Proceedings of the 2005 IEEE
International Symposium on Intelligent Control
Limassol, Cyprus, June 27-29, 2005



# Optimal control of discrete-time hybrid automata under safety and liveness constraints

Dmitry Gromov* , Eckart Mayer⋆, Jörg Raisch*,⋆, Daniele Corona‡, Carla Seatzu‡ and Alessandro Giua‡

*Lehrstuhl für Systemtheorie technischer Prozesse, Otto-von-Guericke-Universität Magdeburg, Germany
Email: {dmitry.gromov, joerg.raisch}@e-technik.uni-magdeburg.de
⋆ Systems and Control Theory Group, Max-Planck-Institut für Dynamik komplexer technischer Systeme, Magdeburg
‡Dip. di Ing. Elettrica ed Elettronica, Università di Cagliari, Italy
Email: {daniele.corona,seatzu, giua}@diee.unica.it

*Abstract*— **In this contribution we address an optimal control problem for a class of discrete-time hybrid automata under safety and liveness constraints. The solution is based on a hierarchical decomposition of the problem, where the low-level controller enforces safety and liveness constraints while the high-level controller exploits the remaining degrees of freedom for performance optimisation. Lower-level control is based on a discrete abstraction of the continuous dynamics. The action of low-level control can be interpreted as restricting invariants in the hybrid automaton representing the plant model.**

## I. Introduction

Hybrid automata are dynamic systems that consist of both continuous dynamics (modelled by a set of differential or difference equations) and a switching scheme (modelled by invariants and guards). Hybrid automata (and other modelling paradigms for hybrid systems) have been widely investigated because of their importance in many application areas. Often, the control objective for such systems is to minimise a cost function while respecting safety and liveness constraints. There are a number of abstraction-based control synthesis approaches that address safety and liveness issues while largely ignoring performance optimisation aspects. On the other hand, optimal control approaches to hybrid problems are often not able to handle "hard" safety constraints. It is therefore natural to combine both approaches to provide a method for synthesising a closed loop control strategy which minimises a given cost function under certain safety and liveness constraints.

In a previous paper, [1], the authors merged a supervisory control problem (addressing safety and liveness constraints) with an optimal control problem (addressing cost minimisation) for a hybrid automaton where only a *finite* number of switches is allowed. In this contribution an *infinite* number of switches is allowed and all dynamics may be non-Hurwitz. Furthermore in contrast to [1], both the supervisory control problem and the optimal control problem are solved within the framework of *discrete-time* hybrid automata. This allows a much more coherent way of treating both control synthesis aspects.

Work partially done in the framework of the HYCON Network of Excellence, contract number FP6-IST-511368

This contribution is organised as follows: in Section 2, we recall some basic facts on hybrid automata, introduce the plant model and formalise the specifications. In Section 3, the safety and liveness requirements are addressed using $\ell$-complete abstraction of the continuous plant dynamics. In Section 4, the remaining degrees of freedom are used to minimise a quadratic cost function. In Section 5, a numerical example is provided.

Finally, a remark regarding terminology. As time, i.e. the domain of signals, is discrete throughout this paper, the words "continuous" and "discrete" will always refer to the range of signals: continuous signals live in dense subsets of some Euclidean space, whereas discrete signals live in discrete, and for the purpose of this paper, finite sets; continuous (respectively discrete) systems are characterised by continuous (respectively discrete) signals.

## II. Plant Model and Specifications

In this section we first define the class of Hybrid Automata (*HA*) on which we focus attention. Then we formally describe the safety specifications and the optimal control problem.

### A. Hybrid Automata

A discrete-time hybrid automaton *HA* consists of a "classic" automaton extended with a continuous state $x \in \mathbb{R}^n$ that evolves in discrete time with arbitrary dynamics [2], [3]. The hybrid automaton considered here is a structure $HA = (L, act, inv, E)$ defined as follows, (see, e.g. [4]).

- $L = \{1, \dots, \alpha\}$ is a finite set of locations.
- $X \subseteq \mathbb{R}^n$ is a continuous state space.
- $act : L \to \{X \to X\}$ is a function that associates to each location $i \in L$ a discrete time difference equation of the form
$$x(k+1) = f_i(x(k)). \quad (1)$$
- $inv : L \to 2^X$ is a function that associates to each location $i \in L$ an invariant $inv_i \subseteq X$.
- $E \subset L \times 2^X \times L$ is the set of edges. An edge $e_{i,j} = (i, g_{i,j}, j) \in E$ is an arc between locations $i$ and $j$ with associated guard region $g_{i,j}$.

We denote by hybrid state the pair $(i, x)$ where the index $i$ identifies the discrete location $i \in L$ and $x \in \mathbb{R}^n$ is the continuous state.

