On the Stabilization of Linear Discrete–time Hybrid Automata[1]

Marco Zoncu[†,♭], Andrea Balluchi[†],
Alberto L. Sangiovanni–Vincentelli[†,‡], and Antonio Bicchi[♭]

[†]PARADES, Via di San Pantaleo, 66, 00186 Roma, Italy.
{mzoncu, balluchi, alberto}@parades.rm.cnr.it

[♭]Centro Interdipartimentale di Ricerca "Enrico Piaggio". Università di Pisa,
Via Diotisalvi 2, 56100 Pisa, Italy, bicchi@ing.unipi.it

[‡]Electrical Engineering and Computer Science Dept.,
University of California at Berkeley, Berkeley, CA 94720, USA

Abstract

The problem of stabilizing linear *discrete–time* hybrid automata is considered. A synthesis methodology is obtained by extending to hybrid systems the stabilization techniques based on stable convex combinations, originally developed for switching systems. An algorithm to explore the candidate stabilizing controller actions is proposed and an application to an automotive engine control problem is described.

1 Introduction

The problem of controlling *discrete–time* hybrid automata has been approached extensively by several research groups. An important approach to hybrid controller synthesis for these systems was proposed by Bemporad and Morari (see [3]), who developed hybrid predictive and optimal control algorithms based on the MLD modeling framework. The generality and conceptual simplicity of the MLD method is very attractive; however, these advantages are paid in terms of computational complexity, since the approach is based on the solution of an integer programming problem, an NP-hard problem. The computational complexity of the most used algorithms for the solution of an integer programming problem is exponential in the dimension of the problem, where the dimension is given in terms of the number of variables and constraints. Since the number of variables is proportional to the number of time steps needed to cover the temporal horizon considered, the method is applicable to reasonably small problems.

In this paper, we present a method that takes advantage of the *structural properties* of the system to lower the computational complexity.
We were inspired by the work on the stability properties of continuous–time switched systems carried out in [4], [8], [6]and [9] and in particular, by the results of Wicks who introduced the method of *stable convex combinations* (see [11]), and by those of Morse for the synthesis of robust supervisory controllers (see [10]). Unfortunately, these results cannot be applied directly to general hybrid systems. In fact, in switched control problems, there are no constraints on the possible sequences of discrete control actions applied by the supervisor, while, in hybrid automata, transitions are controlled by either external (controllable or uncontrollable) events and/or by the evolution of dynamics corresponding to each state of the automaton, and the set of allowed controller discrete actions depends on the hybrid state. Our idea is *to "convert" the hybrid control problem into a switched one by leveraging the structure of the hybrid automaton.* In particular, given an initial state q_0, we find all cycles in the automaton that pass through q_0. Then the sequences of control actions that correspond to each of these cycles can be applied in any order. If we consider these sequences as atomic control actions, we have a control problem that is a switched control problem. Consequently, it can be solved applying the stable convex combinations approach [11]. This approach is less general than the MLD approach since we consider only a subset of all possible control laws, but it is more computationally attractive since finding cycles and solving linear matrix inequalities (the core of the stable combinations method) has polynomial complexity. However, considering sequences that correspond to going more than one time around the cycle is advantageous for our purposes. The number of turns is in principle unbounded. We have derived conditions that allow us to truncate

[1]The work has been conducted with partial support by the European Community Projects IST-2001-33520 CC (Control and Computation) and IST-2001-37170 RECSYS.

these sequences when they do not offer any advantage and in case these conditions do not apply, we artificially set a limit on the number of turns to maintain the computational attractiveness of the method. The paper is organized as follows. In Section 2, the problem is formulated. In Section 3, sufficient conditions for the synthesis of stabilizing switching control laws are proposed, exploiting the use of regular expressions for the representation of the cyclic paths in the automaton. In Section 4, an algorithm to carry out an exploration of the sets of admissible stabilizing controller actions is described. Finally, in Section 5, the proposed technique is applied to the design of a stabilizing controller for an automotive engine.

2 Problem formulation

Let us consider the problem of stabilizing DT (Discrete Time) hybrid automata (see [5]), defined as follows: a *DT hybrid automaton* is a tuple $H = (\mathcal{C}, \mathcal{C}_0, \Sigma, M, \varphi, \mathcal{F})$. Elements of $\mathcal{C} = \mathcal{Q} \times X$ are called *configurations*, where $\mathcal{Q} = \{q_1, \ldots, q_N\}$ is a finite set of *locations* and $X \subseteq \mathbb{R}^n$ a continuous[1] set of *states*. The set $\mathcal{C}_0 \subseteq \mathcal{C}$ is the subset of admissible initial configurations. The controller input comes from the domain Σ, where Σ is a finite set of *discrete control events*. The *discrete controller move function* $M : \mathcal{C} \to 2^\Sigma \setminus \{\}$ defines a subset of allowable discrete input values for every configuration. The evolution of the discrete state q is given by the deterministic *transition function* $\varphi : \mathcal{Q} \times \Sigma \to \mathcal{Q}$

$$q(k+1) = \varphi(q(k), \sigma(k+1)) ,$$

with input events satisfying $\sigma(k+1) \in M(q(k), x(k))$. The function $M(q, x)$ is assumed to be expressed by a set of quadratic *guard conditions* $G_i(x) = x^T W_i x + v_i^T x + c_i \leq 0$, each one associated to an event σ_i. The dynamics of the continuous state x is modeled by the family $\mathcal{F}$ of autonomous transition-dependent next-state linear functions $f_{(q(k),q(k+1))}$: given $(x(k), q(k)) \in \mathcal{C}$, for all $q(k+1) \in \mathcal{Q}$ such that $q(k+1) = \varphi(q(k), \sigma)$ for some $\sigma \in M(q(k), x(k))$,

