Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems – MTNS 2010 • 5–9 July, 2010 • Budapest, Hungary

Temporal logic control for piecewise-affine hybrid systems on polytopes

L,C.G.J.M. Habets and C. Belta

***Abstract*— In this paper, a method is proposed for the design of control laws for hybrid systems with continuous inputs. The objective is to influence their behavior in such a way that the discrete component of the closed-loop system satisfies a given condition, described by a temporal logic formula. For this purpose, a transition system is constructed, by abstracting from the continuous dynamics of the hybrid system. It is shown that a controller for this transition system, realizing the given control objective, corresponds to a controller for the original hybrid system, realizing the same objective, and vice versa.**

I. Introduction

Hybrid systems are dynamical systems with interacting discrete-event and continuous dynamics. Many control systems have a hybrid character, and examples of hybrid control systems are abundantly available, ranging from car engines, air traffic and robot planning control to electric power networks.

Within the field of hybrid systems the class of piecewise-affine hybrid systems on polytopes has been studied quite extensively. In the literature different approaches have been proposed for the design of controllers for this class of systems. One of them is based on the idea of control-to-facet ([8]). Assuming that the discrete switching is determined by the facet through which the continuous state leaves a polytope, the continuous input can be used to influence the behavior of the discrete dynamics. In [9] this approach was used to solve a reachability problem for hybrid systems with safety constraints.

This paper may be considered as an extension of [9] in two directions: (1) instead of a reach-avoid condition, the goal is to achieve a control objective stated as a linear temporal logic formula over the discrete modes of the hybrid system, (2) instead of fixing the applied continuous feedback in each discrete mode, a new dynamic hybrid feedback mechanism is studied that allows the application of different continuous feedbacks in one discrete mode. A crucial step is the construction of a discrete transition system from a hybrid system, by abstracting the continuous dynamics. It is shown that there exists a control automaton that realizes the required control objective for this transition system if and only if there exists a so-called feedback control automaton that realizes the same objective for the original hybrid system. Furthermore, it is shown how this hybrid feedback mechanism can be obtained from the control automaton for the discrete transition system. In the paper, the main emphasis is on the rigorous definition of hybrid systems, hybrid automata and their semantics, and on the introduction of the fundamental notion of feedback control automata. The proofs of the results are omitted and will be published elsewhere.

L. Habets is with the Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, P.O. Box 513, NL - 5600 MB Eindhoven, The Netherlands, and is part-time affiliated with CWI - Centrum voor Wiskunde en Informatica, P.O. Box 94079, NL - 1090 GB Amsterdam, The Netherlands. E-mail: l.c.g.j.m.habets@tue.nl

C. Belta is with the Center for Information and Systems Engineering, Boston University, 15 Saint Mary's Street, Brookline, MA 02446, USA. E-mail: cbelta@bu.edu

II. Definition and semantics of hybrid automata

For the study and design of feedback control mechanisms for hybrid systems with continuous inputs, one first needs a formal definition of a hybrid system without inputs. In this section we introduce these so-called hybrid automata (see e.g. [11]), and describe their behavior, both in terms of hybrid trajectories and in terms of the words generated by these trajectories.

Throughout this paper we use the notation $\mathbb{N}$ for the natural numbers without 0, and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ for the natural numbers, including 0.

Definition 2.1: A hybrid automaton with arbitrary continuous initial states is a tuple

$$\mathcal{HA} = (Q, Q_0, T, N, \mathit{Inv}, G, \mathcal{S}), \tag{1}$$

where

- Q is a finite set of discrete states (also called modes, or locations),
- Q_0 denotes the set of initial discrete states,
- $T \subseteq Q \times Q$ is a set of discrete transitions,
- $N : Q \to \mathbb{N}$ is a map giving the dimension N_q of the continuous state in each mode $q \in Q$,
- $\mathit{Inv} : Q \to 2^{\mathbb{R}^{N_q}}$ is the invariant map, where $\mathit{Inv}(q)$ is a full dimensional closed convex subset of $\mathbb{R}^{N_q}$,
- G is a guard that associates a subset of $\partial \mathit{Inv}(q)$ to each transition $(q, q') \in T$,
- $\mathcal{S} = (h_q)_{q \in Q}$ is an assignment of Lipschitz continuous vector fields $h_q : \mathrm{Dom}_q \to \mathbb{R}^{N_q}$, $(q \in Q)$, with for all $q \in Q$, Dom_q an open set that contains $\mathit{Inv}(q)$.