Starting from initial state $\xi_0 = (i, x_0) \in L \times X$, $x_0 \in inv_i$, the continuous state $x$ may evolve according to the corresponding discrete-time transition function $f_i$, i.e., $x(k+1) = f_i(x(k))$, until it is about to leave the invariant $inv_i$, i.e. $f_i(x(k)) \notin inv_i$, $k \in \mathbb{N}$. This enforces a switch to another location $j$ satisfying the guard constraint $x(k+1) = f_i(x(k)) \in g_{i,j}$, and the future evolution of the continuous state is now determined by the transition function $f_j$. If several potential "follow-up" locations satisfy the constraint, this degree of freedom can be exploited by an appropriate discrete control scheme. Thus, the sequence $l(k)$ of discrete locations can be interpreted as a constrained control input. Note that the hybrid automaton may also switch to a "new" location $j$ *before* being forced to leave its "old" location $i$, if the corresponding guard constraint is satisfied.

### B. The Plant Model

In this paper we assume that the uncontrolled plant is modelled as a discrete-time hybrid automaton satisfying the following assumptions:

A1. (1) is linear, i.e.

$$x(k+1) = A_i x(k) \quad \forall i \in L, k \in \mathbb{N}_0.$$

A2. $inv_i = X \quad \forall i \in L.$

A3. $g_{i,j} = X \quad \forall (i, g_{i,j}, j) \in E.$

Hence, the uncontrolled plant is a switched linear system with no restrictions regarding the continuous evolution of the state (see A2) and the possibility to switch.

It will turn out in Section 3 that adding low-level control to the plant model will add nontrivial invariants to the plant automata. This may be interpreted as adding state space constraints that force the plant dynamics to respect safety and liveness constraints.

### C. Safety Specification

To formalise *safety* specifications, the continuous plant state space $X$ is partitioned via a function $q : X \to Y_d$, where $Y_d$ is a finite set of symbols. To express both static and dynamic safety constraints, certain sequences of $Y_d$ symbols are declared illegal or, in other words, the evolution of the hybrid automaton needs to be restricted such that only legal $Y_d$ strings are generated. It is assumed that this set of strings can be realised by a finite automaton $SP_Y$.

The *liveness* requirement implies that $\forall\ i \in L, \forall\ k \in \mathbb{N}_0$, the following holds: $x(k) \in inv_i$, $f_i(x(k)) \notin inv_i \Rightarrow \exists e = (i, g_{i,j}, j)$, $x(k) \in g_{i,j}$ and $x(k+1) = f_i(x(k)) \in inv_j$.

Note that the liveness condition guarantees the existence of an evolution $(i(k), x(k))$, $k \in \mathbb{N}_0$ from every initial hybrid state $(i, x_0)$.

### D. Optimal Control Problem

Subject to plant model (Sec. 2.2), safety and liveness constraints (Sec. 2.3), we aim at minimising the cost function

$$J = \sum_{k=0}^{\infty} x(k)' Q_{i(k)} x(k), \tag{2}$$

where, for each $k \geq 0$ $i(k) \in L$, $Q_{i(k)}$ is a positive semidefinite real matrix.

This problem will now be approached using a high-level control hierarchy. Safety and liveness requirements are being taken care of by the low-level control. This is described in Section 3. The remaining degrees of freedom are used to minimise the cost function (2). This is described in Section 4.

## III. The low-level task

In a first step, the hybrid plant automaton is approximated by a finite state machine using the $\ell$-complete approximation approach [5], [6]. Subsequently, Ramadge and Wonham's supervisory control theory [7] is implemented to synthesise a least restrictive supervisor. Note that, in general, controller synthesis and approximation refinement are iterated until a nontrivial supervisor guaranteeing liveness and safety for the approximation can be computed. Attaching the resulting supervisor to the hybrid plant model amounts to introducing restricted invariants. The resulting hybrid automaton represents the plant under low-level control and can be guaranteed to respect both safety and liveness constraints.

### A. Ordered set of discrete abstractions

The low-level control deals with a continuous system (1) with discrete external signals. $l : \mathbb{N}_0 \to L$ is the discrete control input and $y_d : \mathbb{N}_0 \to Y_d$ a discrete measurement signal. The set of output symbols, $Y_d$, is assumed to be finite: $Y_d = \{y_d^{(1)}, \ldots, y_d^{(\beta)}\}$, and $q_y : X \to Y_d$ is the output map. Without loss of generality, the latter is supposed to be surjective (*onto*). The output map partitions the state space into a set of disjoint subsets $Y^{(i)} \subset X$, $i = 1, \ldots, \beta$, i.e.

$$\bigcup_{i=1}^{\beta} Y^{(i)} = X,$$
$$Y^{(i)} \cap Y^{(j)} = \emptyset \quad \forall i \neq j\,.$$

To implement supervisory control theory, the hybrid plant model is approximated by a purely discrete one. This is done using the method of $\ell$-complete approximation [5], [8], which is described in the following paragraphs.