$$x(k+1) = f_{(q(k),q(k+1))}(x(k)) = F_{(q(k),q(k+1))}x(k) .$$

Given a location $\bar{q} \in \mathcal{Q}$, consider all control sequences that cause the system to cycle back on $\bar{q}$. These sequences correspond to cycles in the graph of the hybrid automaton that start and end at $\bar{q}$. The main idea of the paper is that, since each one of these sequences can follow any other then hybrid system stabilization can be solved applying the stable convex combinations approach proposed for switching systems [11] to cyclic paths. The resulting stabilizing control law will be the correspoding composition of transformations associated to the cyclic paths. According to these considerations, we state the stabilization problem as follows:

Problem 1 *Given a linear DT hybrid automaton $\mathcal{H} = (\mathcal{C}, \mathcal{C}_0, \Sigma, M, \varphi, \mathcal{F})$, find a hybrid feedback control $\sigma \in M(q, x)$, such that a location $\bar{q} \in Q$ and a function $V(x) = x^T P x$ exist, with $P \in \mathbb{R}^{n \times n}$ and $P = P^T > 0$, for which we have that $\forall$ $(q(0), x(0)) \in \mathcal{C}_0$ and $\forall i \geq 0$, $V(x(h_{i+1})) - V(x(h_i)) < 0$, where h_i is the sequence of indexes for which $q(h_i) = \bar{q}$, $\lim_{i\to\infty} h_i = \infty$ and $h_{i+1} - h_i$ is bounded.*

Notice that the previous definition requires strict contractivity of the function $V(x)$ only on the subsequence h_i of the discrete time instants. This is enough for our purposes, since for linear hybrid automata, the continuous state has bounded evolutions over bounded time windows $h_i, \ldots, h_{i+1}$.

3 Stabilization of DT Hybrid Automata

In this section, we give sufficient conditions for stabilizing DT hybrid automata by switching between feasible cyclic paths. The number of such paths to be considered is quite large, since we have to look at simple cycles, as well as at multiple ones. In the following, we provide sufficient conditions for the elimination of cyclic paths that are dominated by others, thus reducing the complexity of the computation and of the resulting stabilizing control law. In the characterization of cyclic paths, we make use of the theory of regular expressions. A *regular expression* E is obtained by composing the symbols of a finite alphabet $\mathcal{A}$ with the operators '+' (union), '·' (concatenation) and '*' (infinite repetition). A *string* is a finite concatenation of symbols. A rigorous and comprehensive treatment of this topic can be found in [7]. In the following, we identify cyclic paths on a DT hybrid automaton $\mathcal{H}$ with the strings accepted by the finite automaton $\mathcal{H}_F$, obtained from $\mathcal{H}$ by abstracting away the continuous dynamics.

Definition 2 *Given a DT hybrid automaton $\mathcal{H}$, define the finite automaton $\mathcal{H}_F$ as a tuple $(\mathcal{Q}, \mathcal{A}, \varphi_D, q_0, \mathcal{Q}_f)$ (see [7]) where: the set of states is the set $\mathcal{Q}$ of locations of $\mathcal{H}$; the set of input symbols $\mathcal{A}$ is defined by a map $\Psi : \mathcal{Q} \times \mathcal{Q} \to \mathcal{A}$, such that $a = \Psi(q_1, q_2)$ if $\varphi(q_1, \sigma) = q_2$ for some $\sigma \in M(q_1, x)$ and $x \in \mathbb{R}^n$, with Ψ invertible in $\mathcal{A}$; the transition function $\varphi_D : \mathcal{Q} \times \mathcal{A} \to \mathcal{Q}$ is defined as $q_2 = \varphi_D(q_1, a)$ for all $a = \Psi(q_1, q_2)$; the initial location q_0 is any location in $\mathcal{Q}$ reachable by $\mathcal{H}$ from any $(q, x) \in \mathcal{C}_0$; the set $\mathcal{Q}_f$ of accepting locations is $\{q_0\}$. Then, we say that a regular language $\mathcal{L}$ is accepted by $\mathcal{H}$, if and only if it is accepted by $\mathcal{H}_F$.*

[1] The term *continuous* refers to the domain of the state values. For DT hybrid automata, the time domain is obviously discrete.

Let $\mathcal{L}$ be the regular language accepted by $\mathcal{H}_F$. Since $\mathcal{Q}_f = \{q_0\}$, then $\mathcal{L}$ can always be written as $\mathcal{L} = (E_1 + \ldots + E_S)^*$ where $E_1 \ldots E_S$ are regular expressions on the input alphabet $\mathcal{A}$. These expressions can be connected in any possible way, obtaining regular expressions still accepted by $\mathcal{H}_F$. We focus on strings accepted by $\mathcal{H}_F$: indeed, a string on the alphabet $\mathcal{A}$ corresponds biunivocally to a sequence of transitions of $\mathcal{H}$, i.e. to a composition of next-state functions.