Informally, a hybrid automaton is an automaton with a continuous-time autonomous system at each discrete mode q, governed by the differential equation $\dot{x}_q(t) = h_q(x_q(t))$. This differential equation remains valid as long as $x_q(t)$ is an element of the invariant set $\mathit{Inv}(q)$. Exactly at the time-instant that the continuous state x_q leaves $\mathit{Inv}(q)$ by crossing a guard $G(q, q')$, the discrete mode switches to q', and the continuous state x'_q is restarted in an arbitrary point $x_{q',0} \in \mathit{Inv}(q')$ and continues to evolve according to differential equation $\dot{x}_{q'}(t) = h_{q'}(x_{q'}(t))$.

The *departure set* $\mathit{Dept}(P, h)$ of a dynamical system on a convex set P consists of all points in ∂P through which the

ISBN 978-963-311-370-7

continuous state x may leave the state set P:

$$\begin{aligned} \mathit{Dept}(P,h) \quad = \quad & \{x_0 \in \partial P \mid \exists \varepsilon > 0 \text{ s.t. for all } t \in (0,\varepsilon) \\ & \text{the solution of } \dot{x}(t) = h(x(t)), \text{ with} \\ & x(0) = x_0 \text{ satisfies } x(t) \notin P\}. \end{aligned} \tag{2}$$

So, in mode q of a hybrid automaton, $\mathit{Dept}(\mathit{Inv}(q), h_q)$ is the corresponding departure set. In order to guarantee that a hybrid automaton is *non-blocking* all points in this departure set must belong to a guard (see e.g. [11]):

$$\forall q \in Q \ \forall x \in \mathit{Dept}(\mathit{Inv}(q), h_q) \ \exists q' \in Q \text{ s.t. } x \in G(q,q'). \tag{3}$$

In the sequel we will only consider non-blocking hybrid automata. However, by abuse of terminology we will often omit the adjective "non-blocking", and just speak of hybrid automata. Note that a non-blocking hybrid automaton may be non-deterministic: if a point $\hat{x} \in \mathit{Dept}(\mathit{Inv}(q), h_q)$ belongs to more than one guard, e.g. $\hat{x} \in G(q,q')$ and $\hat{x} \in G(q,q'')$, then both the transition to q' and q'' are enabled when the continuous state x_q leaves $\mathit{Inv}(q)$ through the point $\hat{x}$.

Next we specify the semantics of a hybrid automaton in a more formal way. Since in none of the discrete modes the continuous initial state is specified, we combine the behavioral approach (in the fashion of Willems, see e.g. [12], [13]) with the notion of hybrid time trajectories ([11], [5] and [6]) to describe the set of all hybrid state trajectories of a hybrid automaton.

Definition 2.2: Let Q be a finite set of discrete states, and let $\mathit{Inv}(q)$, $(q \in Q)$, be the closed convex set of continuous states corresponding to q. A *hybrid trajectory* is a finite or infinite sequence

$$\rho = (q_i, t_i, x_i)_{i=0}^n,$$

(i.e. either $n \in \mathbb{N}_0$ or $n = \infty$), with $q_i \in Q$, $(t_i)_{i=0}^n$ a non-decreasing sequence in $\mathbb{R}$, and $x_i : [t_i, t_{i+1}] \to \mathit{Inv}(q_i)$ a trajectory of the continuous state at location q_i. The number n, which may be infinite, is the number of discrete transitions in ρ. If n is finite, then $x_n : [t_n, \infty) \to \mathit{Inv}(q_n)$.

A hybrid trajectory ρ provides an exact description of both the evolution of the discrete mode and of the corresponding continuous state of one particular execution of a hybrid automaton. The set of all possible executions of a hybrid automaton is called its behavior:

Definition 2.3: The *behavior* $\mathcal{B}(\mathcal{HA})$ of a hybrid automaton $\mathcal{HA}$ is the set of *all* hybrid trajectories ρ that satisfy the following conditions:

(i) $q_0 \in Q_0$,

(ii) For all $0 \le i < n$:
1. x_i is continuous on $[t_i, t_{i+1}]$ and $\dot{x}_i(t) = h_{q_i}(x_i(t))$ for all $t \in (t_i, t_{i+1})$,
2. $x_i(t_{i+1}) \in \mathit{Dept}(\mathit{Inv}(q_i), h_{q_i}) \cap G(q_i, q_{i+1})$,

(iii) If the number n of discrete transitions in ρ is finite (i.e. $n \in \mathbb{N}_0$), then x_n is continuous on $[t_n, \infty)$ and $\dot{x}_n(t) = h_{q_n}(x_n(t))$ for all $t \in (t_n, \infty)$.