Denote the behaviour of the hybrid plant model by $\mathcal{B}_{plant}$, i.e. $\mathcal{B}_{plant} \subseteq (L \times Y_d)^{\mathbb{N}_0}$ is the set of all pairs of (discrete valued) input/output signals $w = (l, y_d)$ that (1) admits. In general, a time-invariant system with behaviour $\mathcal{B}$ is called $\ell$-complete if

$$w \in \mathcal{B} \Leftrightarrow \sigma^k w|_{[0,\ell]} \triangleq w|_{[k,k+\ell]} \in \mathcal{B}|_{[0,\ell]}\ \forall k \in N_0,$$

where $\sigma$ is the unit shift operator and $w|_{[0,\ell]}$ denotes the restriction of the signal $w$ to the domain $[0,\ell]$ [9]. For $\ell$-complete systems we can decide whether a signal belongs to the system behaviour by looking at intervals of length $\ell$. Clearly, an $\ell$-complete system can be represented by a difference equation in its external variables with lag $\ell$. The hybrid plant model (1) is, except for trivial cases, not $\ell$-complete. For such systems, the notion of *strongest $\ell$-complete approximation* has been introduced in [8]: a time-invariant dynamical system with behaviour $\mathcal{B}_\ell$ is called strongest $\ell$-complete approximation for $\mathcal{B}_{plant}$ if

- $(i)$ $\mathcal{B}_\ell \supseteq \mathcal{B}_{plant}$,
- $(ii)$ $\mathcal{B}_\ell$ is $\ell$-complete,
- $(iii)$ $\mathcal{B}_\ell \subseteq \tilde{\mathcal{B}}_\ell$ for any other $\ell$-complete $\tilde{\mathcal{B}}_\ell \supseteq \mathcal{B}_{plant}$,

i.e. if it is the "smallest" $\ell$-complete behaviour containing $\mathcal{B}_{plant}$. Obviously, $\mathcal{B}_\ell \supseteq \mathcal{B}_{\ell+1}\ \forall \ell \in \mathbb{N}$, hence the proposed approximation procedure may generate an ordered set of abstractions. Clearly, $w \in \mathcal{B}_\ell \Leftrightarrow w|_{[0,\ell]} \in \mathcal{B}_{plant}|_{[0,\ell]}$. For $w|_{[0,\ell]} = (l_0,\ldots,l_\ell, y_d^{(i_0)},\ldots,y_d^{(i_\ell)})$ this is equivalent to

$$\begin{aligned} &f_{l_{\ell-1}}\left(\ldots f_{l_1}\left(f_{l_0}\left(q_y^{-1}(y_d^{(i_0)})\right)\cap\left(q_y^{-1}(y_d^{(i_1)})\right)\right)\right.\\ &\ldots (q_y^{-1}(y_d^{(i_{\ell-1})})\Big)\cap q_y^{-1}(y_d^{(i_\ell)}) \triangleq X(w|_{[0,\ell]}) \neq \emptyset, \end{aligned} \tag{3}$$

where $l_i \in L$. Note that for a given string $w|_{[0,\ell]}$, $X(w|_{[0,\ell]})$ represents the set of possible values for the continuous state variable $x(\ell)$ if the system has responded to the input string $l(0) = l_0,\ldots,l(\ell-1) = l_{\ell-1}$ with the output $y_d(0) = y_d^{i_0},\ldots,y_d(\ell) = y_d^{i_\ell}$ and that (3) does not depend on $l(\ell)$. For linear and affine systems evolving on discrete time $\mathbb{N}_0$, (3) can be checked *exactly*.

As both input and output signal evolve on finite sets $L$ and $Y_d$, $\mathcal{B}_\ell$ can be realised by a (nondeterministic) finite automaton. In [5], [8], a particularly intuitive realisation is suggested, where the approximation state variable stores information on past values of $l$ and $y_d$. More precisely, the automaton state set can be defined as

$$X_d := \bigcup_{j=0}^{\ell-1} X_{d_j},\ \ell \geq 1,$$

where $X_{d_0} = Y_d$
and $X_{dj}$ is the set of all strings such that $\exists\ l_j \in\ L : (l_0,\ldots,l_j,y_d^{(i_0)},\ldots,y_d^{(i_j)}) \in \mathcal{B}|_{[0,j]}$.

The temporal evolution of the automaton can be illustrated as follows:
From initial state $x_d(0) \in X_{d0}$, it evolves through states

$$x_d(j) \in X_{dj},\ 1 \leq j \leq \ell-1$$

while

$$x_d(j) \in X_{d\ell-1},\ j \geq \ell-1.$$

Hence, until time $\ell-1$, the approximation automaton state is a complete record of the system's past and present, while from then onwards, it contains only information on the "recent" past and present.

As the states $x_d^{(i)} \in X_d$ of the approximation realisation are strings of input and output symbols, we can associate $x_d^{(i)}$ with a set of continuous states, $X(x_d^{(i)})$, in completely the same way as in (3).

Note that we can associate $y_d^{(i)}$ as the unique output for each discrete state $x_d^{(i)} \in X_d$. The resulting (nondeterministic) Moore-automaton $M_\ell = (X_d, L, Y_d, \delta, \mu, X_{d_0})$ with state set $X_d$, input set $L$, output set $Y_d$, transition function $\delta : X_d \times L \to 2^{X_d}$, output function $\mu : X_d \to Y_d$, and initial state set $X_{d_0}$ is then a realisation of $\mathcal{B}_\ell$. Note that the state of $M_\ell$ is instantly deducible from observed variables.

To recover the framework of supervisory control theory [7] as closely as possible, we finally convert $M_\ell$ into an equivalent automaton without outputs, $G_\ell = (\tilde{X}_d, \Sigma, \tilde{\delta}, \tilde{X}_{d_0})$, where $\Sigma = L \cup Y_d$, $L$ represents the set of controllable events and $Y_d$ the set of uncontrollable events.