Definition 3 *Given a DT hybrid automaton $\mathcal{H}$, let $S = a_1 \cdot a_2 \cdot \ldots \cdot a_\ell$, with $a_i \in \mathcal{A}$, be a string belonging to the regular language $\mathcal{L}$ accepted by $\mathcal{H}$ according to Definition 2:*

- *S corresponds to an* admissible control cycle *for the DT hybrid automaton $\mathcal{H}$ if and only if*
  (1) S is a cycle, i.e.[2] *$a_\ell = \Psi(q', q_0)$ with $q' \in \mathcal{Q}$;*
  (2) there exists $x_0 \in \mathbb{R}^n$ such that S can be executed[3] *by $\mathcal{H}$ if $\mathcal{H}$ is initialized at (q_0, x_0);*
- *to S is associated the matrix*[4]

$$A = \prod_{(q_m,q_n)=\Psi^{-1}(a_j),j=1,\ldots,l} F_{(q_m,q_n)} \;.$$

3.1 Stabilizing control actions

In this section, the stabilization problem for a given set of admissible control cycles is expressed by means of a DT version of the stable convex combination method (see [11]). Consider a linear DT hybrid automaton $\mathcal{H} = (\mathcal{C}, \mathcal{C}_0, \Sigma, M, \varphi, \mathcal{F})$ and let $\mathcal{L} = (E_1 + \ldots + E_s)^*$ be the regular language accepted by $\mathcal{H}$. Let $\mathcal{L}^k = (S_1 + \ldots + S_k)^*$ denote a regular sublanguage of $\mathcal{L}$, defined by strings $S_1 \ldots S_k$ corresponding to admissible control cycles, with $S_i = a_{i1} \ldots a_{i\ell_i}$ for $i = 1, \ldots, k$. Let $A_{i1} \ldots A_{i\ell_i}$ be the continuous dynamic transformation matrices associated to the symbols $a_{i1} \ldots a_{i\ell_i}$ and let $A_1 \ldots A_k$ be the matrices associated to $S_1 \ldots S_k$, according to Definition 3.
Given a matrix $P \in \mathbb{R}^{n\times n}$, with $P = P^T > 0$, define

$$\tilde{A}_i = \begin{bmatrix} A_i & 0 \\ 0 & 1 \end{bmatrix},\; \tilde{A}_{ij} = \begin{bmatrix} A_{ij} & 0 \\ 0 & 1 \end{bmatrix},\; \tilde{P} = \begin{bmatrix} P & 0 \\ 0 & 1 \end{bmatrix}.$$

Let $\mathcal{P}_c^k$ denote the LMI problem for positive coefficients $\delta_1 \ldots \delta_k$ and $\mu_{11} \ldots \mu_{1\ell_1} \ldots \mu_{k1} \ldots \mu_{k\ell_k}$ defined by

$$\textstyle\sum_{i=1}^k \delta_i \left(A_i^T P A_i - P\right) < 0 \tag{1}$$

$$\begin{cases} H_{i1} - \mu_{i1}\left(\tilde{A}_i^T \tilde{P} \tilde{A}_i - \tilde{P}\right) \le 0 \\ \tilde{A}_{i1}^T H_{i2} \tilde{A}_{i1} - \mu_{i2}\left(\tilde{A}_i^T \tilde{P} \tilde{A}_i - \tilde{P}\right) \le 0 \\ \vdots \\ \left(\tilde{A}_{i1}^T \ldots \tilde{A}_{i\ell_i-1}^T\right) H_{i\ell_i} \left(\tilde{A}_{i\ell_i-1} \ldots \tilde{A}_{i1}\right) - \\ \mu_{i\ell_i}\left(\tilde{A}_i^T \tilde{P} \tilde{A}_i - \tilde{P}\right) \le 0 \\ \text{for every } i \in \{1 \ldots k\} \end{cases} \tag{2}$$

where $H_{i1} \ldots H_{i\ell_i} \in \mathbb{R}^{(n+1)\times(n+1)}$ are defined by

$$G_{ij}(x) = \begin{bmatrix} x^T & 1 \end{bmatrix} H_{ij} \begin{bmatrix} x \\ 1 \end{bmatrix} \le 0,\; H_{ij} = \begin{bmatrix} W_{ij} & \frac{v_{ij}}{2} \\ \frac{v_{ij}^T}{2} & c_{ij} \end{bmatrix}.$$

If $\mathcal{P}_c^k$ has a solution, $\forall x \in \mathbb{R}^n$ there always exists at least an index $i \in \{1 \ldots k\}$, depending on x, such that $x^T \left(A_i^T P A_i - P\right) x < 0$ and the control cycle corresponding to A_i is admissible on x. This construction naturally induces a family of stabilizing feedback laws, among which a possible choice is

$$i = \operatorname{argmin}_j \left\{x^T \left(A_j^T P A_j - P\right) x\right\} \tag{3}$$

The following theorem establishes[5] the relation between $\mathcal{P}_c^k$ and Problem 1.

Theorem 4 *Let $\mathcal{L} = (E_1 + \ldots + E_s)^*$ be the regular language accepted by $\mathcal{H}$. Consider the regular language $\mathcal{L}^k = (S_1 + \ldots + S_k)^* \subseteq \mathcal{L}$, where $S_1 \ldots S_k$ are strings. If the problem $\mathcal{P}_c^k$ associated to $\mathcal{L}^k$ has a solution, then Problem 1 admits a solution.*

Notice that $\mathcal{P}_c^k$ is a conservative approximation of Problem 1. Nevertheless, to the best of our knowledge, we are not aware of any necessary and sufficient condition for the stabilizability of switched systems, except for the case of two control modes.

