\pi^j_i} := pre_c(pre_d^{i+1}(W'_{\pi^j_{i+1}})) \setminus pre_c(\bar{F}^{on}_{\pi^j_i} \cup \bar{F}^{off}_{\pi^j_i})$
    end
  end
  $I^j := W'_{\pi^j_i} \cap I, \quad I := I \setminus I^j$
until $I = \emptyset$ **or** $j = K$
if $I = \emptyset$ **then** $\exists$ *feasible solution*
$I := (q_0, X_0)$

Like algorithm 1, algorithm 2 starts by ranking out the K shortest paths from the initial to the target location. Then it performs reachability analysis along the first identified path π^1. The safety synthesis procedure takes place at each location of π^1 seperately. The algorithm has to check if the transition from the current location $q = \pi^1_i$ to the next location $q' = \pi^1_{i+1}$ is controllable or not. This information is vital for the computation of the safe set $W'_{\pi^1_i}$. The reason behind is that the controller can take no forcing or preventing action on uncontrollable transitions. Therefore the computation of the safe set in the involved locations needs to be more conservative. Hence, if the transition (π^1_i, π^1_{i+1}) is controllable, the safe set $W'_{\pi^1_i}$ is computed directly using the reach–avoid operator with respect to the set of forbidden states $\bar{F}^{off}_{\pi^1_i}$ that can drive the system out of the path. In the case that (π^1_i, π^1_{i+1}) is uncontrollable, the algorithm computes all the states that reach the local target set $pre_d(W'_{\pi^1_{i+1}})$

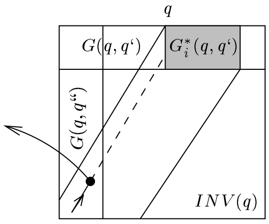

Fig. 2. The system is driven out of the path due to the uncontrollable transition (q, q'')

and then from this set it removes all the states that reach either $\bar{F}^{on}_{\pi^1_i}$ or $\bar{F}^{off}_{\pi^1_i}$ to obtain the safe set $W'_{\pi^1_i}$. After the safe set W'_{π^1} along π^1 has been computed, the algorithm carries on exactly the same way as algorithm 1. If the set W'_{π^1} does not contain the whole initial set $I = (q_0, X_0)$, the same procedure is repeated on the next identified path π^2. Again, the algorithm terminates when either there are no more initial states, for which a path has not been found ($I = \emptyset$), or the K^{th} has been tested ($j = K$). Obviously, the discussion of the previous section over semi-decidability applies in this case as well.

Proposition 2. If algorithm 2 terminates on the first condition then the target control problem is feasible and the control sequence for $(q_0, \mathbf{x_0}) \in I^j \subseteq I = (q_0, X_0)$ is defined as:

$$l_i^j = (\pi_i^j, G_i^*(\pi_i^j, \pi_{i+1}^j)) \qquad (11)$$

with $G_i^*(\pi_i^j, \pi_{i+1}^j) = proj_X(W'_{\pi_i^j}) \cap G(\pi_i^j, \pi_{i+1}^j)$, while for the transition cost it holds:

$$J^*(\pi^j) = \sum_{i=1}^{m^j-1} c\left(\pi_i^j, \pi_{i+1}^j\right) \qquad (12)$$

Intuitively speaking, algorithm 2 "shrinks" the reachable set W_π along a path π such that, for all trajectories in W'_π there is no possibility to be driven off target due to an uncontrollable transition. Therefore, the new control guards have to be defined with respect to the "shrunk" reachable set W'_π.

5. CONCLUSION

A solution to the target control problem on hybrid automata has been presented. Based on reachability analysis, algorithm 1 was derived which provides the necessary information for the design of the control strategy. Also, our study was extended to automata with uncontrollable transitions. Uncontrollability was employed to model uncertainty stemming either from the environment or the system itself. Algorithm 2 was the result of this extension.

However, the implementation of the algorithms was not among the objectives of this paper. Computational issues are not addressed at all. Besides, our aim was to present a neat solution to the supervisory target control problem, emphasizing on automata with uncontrollable transitions, and to establish target control as a stand-alone problem rather than the dual of safety. Of course the implementation of the algorithms and mainly the computation of reachable sets is an open problem for further research. Moreover, optimal target control, with respect to both continuous and discrete dynamics, is another research direction that should be investigated in a future work.

ACKNOWLEDGEMENT. Research supported by ESPRC, grant: GR/M51260

6. REFERENCES

Asarin, E., O. Bournez, T. Dang, O. Maler and A. Pnueli (2000). Effective synthesis of switching controllers for linear systems. *IEEE Proceedings* (88), 1011–1025.

Asarin, E., S. Bansal, B. Espiau, T. Dang and O. Maler (2001). On hybrid control of underactuated mechanical systems. In: *Hybrid Systems: Computation and Control.* LNCS 2034. Springer.

Dang, T. (2000). Verification and Synthesis of Hybrid Systems. PhD thesis. VERIMAG. Grenoble.

Martins, E., M. Pascoal and J. Dos Santos (1998). The k shortest paths problem. Technical report. Department of Mathematics, Coimbra University. Portugal.

Martins, E., M. Pascoal and J. Dos Santos (2000). Labelling algorithms for ranking shortest paths. Technical report. Department of Mathematics, Coimbra University. Portugal.

Tittus, M. and B. Egardt (1998). Control design for integrator hybrid systems. *IEEE Transactions on Automatic Control: Special Issue on Hybrid Systems* **43**(4), 491–500.

Tomlin, C., J. Lygeros and S. Sastry (1999). Controllers for reachability specifications for hybrid systems. *Automatica* **35**(3), 349–370.

Trontis, A. and M. Spathopoulos (2001*a*). Target control for hybrid systems with linear continuous dynamics. *Proceedings of the 40th CDC.*

Trontis, A. and M. Spathopoulos (2001*b*). Target control for linear hybrid systems. *Proceedings of the ECC01.*

Wong-Toi, H. (1997). The synthesis of controllers for linear hybrid automata. *Proceedings of the 36th CDC.*