Technically, this procedure is carried out according to the following scheme (for an illustration, see Fig.1):

- Each state $x_d^{(j)} \in X_d$ is split into two states: $x_d^{(j)}$ and $\hat{x}_d^{(j)}$. Thus, the new state set is formed as $\tilde{X}_d = X_d \bigcup \hat{X}_d$. The set of initial states remains the same, $\tilde{X}_{d_0} = X_{d_0}$.
- The new transition function $\tilde{\delta}$ is defined as a union of two transition functions with nonintersecting domains:

$$\tilde{\delta}(\tilde{x}_d^{(i)}, \sigma^{(j)}) = \begin{cases} \delta(\backsim\tilde{x}_d^{(i)}, \sigma^{(j)}), & \tilde{x}_d^{(i)} \in \hat{X}_d,\ \sigma^{(j)} \in L, \\ \backsim\tilde{x}_d^{(i)}, & \tilde{x}_d^{(i)} \in X_d, \\ & \quad \sigma^{(j)} = \mu(\tilde{x}_d^{(i)}) \in Y_d, \\ \emptyset, & \text{otherwise,} \end{cases}$$

  where $\backsim$ denotes an operation of taking the complementary state, i.e. $\backsim\hat{x}_d^{(i)} \triangleq x_d^{(i)}$ and vice versa. Note that the first event always belongs to the set $Y_d$, the following evolution consists of sequences where events from $L$ and $Y_d$ alternate.

### B. Specification and supervisor design

Safety requirements can often be formalised as a set of acceptable pairs of input/output signals. In many applications we have independent specifications for both inputs and outputs, which can be realised by finite automata $SP_L = (S_L, L, \delta_L, S_{L0})$ and $SP_Y = (S_Y, Y_d, \delta_Y, S_{Y0})$. These automata can be characterised according to their *current-state observability*:

*Definition 1:* [10] A finite state machine $A = (Q, \Sigma, \phi)$ is said to be *current-state observable* if there exists a nonnegative integer K such that for every $i \geq K$, for any initial state $q(0)$, and for any sequence of events $\sigma(0)\ldots\sigma(i-1)$ the state $q(i)$ can be uniquely determined. The parameter $K$ is referred to as the *index of observability*.

Deterministic finite automata are basically current-state observable. If there are indistinguishable states, they can be merged without changing the behaviour. Thus, we can use the current-state observability indices $K_L$ and $K_Y$ to characterise the specification automata $SP_L$ and $SP_Y$.

The next step is to design an overall specification modelled by a finite automaton $SP = SP_L \| SP_Y$.

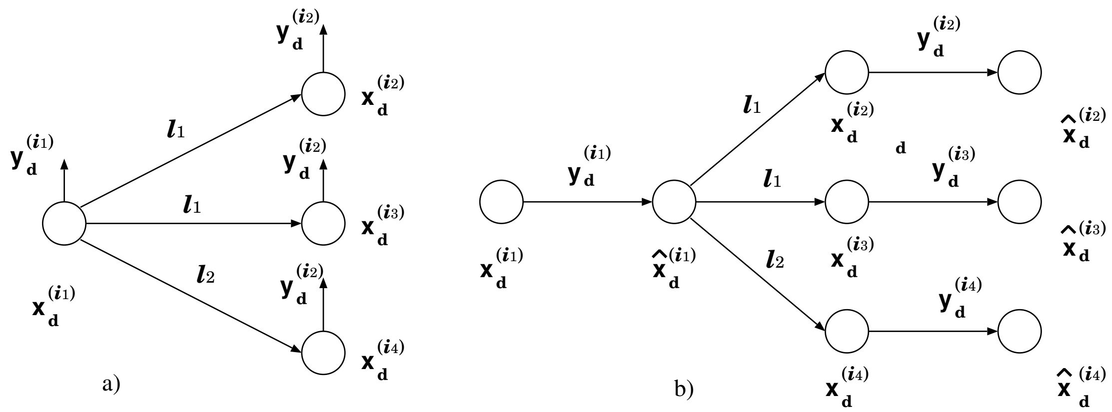


Fig. 1. *Moore-automaton a) and an equivalent automaton without outputs b). Note that $y_d^{(i_j)} = \mu(x_d^{(i_j)}) \in Y_d$ is the output symbol associated with the discrete state*

Given an approximating automaton $G_l$ and a specification automaton $SP$, supervisory control theory checks, whether there exists a nonblocking supervisor and, if the answer is affirmative, provides a least restrictive supervisor $SUP$ via "trimming" of the product of $G_l$ and $SP$. Hence the state set of the supervisor, $X_{SUP}$, is a subset of $\tilde{X} \times S$.

The functioning of the resulting supervisor is very simple. At time $t_k$ it "receives" a measurement symbol which triggers a state transition. In its new state $x_{sup}^{(j)}$, it enables a subset $\Gamma(x_{sup}^{(j)}) \subseteq L$ and waits for the next feedback from the plant. As shown in [8], the supervisor will enforce the specifications not only for the approximation, but also for the underlying hybrid plant model (1).