3.2 Stabilizability analysis

In this subsection, we present necessary conditions for the solvability of $\mathcal{P}_c^k$, as the index k increases. A necessary condition for $\mathcal{P}_c^k$ to be solvable is the solvability of the problem $\mathcal{P}^k$ defined by (1). Since $\mathcal{P}^k$ is obtained from $\mathcal{P}_c^k$ by removing guard conditions (2), if $\mathcal{P}^{k+1}$ is not solvable, neither is $\mathcal{P}_c^{k+1}$. The following lemma relates the solvability of problems $\mathcal{P}^k$ and $\mathcal{P}^{k+1}$:

Lemma 5 *Let $\mathcal{L} = (E_1 + \ldots + E_s)^*$ be the regular language accepted by $\mathcal{H}$. Consider two regular languages $\mathcal{L}^k = (S_1 + \ldots + S_k)^* \subseteq \mathcal{L}$ and $\mathcal{L}^{k+1} = (S_1 + \ldots + S_k + S_{k+1})^* \subseteq \mathcal{L}$, with S_i a succession of strings and $\mathcal{L}^k \subset \mathcal{L}^{k+1}$. Let $A_1 \ldots A_k, A_{k+1}$ be the corresponding matrices. If $\mathcal{P}^k$ has no solution, and if*

[2] Since $S \in \mathcal{L}$, then $a_1 = \Psi(q_0, q')$.

[3] Assuming $(q(0), x(0)) = (q_0, x_0)$, for $k = 0, \ldots, \ell$ we have $q(k+1) = \varphi(q(k), \sigma(k))$, with $\Psi(q(k), q(k+1)) = a_k$, for some $\sigma(k) \in M(q(k), x(k))$, and $x(k+1) = F_{(q(k),q(k+1))} x(k)$.

[4] The symbol $\prod$ stands for the matrix left–product.

[5] Proofs are omitted due to space limitation.

positive coefficients $\tau_1 \dots \tau_k$ *exist such that*

$$\sum_{i=1}^{k} \tau_i \left(A_i^T P A_i - P\right) - \left(A_{k+1}^T P A_{k+1} - P\right) \leq 0 \quad (4)$$

then $\mathcal{P}^{k+1}$ *has no solution.*

In fact, if (4) has a solution, it follows that $\forall x \in \mathbb{R}^n$ satisfying $x^T \left(A_{k+1}^T P A_{k+1} - P\right) x < 0$, there exists at least one $i \in \{1 \dots k\}$ such that $x^T \left(A_i^T P A_i - P\right) x < 0$. Consequently, the addition of the matrix A_{k+1} does not bring any benefit to the stabilizability of the system, with respect to the matrix P. We shall indicate this fact by writing $\mathcal{L}^{k+1} \sqsubseteq_P \mathcal{L}^k$.
The following theorem relates the solvability of problems $\mathcal{P}^{k'}$ and $\mathcal{P}^k$, in the general case $k > k'$ and provides a criterion for the choice of the matrix P.

Theorem 6 *Let* $\mathcal{L} = (E_1 + \dots + E_s)^*$ *be the regular language accepted by* $\mathcal{H}$*. Consider the succession of regular languages* $\{\mathcal{L}^k\}$ *defined by* $\mathcal{L}^k = (S_1 + \dots + S_p + S_{s_0} + \dots + S_{s_{k-1}} + S_{s_k})^* \subseteq \mathcal{L}$*, where* $S_1 \dots S_p$ *are strings and* $S_{s_h} = \alpha\beta^h\gamma$*, for* $h = 0, \dots, k$*, with* α*,* β *and* γ *strings. Let* $A_{s_h} = A_\gamma A_\beta^h A_\alpha$ *be the matrix associated to* S_{s_h}*. Assume that* $det(A_\alpha) \neq 0$*. If* $\bar{k} > 0$ *exists such that one of the following conditions holds:*

- $\exists \delta \in (0, 1]$, $\exists P \in \mathbb{R}^{n\times n}$, $P = P^T > 0$, *such that*

$$\begin{cases} \left(A_{s_{\bar{k}-1}}^T P A_{s_{\bar{k}-1}} - P\right) - \delta\left(A_{s_{\bar{k}}}^T P A_{s_{\bar{k}}} - P\right) \leq 0 \\ \left(A_\alpha^T A_\beta^T A_\alpha^{-T}\right) P \left(A_\alpha^{-1} A_\beta A_\alpha\right) - P \leq 0 \end{cases} \quad (5)$$

  and A_β *is stable*

- $\exists \delta > 1$, $\exists P \in \mathbb{R}^{n\times n}$, $P = P^T > 0$, *such that*

$$\begin{cases} \left(A_{s_{\bar{k}-1}}^T P A_{s_{\bar{k}-1}} - P\right) - \delta\left(A_{s_{\bar{k}}}^T P A_{s_{\bar{k}}} - P\right) \leq 0 \\ \left(A_\alpha^T A_\beta^T A_\alpha^{-T}\right) P \left(A_\alpha^{-1} A_\beta A_\alpha\right) - P \geq 0 \end{cases} \quad (6)$$

  and A_β *is unstable*

then $\mathcal{L}^k \sqsubseteq_P \mathcal{L}^{k-1}$*,* $\forall k \geq \bar{k}$*.*

The following corollaries are useful to deal with regular expressions E_i containing more than one not nested '*' operator. This occurs when two or more distinct cycles are present on the same control sequence.