In the following, we will be interested in the special case of *quasi-static* specifications. To explain this notion, let $p_{app} : X_{SUP} \to \tilde{X}$ denote the projection of $X_{SUP} \subseteq \tilde{X} \times S$ onto its first component. If $p_{app}$ is injective, the specification automaton is called quasi-static with respect to the approximation automaton $G_l$.

*Proposition 1:* $SP$ is quasi-static with respect to $G_\ell$ if

$$\ell \geq max(K_L, K_Y - 1).$$

### C. Closed loop model

For the case of quasi-static specifications, each supervisor state $p_{app}(x_{sup}^{(i)})$ corresponds exactly to a state $\tilde{x}_d^{(i)} = p_{app}(x_{sup}^{(i)})$ of the approximating automaton, which, in turn, can be associated with a set $X(\tilde{x}_d^{(i)}) = X(p_{app}(x_{sup}^{(i)}))$. Note that on the underlying, physical level the state $x_d^{(i)} \in X_d$ and its complement $\hat{x}_d^{(i)} \in \hat{X}_d$ are equivalent in sense that $X(x_d^{(i)}) \triangleq X(\hat{x}_d^{(i)})$.

For $k \geq \ell$, attaching the discrete supervisor to the plant model (1) is therefore equivalent to restricting the invariants for each location $l_j \in L$ according to

$$inv_{l_j} = \bigcup_{\substack{l_j \in \Gamma(\hat{x}_s^{(i)}) \\ i,\, p_{app}(x_{sup}^{(i)}) \in \hat{X}_{d_{\ell-1}}}} X(p_{app}(x_{sup}^{(i)})). \qquad (4)$$

Note that for the initial time segment, i.e. $k \leq \ell$, (4) is more restrictive than the discrete supervisor computed in Sec.III-B.

The union of all invariants $inv_{l_j}$, $j = 1, \ldots, \alpha$ forms the refined state set that contains only safe points, i.e. points for which exists at least one sequence of control symbols such that the resulting behaviour satisfies the specification.

The resulting hybrid automaton represents the plant model (1) under low-level control (for $k \geq \ell$). As control system has been based on an $\ell$-complete approximation of (1), it is guaranteed that the resulting hybrid automaton satisfies safety and liveness requirements. The remaining degrees of freedom in choosing $l(k)$ can be used in a high-level controller addressing performance issues.

## IV. The high-level task

The high-level task requires the solution of an optimal control problem of the form (2).

In previous works the authors proposed a technique to solve this problem in the particular cases where

(a) $inv_i \subseteq \mathbb{R}^n$ and a finite number of allowed switches $N$ [1];
(b) $inv_i \equiv \mathbb{R}^n$, $\forall\ i$ and an infinite number of switches allowed [11].

In both cases the method consisted in using dynamic programming approach over an infinite time horizon. The solution is a partition $C$ of the state space for each location, that we named as *switching tables*. When the system evolves in a given location $i \in L$, then the controller considers the corresponding table and imposes one switch to location $j$, iff the value of $x$ enters a partition mapped by $j$. In case (a) we had one table per location $i$ and per number of missing switches $m$, that we called $C^i_m$. In case (b) we proved that the switching tables converge to the same one when $m$ grows. More precisely we proved that there exists a sufficiently big value of $N$ such that $\forall\ m > N$, $C^i_m = C^i_{N+1}$. We called this table by $C^i_\infty$. Furthermore in paper [12] we proved that if the automaton is completely connected then $C^i_\infty = C^j_\infty = C_\infty$, $i \neq j$.

Here we investigate the possibility of extending the previous results. In particular we merge (a) and (b), i.e., we consider a constrained problem $inv_i \subseteq \mathbb{R}^n$ as in case (a), where we additionally allow the number of switches $N$ to grow indefinitely, as in (b). To this aim we introduce some new definitions and propose some new results. For simplicity we will only deal with completely connected automata.

*Definition 2 (Forbidden region):* A forbidden region for the $HA$ is a set $X_f \subset X : X_f = X \setminus \bigcup_{i=1}^{s} inv_i$, where $s$ is the number of locations. ■

Thus $X_f$ is a region forbidden to *all* dynamics of the $HA$.

*Definition 3 (Augmented HA and OP):* An *augmented automaton* $\overline{HA} = (\overline{L}, \overline{act}, \overline{inv}, \overline{E})$ of $HA = (L, act, inv, E)$ and the corresponding optimal control problem $\overline{OP}$ of $OP$, are related as follows:

(i) $\overline{HA}$ includes a new dynamics $A_{\alpha+1}$ and $\overline{OP}$ includes a corresponding weight matrix $Q_{\alpha+1} = q\tilde{Q}_{\alpha+1}$ (with $rank(\tilde{Q}_{\alpha+1}) \neq 0$, and $q > 0$). such that $\forall\ x_0 \in X$ the cost value

$$J(x_0) = \sum_{k=0}^{\infty} x(k)' Q_{\alpha+1} x(k)$$
$$s.t. \quad x(k+1) = A_{\alpha+1} x(k)$$

is finite[1].

(ii) A new invariant $inv_{\alpha+1} = \mathbb{R}^n$ is associated to the new dynamics.