Corollary 7 *Let* $\mathcal{L} = (E_1 + \dots + E_s)^*$ *be the regular language accepted by* $\mathcal{H}$*. Consider the successions of regular languages* $\{\mathcal{L}_1^k\}$*,* $\{\mathcal{L}_2^k\}$ *defined by*

$$\mathcal{L}_1^k = (S_1 + \dots + S_p + S_{s_0}^1 + \dots + S_{s_{k-1}}^1 + S_{s_k}^1)^* \subseteq \mathcal{L}$$
$$\mathcal{L}_2^k = (S_1 + \dots + S_p + S_{s_0}^2 + \dots + S_{s_{k-1}}^2 + S_{s_k}^2)^* \subseteq \mathcal{L}$$

where $S_1 \dots S_p$ *are strings,* $S_{s_h}^1 = \alpha_1\beta^h\gamma$ *and* $S_{s_h}^2 = \alpha_2\beta^h\gamma$*, with* α_1*,* α_2*,* β *and* γ *strings. Associated to* $S_{s_h}^1$ *and* $S_{s_h}^2$ *are the matrices* $A_\gamma A_\beta^h A_{\alpha_1}$ *and* $A_\gamma A_\beta^h A_{\alpha_2}$*, respectively. Assume that* $det(A_{\alpha_1}) \neq 0$ *and* $det(A_{\alpha_2}) \neq 0$*. Let* $P_1 > 0$ *be a matrix satisfying (5–6) for all* $\mathcal{L}_1^k$*, with* $k \geq k'$ *for some* k'*, and let* $P_2 = \left(A_{\alpha_2}^T A_{\alpha_1}^{-T}\right) P_1 \left(A_{\alpha_1}^{-1} A_{\alpha_2}\right)$*. If either* A_β *is stable and* $P_2 - P_1 \leq 0$*, or* A_β *is unstable and* $P_2 - P_1 \geq 0$*, then* $\mathcal{L}_2^{k+1} \sqsubseteq_{P_1} \mathcal{L}_2^k$*,* $\forall k \geq k'$*.*

Corollary 8 *Let* $\mathcal{L} = (E_1 + \dots + E_s)^*$ *be the regular language accepted by* $\mathcal{H}$*. Consider the successions of regular languages* $\{\mathcal{L}_1^k\}$*,* $\{\mathcal{L}_2^k\}$ *defined by*

$$\mathcal{L}_1^k = (S_1 + \dots + S_p + S_{s_0}^1 + \dots + S_{s_{k-1}}^1 + S_{s_k}^1)^* \subseteq \mathcal{L}$$
$$\mathcal{L}_2^k = (S_1 + \dots + S_p + S_{s_0}^2 + \dots + S_{s_{k-1}}^2 + S_{s_k}^2)^* \subseteq \mathcal{L}$$

where $\{S_1 \dots S_p\}$ *are strings,* $S_{s_h}^1 = \alpha\beta^h\gamma_1$ *and* $S_{s_h}^2 = \alpha\beta^h\gamma_2$*, with* α*,* β*,* γ_1 *and* γ_2 *strings. Associated to* $S_{s_h}^1$ *and* $S_{s_h}^2$ *are the matrices* $A_{\gamma_1} A_\beta^h A_\alpha$ *and* $A_{\gamma_2} A_\beta^h A_\alpha$*, respectively. Assume that* $det(A_{\gamma_1}) \neq 0$ *and* $det(A_{\gamma_2}) \neq 0$*. Let* $P_1 > 0$ *be a matrix satisfying (5–6) for all* $\mathcal{L}_1^k$*, with* $k \geq k'$ *for some* k'*, and let* $P_2 = \left(A_{\gamma_2}^{-T} A_{\gamma_1}^T\right) P_1 \left(A_{\gamma_1} A_{\gamma_2}^{-1}\right)$*. If either* A_β *is stable and* $P_2 - P_1 \geq 0$*, or* A_β *is unstable and* $P_2 - P_1 \leq 0$*, then* $\mathcal{L}_2^{k+1} \sqsubseteq_{P_2} \mathcal{L}_2^k$*,* $\forall k \geq k'$*.*

4 Exploration algorithm

In this section, an algorithm for the exploration of sequences of discrete control actions solving Problem 1 is proposed. The development is limited to the case, of interest in our engine control application, in which all the expressions E_i in the regular language $\mathcal{L} = (E_1 + \dots + E_S)^*$ do not contain nested '*' operators. Assume that each $E \in \{E_1, \dots, E_S\}$ is of the form

$$E = Y_1^E \cdots Y_{\ell_E}^E \text{ with } Y_j^E = \alpha_j(\beta_j)^*\gamma_j \quad \text{for } j = 1 \dots \ell_E$$

where α_j, β_j and γ_j are strings (possibly equal to ϵ) to which linear transformations A_{α_j}, A_{β_j}, A_{γ_j} are associated. Given a regular expression E, by specifying a vector $k^E = (k_1^E, \dots, k_{\ell_E}^E) \in \mathbb{N}^{\ell_E}$, the string

$$S^E(k^E) = \alpha_1\beta_1^{k_1^E}\gamma_1 \cdots \alpha_{\ell_E}\beta_{\ell_E}^{k_{\ell_E}^E}\gamma_{\ell_E} \quad (7)$$

is defined. Furthermore, to each Y_j^E in E is assigned a prefix $\alpha_1\beta_1^{k_1^E}\gamma_1 \cdots \alpha_{j-1}\beta_{j-1}^{k_{j-1}^E}\gamma_{j-1}\alpha_j$, a suffix $\gamma_j\alpha_{j+1}\beta_{j+1}^{k_{j+1}^E}\gamma_{j+1} \cdots \alpha_{\ell_E}\beta_{\ell_E}^{k_{\ell_E}^E}\gamma_{\ell_E}$, and the corresponding matrices

$$\begin{aligned} \bar{A}_{E\alpha}^{(j)}(k_1^E \dots k_{j-1}^E) \;=\; & A_{\alpha_j}(A_{\gamma_{j-1}}A_{\beta_{j-1}}^{k_{j-1}^E}A_{\alpha_{j-1}}) \cdots \quad (8) \\ & (A_{\gamma_2}A_{\beta_2}^{k_2^E}A_{\alpha_2}) \cdot (A_{\gamma_1}A_{\beta_1}^{k_1^E}A_{\alpha_1}) \end{aligned}$$

$$\bar{A}^{(j)}_{E\gamma}(k^E_{j+1}\dots k^E_{\ell_E}) \;=\; (A_{\gamma_{\ell_E}} A^{k_{\ell_E}}_{\beta_{\ell_E}} A_{\alpha_{\ell_E}})\cdots (A_{\gamma_{j+1}} A^{k^E_{j+1}}_{\beta_{j+1}} A_{\alpha_{j+1}}) A_{\gamma_j} \quad (9)$$

Let $(k^{E_1}\dots k^{E_S}) \equiv (k^{E_1}_1 \dots k^{E_1}_{\ell_1}, k^{E_2}_1 \dots k^{E_S}_{\ell_S}) \in \mathbb{N}^{\Sigma \ell_{E_i}}$ be the collections of vectors $k^{E_1}\cdots k^{E_S}$. Introduce the regular language $\mathcal{L}^{(k^{E_1}\dots k^{E_S})} \subseteq \mathcal{L}$ obtained from $\mathcal{L}$ by replacing the regular expressions E_i by the strings $S^{E_i}(h^{E_i})$ defined in (7), with $h^{E_i} = (h^{E_i}_1, \dots, h^{E_i}_{\ell_{E_i}})$ and $0 \le h^{E_i}_j \le k^{E_i}_j$. Let $\mathcal{P}^{(k^{E_1}\dots k^{E_S})}_c$ denote the problem (1–2) defined on the regular language $\mathcal{L}^{(k^{E_1}\dots k^{E_S})}$ and $\mathcal{P}^{(k^{E_1}\dots k^{E_S})}$ the corresponding unconstrained problem.

The proposed exploration algorithm is reported in Figure 1. The algorithm performs an implicit visit of the languages $\mathcal{L}^{(k^{E_1}\dots k^{E_S})} \subseteq \mathcal{L}$, defined by $(k^{E_1}\dots k^{E_S})$ with entries k^E_j upper bounded by a maximum value $K^{\max}$, given as input parameter. The solutions are returned in the set $\mathcal{S}$, as pairs of a vector $(k^{E_1}\dots k^{E_S})$ defining the language and the corresponding Lyapunov matrix P^E_j. The idea that lies behind this algorithm is that of determining, for every regular expression $Y^E_j = \alpha_j(\beta_j)^*\gamma_j$, the maximum value up to which it is useful, or allowed, to push the exponent h^E_j of β_j in order to solve problem $\mathcal{P}_c$. Variable K^E_j represents the maximun exponent for β_j in Y^E_j currently assessed. K^E_j is initialized to $K^{\max}$ and is reduced each time a language inclusion condition is verified. At each iteration, a matrix P^E_j is chosen such that $P^E_j = (P^E_j)^T > 0$ and

$$({A^{(j)}_{E\alpha}}^T {A_{\beta_j}}^T {A^{(j)}_{E\alpha}}^{-T}) P^E_j ({A^{(j)}_{E\alpha}}^{-1} A_{\beta_j} A^{(j)}_{E\alpha}) - P^E_j \quad (10)$$

is not indefinite, where $A^{(j)}_{E\alpha} = \bar{A}^{(j)}_{E\alpha}(k^E_1 \dots k^E_{j-1})$ is as in (8), with $k^E_i = L$ for $i = 1, \dots, j-1$. It is assumed that matrices (8–9) are not singular. Then, the problem $\mathcal{P}^{(k^{E_1}\dots k^{E_S})}|^{k^E_j=h}_{\star=L}$, defined as $\mathcal{P}^{(k^{E_1}_1 \dots k^E_j \dots k^{E_S}_{\ell_S})}$ with $k^E_j = h$ and $k^{E_r}_i = L$ for $E_r \neq E$ or $i \neq j$, is evaluated for increasing h, i.e. increasing power of β_j. If it is solvable for some h, then the corresponding constrained problem $\mathcal{P}^{(k^{E_1}\dots k^{E_S})}_c|^{k^E_j=h}_{\star=L}$ is tested and, if the latter is solvable, the corresponding $(k^{E_1}\cdots k^{E_S})$ vector is stored in $\mathcal{S}$. Hence, if the unconstrained problem $\mathcal{P}^{(k^{E_1}\dots k^{E_S})}|^{k^E_j=h}_{\star=L}$ is not solvable, for h up to K^E_j, then Theorem 6 is tested on