(iii) The edges $e_{i,\alpha+1} \in \overline{E}$ and $e_{\alpha+1,i} \in \overline{E}$ are defined $\forall\ i \in \overline{L}$.

■

Thus the augmented automaton $\overline{HA}$ is the same as $HA$ except for an extra location $(\alpha+1)$ completely connected to all the locations in the $HA$. Its invariant set coincides with $inv_{\alpha+1} = \mathbb{R}^n$ and its dynamics is $A_{\alpha+1}$. The corresponding $\overline{OP}$ weights location $(\alpha+1)$ with matrix $Q_{\alpha+1} \geq 0$.

Now we implement the switching table procedure [1] to the augmented problem $\overline{OP}(\overline{HA})$ with a finite number of switches $N$. If we increase $N$ recursively, as described in [11], we obtain the following results, whose proofs are briefly sketched, being simple extensions of known results.

*Proposition 2:* All tables converge when $N$ grows, i.e., $\forall\ i \in \overline{L}$

$$\lim_{N\to\infty} \overline{C}^i_N = \overline{C}^i_\infty$$

[1]Note that a structural property that certainly ensures this condition is that the matrix $A_{\alpha+1}$ is Hurwitz stable, i.e., all its eigenvalues are inside the unit circle. Nevertheless this condition is not strictly necessary.

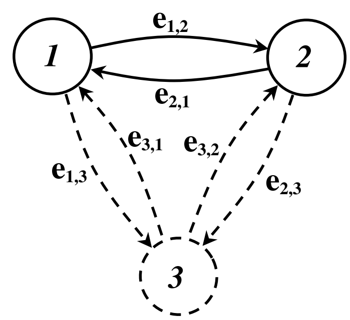


Fig. 2. *Graph of the automaton HA (continuous) and $\overline{HA}$ (continuous and dashed) described in the example.*

Moreover if the $\overline{HA}$ is completely connected then

$$\overline{C}^i_\infty = \overline{C}_\infty,\ \forall\ i \in \overline{L},$$

i.e., all tables converge to the same one. ■

*Proposition 3:* Assume that there exists an exponentially stabilising switching law for problem $OP(HA)$. Then there also exists a sufficiently large value of $q > 0$ in the $\overline{OP}(\overline{HA})$, such that the tables $\overline{C}^i_\infty$, solution of $\overline{OP}(\overline{HA})$, $i = 1, \ldots, \alpha+1$, contain the color of $A_{\alpha+1}$ at most in $X_f$. ■

Note that Proposition 2 is formally proved in [11] in absence of state space constraints. It can be trivially extended to this case, provided that the invariants calculated in Section III guarantee the liveness of the $HA$. Proposition 3 allows one to consider the solution of $\overline{OP}(\overline{HA})$ equivalent to the solution of $OP(HA)$. This follows from the fact that the dynamics $A_{\alpha+1}$ does not influence at all any solution of the augmented problem. Therefore it can be removed from the augmented automaton. These results are formally proved in [12], in absence of state space constraints. As before this result can be trivially extended if the liveness of the automaton is guaranteed. In fact, by definition, it holds that, for any initial couple $(i, x_0) \notin X_f$ of the $HA$, the hybrid trajectory, solution of $OP(HA)$, $(i(k), x_(k))$, never enters $X_f$.

## V. Numerical example

Consider the $HA$ with two locations and corresponding dynamics

$$A_1 = \begin{bmatrix} 0.981 & 0.585 \\ -0.065 & 0.981 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.981 & 0.065 \\ -0.585 & 0.981 \end{bmatrix}$$

whose eigenvalues are, for both dynamics, $\lambda_{1,2} = 0.9808 \pm j0.1951$, of norm 1 (see Figure 4 for the corresponding trajectories at the limit cycle). The safety constraint in the state space is given by the forbidden state set

$$X_f = \{x \in \mathbb{R}^2 | H'x \leq h\}$$

where

$$H = \begin{bmatrix} 0 & 0 & 1 & -1 \\ 1 & -1 & -1 & -1 \end{bmatrix}$$
$$h = \begin{bmatrix} 0.8 & -0.2 & 0 & 0 \end{bmatrix}. \tag{5}$$

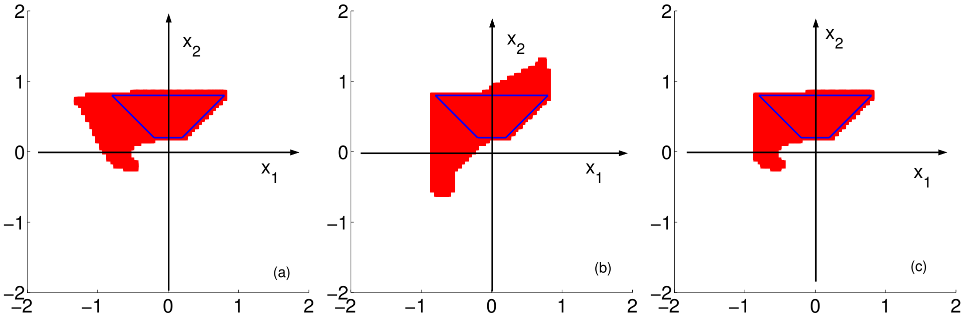


Fig. 3. *Invariants (in white) of locations* 1 *(a) and* 2 *(b) and (c) the forbidden region* $X_f = X \setminus (inv_1 \bigcup inv_2)$ *defined in Def. 2. The interior of the blue trapezoid is the forbidden region* $X_d$

$X_d$ is the trapezoid depicted in Fig.3. Note that the set $X \setminus X_f$ can be blocking, i.e., some admissible initial points will violate the constraint, regardless of the switching strategy. Thus the previous setup is passed to the procedure described in Section III, in order to compute the invariants $inv_1$ and $inv_2$ that guarantee liveness (i.e., the resulting automaton is non blocking) and safety (i.e., the state never enters $X_f$). This leads to an extension of the forbidden region, as illustrated in Fig.3.

The graph of the hybrid automaton (HA) is depicted in Figure 2 (the part sketched with continuous lines).

Within the given constraints we want to solve an optimal control problem[2] $OP$ of the form (2), where $Q_1 = Q_2 = I$. For this purpose we consider the augmented problem $\overline{OP}(\overline{HA})$, with the following data:

$$A_3 = \begin{bmatrix} 0.9808 & 0.1950 \\ -0.1950 & 0.9801 \end{bmatrix}, \qquad Q_3 = qQ_1, \qquad inv_3 \equiv X$$

where $q = 10^5$, and $A_3$ is stable. The graph of the augmented automaton is depicted in Fig.2 (continuous and dashed part).

*Remark 1:* Let us observe that, for sake of symmetry, the solution of $OP(HA)$ when $inv_i \equiv \mathbb{R}^2$, $i = 1, 2$, is to use dynamics $A_2$ when $x_1x_2 > 0$ and dynamics $A_1$ when $x_1x_2 < 0$. This result is very intuitive if we observe the trajectories of the given dynamics (Figure 4) and if we use the identity matrices as weight matrices in problem (2). Moreover it is simple to prove that for any initial state of the form $x_0 = [a\ 0]'$ or $x_0 = [0\ a]'$, $J(a) = 5.5a^2$.

Note that the augmented problem $\overline{OP}(\overline{HA})$ satisfies the conditions given in Definition 3. The switching table procedure, applied to $\overline{OP}(\overline{HA})$ for a recursively increasing number of switches, converges after $N = 15$ switches. Moreover the tables $C^i_\infty$, $i = 1, 2, 3$ are the same, because $\overline{HA}$ is completely connected, as in Proposition 2.

This table is depicted in Figure 5, and it is clearly affected by numerical disturbances, due to the presence of a state space discretisation. However some important things should be remarked:

[2]Note that neither $A_1$ nor $A_2$ are Hurwitz, hence an infinite number of switches is necessary.

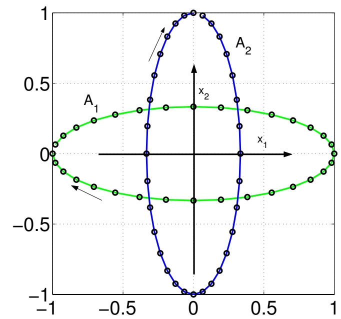


Fig. 4. *Discrete time trajectories of dynamics* $A_1$ *and* $A_2$*, with eigenvalues along the unitary circle*

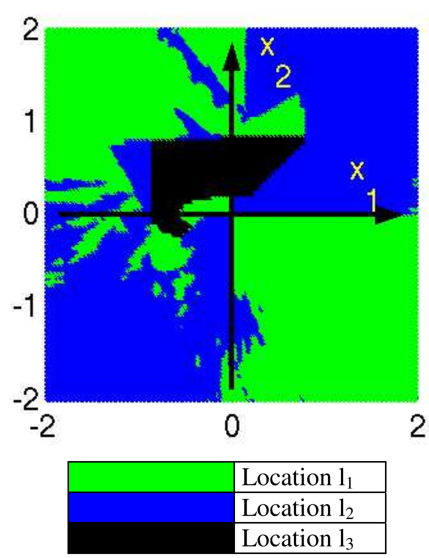


Fig. 5. *Switching table of the problem* $\overline{OP}(\overline{HA})$ *defined in the example*