$$\mathcal{L}^h = \mathcal{L}^{(k^{E_1}\dots k^{E_S})}|^{k^E_j=h}_{\star=L},\ \mathcal{L}^{h+1} = \mathcal{L}^{(k^{E_1}\dots k^{E_S})}|^{k^E_j=h+1}_{\star=L} \quad (11)$$

and, if the languages inclusion condition (5–6) is verified, then Corollaries 7 and 8 are evaluated on $\{\mathcal{L}^h_1\}$, with $\mathcal{L}^h_1$ as in (11), and $\{\mathcal{L}^h_2\}$ with

$$\mathcal{L}^h_2 = \mathcal{L}^{(k^{E_1}\dots k^{E_S})}|^{k^E_j=h}_{\star=m} \text{ with } m = L, \dots, K^{E_r}_i \quad (12)$$

to cut off from the search space exponents for β_j in $Y^E_j = \alpha_j(\beta_j)^*\gamma_j$ greater than h.

$\mathcal{S}$ = Explore($K^{\max}$)
  $\mathcal{S} = \{\}$;
  for each $E \in \{E_1 \dots E_S\}$ and $Y^E_j \in \mathcal{Y}$
    $K^E_j = K^{\max}$;
  end
  for $L = 1 \dots K^{\max}$
    if ($L > \min_{E,j} K^E_j$) then break; end
    for each $E \in \{E_1 \dots E_S\}$ and $Y^E_j \in \mathcal{Y}$
      compute P^E_j as in (10);
      $h = L$;
      while ($\mathcal{P}^{(k^{E_1}\dots k^{E_S})}|^{k^E_j=h}_{\star=L}$ is solvable for P^E_j) do
        if ($\mathcal{P}^{(k^{E_1}\dots k^{E_S})}_c|^{k^E_j=h}_{\star=L}$ is solvable for P^E_j) then
          $\mathcal{S} = \mathcal{S} \cup \left[(k^{E_1}\cdots k^{E_S})|^{k^E_j=h}_{\star=L}, P^E_j \right]$;
        end
        if ($h = K^E_j$) then break; else $h = h + 1$; end
      end
      if ($h < K^E_j$) then
        if (Theorem 6 is verified on (11)) then
          if (Corollaries 7 and 8 hold for (12)) then
            $K^E_j = h$;
          end
        end
      end
    end
  end
end

Figure 1: Exploration algorithm.

5 Engine control application

The motivating problem for this research was the design of feedback control for the stabilization of an automotive engine on a given set point. In Figure 2 it is depicted a DT hybrid automaton representing the torque generation mechanism and the powertrain dynamics of a 4–cylinder in–line engine, which has been obtained from the discretization of a hybrid engine model described in [2] and [1]. The automaton has four locations $\{q_1, \dots, q_4\}$. Automaton transitions are synchronous with continuous state updates and depend on the discrete events σ_i. The three discrete controlled events σ_-, σ_0 and σ_+, represent the choice of three different values of spark advance[6], namely a negative spark advance ϕ_-, the zero spark advance and a positive spark advance ϕ_+. Transitions (q_2, q_4) and (q_3, q_1) are fired by the internal event σ_{dc}, that occurs when one of the four pistons reaches a top dead center. Notice that these transitions are the only exit transitions from locations q_2 and q_3. The DT hybrid automaton has a four-dimensional continuous state x, with components

[6] Specifying different spark ignition times, see [2] for details.

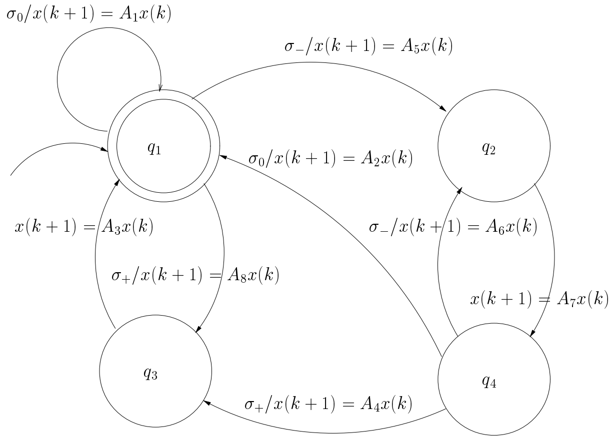

Figure 2: DT hybrid automaton of the engine.

$n - n_0$, $T - T_0$, $m_c - m_{c0}$, and $m_e - m_{e0}$, where: T is the generated torque; n is the crankshaft revolution speed; m_c and m_e are the masses of air loaded in the cylinder that is, respectively, in the compression and in the expansion stroke; and n_0, T_0, m_{c0}, m_{e0} are the corresponding values at set point. At each $q_i \to q_j$ transition of the automaton, the continuous state is updated according to a given linear next–state function $f_{(q_i,q_j)}(x(k), u(k)) = F_{(q_i,q_j)}x(k) + b_{(q_i,q_j)}u(k)$, where the continuous input $u(k)$ represents the mass of air $m_a(k)$ to be loaded in the cylinders[7].

In the sequel, the design of a stabilizing feedback controller using the technique presented in Sections 3 and 4 is illustrated. The requested mass of air $m_a(k)$ is obtained by linear feedbacks $m_a(k) = K_{(q_i,q_j)}x(k)$, where parameters $K_{(q_i,q_j)}$ depends on the automaton transition. In Figure 2, matrices A_i represent the closed–loop matrices so obtained. Let q_1 be the unique initial/accepting location and let us associate to each transition with dynamic matrix A_i the label a_i. The corresponding accepted regular language is $\mathcal{L} = (a_1 + a_5(a_7a_6)^*a_7a_2 + a_5(a_7a_6)^*a_7a_4a_3 + a_8a_3)^* = (E_1 + E_2 + E_3 + E_4)^*$, where $E_1 = a_1$, $E_2 = \alpha_2(\beta_2)^*\gamma_2$, $E_3 = \alpha_3(\beta_3)^*\gamma_3$, and $E_4 = a_8a_3$, with $\alpha_2 = a_5$, $\beta_2 = a_7a_6$, $\gamma_2 = a_7a_2$, $\alpha_3 = a_5$, $\beta_3 = a_7a_6$, $\gamma_3 = a_7a_4a_3$. Invertibility of matrices (8) is verified for feedback gain matrices K_5 such that $det(A_5) \neq 0$. Consider, for any $h_2, h_3 \geq 0$, the regular language $\mathcal{L}^{(h_2,h_3)} = (a_1 + a_5(a_7a_6)^{h_2}a_7a_2 + a_5(a_7a_6)^{h_3}a_7a_4a_3 + a_8a_3)^*$. It is easy to see that, since the regular expressions E_2 and E_3 are not concatenated, both Corollaries 7 and 8 are satisfied for every possible pair of languages $(\mathcal{L}^{(h_2^1,h_3^1)}, \mathcal{L}^{(h_2^2,h_3^2)})$, and invertibility of matrices (9) is not required. Running the exploration algorithm reported in Figure 1, for maximum depth $k_{max} = 20$, the first language tested is $\mathcal{L}^{(0,0)} = (a_1 + a_5a_7a_2 + a_5a_7a_4a_8 + a_3a_8)^*$. The first solutions found are $\mathcal{L}^{(2,0)}$, $\mathcal{L}^{(1,1)}$, $\mathcal{L}^{(2,1)}$. The obtained solutions have been used to design a hybrid controller that implements the control law (3). Then, this controller has been connected to a model of the engine and the closed-loop system has been simulated.

[7] In engines equipped with an electronic throttle valve, an intake manifold controller is in charge of providing the requested air mass for each intake stroke.

6 Conclusions

We presented a novel approach to the stabilization of linear discrete–time hybrid automata based on the stable convex combinations method proposed for switched systems. To apply this approach, we identified a set of control actions for hybrid automata that share the same characteristics as the ones for switched systems. These control actions correspond to cyclic paths in the graph of the automaton. Since the number of candidate control actions may be high, we also give sufficient conditions for the elimination of several paths that are dominated by others, based on which an exploration algorithm has been proposed. The method has been tested on an industrial problem in automotive engine control that motivated this research. Future work will focus on extending the approach to linear discrete–time hybrid automata with continuous inputs.

References

[1] A. Balluchi, L. Benvenuti, M.D. Di Benedetto, and A.L. Sangiovanni-Vincentelli. Idle speed controller synthesis using an assume–guarantee approach. In R. Johansson and A. Rantzer, editors, *Nonlinear and Hybrid Systems in Automotive Control*, pages 229–243. Springer, 2003.

[2] A. Balluchi, L. Benvenuti, M. D. Di Benedetto, C. Pinello, and A. L. Sangiovanni-Vincentelli. Automotive engine control and hybrid systems: Challenges and opportunities. *Proceedings of the IEEE*, 88, "Special Issue on Hybrid Systems" (7):888–912, July 2000.

[3] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraint. *Automatica*, 35(3):407–427, March 1999.

[4] M. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. on Automatic Control*, 43(4):475–482, 1998.

[5] F.A. Cuzzola and M. Morari. A generalized approach for analysis and control of discrete–time piecewise affine and hybrid systems. In M.D. Di Benedetto and A.L. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, vol. 2034 of *Lecture Notes in Computer Science*, pages 189–203. Springer-Verlag, Roma, I, 2001.

[6] Bo Hu, Xuping Xu, A. N. Michel, and P.J. Antsaklis. Stability analysis for a class of nonlinear

switched system. In *Proceedings of the 38th IEEE Conference on Decision and Control*, vol. 5, pages 4374–9, Phoenix, AZ, USA, December 1999.

[7] J. Ullman J. Hopcroft, R. Motwani, editor. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 2001.

[8] M. Johansson and A. Rantzer. Computation of piecewise quadratic lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):555–9, April 1998.

[9] X.D. Koutsoukos and P.J. Antsaklis. Characterization of stabilizing switching sequences in switched linear sytems using piecewise linear laypunov functions. In M.D. Di Benedetto and A.L. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, vol. 2034 of *Lecture Notes in Computer Science*, pages 347–360. Springer-Verlag, Roma, I, 2001.

[10] A.S. Morse. Supervisory control of a families of linear set-point controllers – part 1: Exact matching. *IEEE Trans. on Automatic Control*, 41(10):1413–1431, October 1996.

[11] M. A. Wicks, P. Peleties, and R. DeCarlo. Construction of piecewise lyapunov functions for stabilizing switched systems. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, pages 3492–3497, Lake Buena Vista, FL, USA, 1994.