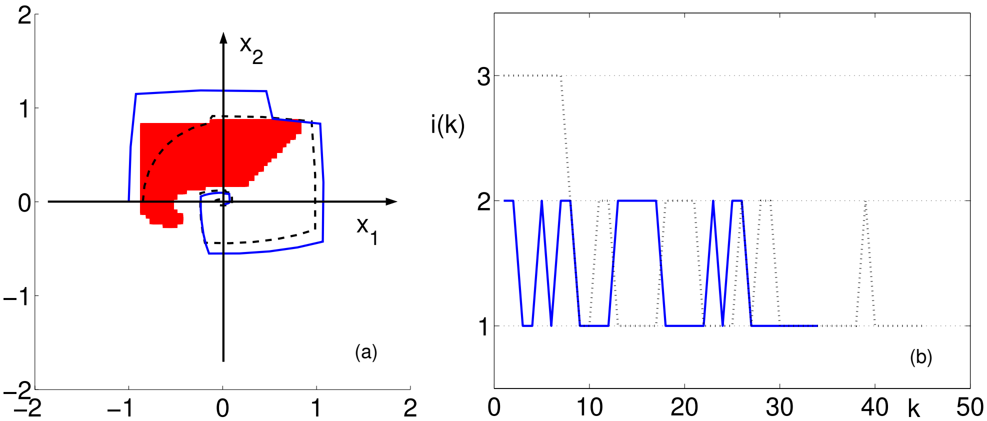


Fig. 6. *Trajectories $x(k)$ (a) and $i(k)$ (b) of the optimal solution of $OP(HA)$ obtained by using the table in Figure 5 for an admissible initial point ($i_0 = 1, x_0 = [-1\ 0]'$) (continuous) and a forbidden one ($i_0 = 1, x_0 = [-0.85\ 0]'$)(dashed)*

(i) The color of the augmented dynamics exactly covers the region $X_f$;
(ii) By virtue of (i) and Proposition 3 the solution of $OP(HA)$ coincides with the solution of $\overline{OP}(\overline{HA})$;
(iii) The solution of $OP(HA)$ is a perturbation, around the forbidden region, of the solution described in Remark 1.

From (i) and (ii) we deduce that there exists a finite optimal solution for any initial hybrid state $(i_0, x_0) \notin X_f$ of the $HA$, and that if $(i_0, x_0) \in X_f$ the optimal solution of $\overline{HA}$ uses dynamics $A_3$ for the minimum time required to leave $X_f$. From then on the optimal solution of $HA$ is used. This can be viewed by the simulations depicted in Figure 6(a) for an admissible point (continuous line) and a forbidden point (dashed line). The optimal cost from the admissible point is $J = 15.7$, and for the other one is $J = 5.05 \cdot 10^5$. For completeness also the index trajectory $i(k)$ is reported in Figure 6(b).

From Figure 6(a) it can be seen that once the "obstacle" $X_f$ is avoided, the systems steers towards the origin by following the solution provided in Remark 1.

The total computational time (Matlab, up to date laptop) for constructing the table in Figure 5 is about 40 hours. This time is extremely big, but a very dense space discretisation was considered ($1.6 \times 10^5$ points). It is important, however, to point out that this computational effort is spent off-line. The on-line part of the procedure consists in measuring the hybrid state $(i(k), x(k))$ and comparing its value with the switching table to decide the optimal strategy.

## VI. Conclusion

We addressed the problem of designing a feedback control law for a discrete time hybrid automaton $HA$. We showed that this law can be designed so that the system's behaviour satisfies two levels of specifications. The former (the *low level* specification) exposes liveness and safety conditions for the $HA$. We showed that the action of the low-level controller is to restrict the invariants of $HA$. The latter (the *high level* task), performs an optimisation search. In particular, within the degree of freedom left by the low level task, for a given initial state it finds the evolution that minimises a given performance index. Although the procedure is theoretically successful, it may lack in numerical robustness. One perspective of interest for future developments is to provide structural conditions of the $HA$ that guarantee the existence of admissible optimal control laws.

## References


[1] D. Corona, C. Seatzu, A. Giua, D. Gromov, E. Mayer, and J. Raisch, "Optimal hybrid control for switched affine systems under safety and liveness constraints," in *Proceedings of CACSD'05*, Taipei, Taiwan, 2004, pp. 35–40.

[2] X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "An approach to the description and analysis of hybrid systems," in *Hybrid Systems, LNCS 736, Springer Verlag*, 1993.

[3] R. Alur and D. Dill, "A theory of timed automata," in *Theoretical Computer Science*, 1994.

[4] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pneuli, "Effective synthesis of switching controllers for linear systems," *Proceedings of the IEEE*, vol. 88, no. 7, 2000.

[5] J. Raisch and S. O'Young, "Discrete approximation and supervisory control of continuous systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 569–573, April 1998.

[6] T. Moor, J. Raisch, and S. O'Young, "Discrete supervisory control of hybrid systems by *l*-complete approximations," *Journal of Discrete Event Dynamic Systems*, vol. 12, no. 1, pp. 83–107, 2002.

[7] P. Ramadge and W. Wonham, "The control of discrete event systems," *Proc. IEEE, Special Issue on Discrete Event Dynamic Systems*, vol. 77, no. 1, pp. 81–98, January 1989.

[8] T. Moor and J. Raisch, "Supervisory control of hybrid systems within a behavioural framework," *Systems and control letters*, vol. 38, pp. 157–166, 1999, special issue on *Hybrid Control Systems*.

[9] J. Willems, "Models for dynamics," *Dynamics reported*, vol. 2, 1989.

[10] P. Caines, R. Greiner, and S.Wang, "Dynamical logic observers for finite automata," in *Proceedings of the 27th Conference on Decision and Control*, Austin, Texas, Dec. 1988, pp. 226–233.

[11] D. Corona, A. Giua, and C. Seatzu, "Optimal control of hybrid automata: an application to the design of a semiactive suspension," *Control Engineering Practice*, vol. 12, pp. 1305–1318, 2004.

[12] ——, "Stabilization of switched systems via optimal control," in *Proc. IFAC World Congress*, Prague, Czech Republic, 2005, submitted.