The time instants $\{t_i \mid i = 1, \ldots, n\}$ are the switching times, i.e. the time instants on which a discrete transition occurs. If n is finite, then condition *(iii)* implies that $x_n(t)$ remains in $\mathit{Inv}(q_n)$ forever. It may happen that $t_{i+1} = t_i$ for some $i \in \mathbb{N}_0$. In that case there is a switch to mode q_i at time t_i, but the corresponding continuous state is started in a point $x_i(t_i) \in \partial \mathit{Inv}(q_i)$ with the property that $x_i(t_i) \in \mathit{Dept}(\mathit{Inv}(q_i), h_{q_i}) \cap G(q_i, q_{i+1})$. So at the same time-instant $t_{i+1} = t_i$ the hybrid automaton will switch to state q_{i+1}. In this case, the discrete mode and corresponding continuous state at time $t_i = t_{i+1}$ is not specified in a unique way.

In this paper we do not study the full behavior of a hybrid automaton but confine ourselves to specifications described in terms of the discrete locations that are visited by these hybrid trajectories. For this we have to make a distinction between locations $q \in Q$ that are left in finite time, and locations $q \in Q$ for which the continuous state remains in $\mathit{Inv}(q)$ forever. Let Q_{inv} be a set of symbols of the same cardinality as Q, such that $Q \cap Q_{\text{inv}} = \varnothing$, and let $\xi : Q \longrightarrow Q_{\text{inv}}$ be a bijective mapping. We define $Q_{\text{tot}} := Q \cup Q_{\text{inv}}$. For any finite set S, we denote by $|S|$ the cardinality of S, by 2^S the set of subsets of S, and by S^ω the set of infinite sequences with elements from S.

Definition 2.4: The *word* $w(\rho)$ generated by a hybrid trajectory ρ is an infinite sequence $w = (w_i)_{i \in \mathbb{N}_0} \in Q_{\text{tot}}^\omega$, defined in the following way:

(1) If $\rho = (q_i, t_i, x_i)_{i=0}^n$ is a *finite* sequence, then $w_i := q_i$ for $i \le 0 \le n$ and $w_i := \xi(q_n)$ for $i > n$.

(2) If $\rho = (q_i, t_i, x_i)_{i=0}^\infty$ is an *infinite* sequence, then $w_i := q_i$ for all $i \in \mathbb{N}_0$.

So the word generated by a hybrid trajectory is simply the enumeration of the modes visited by that trajectory, with infinitely many repetitions of the symbol $\xi(q_n)$ if the mode q_n is reached and then never left.

Definition 2.5: The set of all words that are generated by the hybrid trajectories in the behavior $\mathcal{B}(\mathcal{HA})$ of hybrid automaton $\mathcal{HA}$ is called the *language* generated by $\mathcal{HA}$, and is denoted by

$$\mathcal{L}(\mathcal{HA}) := \{w(\rho) \mid \rho \in \mathcal{B}(\mathcal{HA})\}. \tag{4}$$

The specifications that we want to accommodate in this paper are "rich" temporal and logic statements about the reachability of discrete locations (modes) $q \in Q$ by the hybrid trajectories of a hybrid automaton $\mathcal{HA}$. In particular, we focus on specifications given as formulas of a fragment of Linear Temporal Logic [4] over the set of symbols Q_{tot}. For simplicity, we will call this logic LTL throughout the paper. Informally, such formulas are recursively defined by using the standard Boolean operators (*e.g.*, $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction)) and temporal operators, which include $\mathcal{U}$ ("until"), $\Box$ ("always"), $\Diamond$ ("eventually"). Such formulas are interpreted over the words in Q_{tot}^ω, as generated by the hybrid automaton $\mathcal{HA}$, according to Definition 2.4. If ϕ_1 and ϕ_2 are two LTL formulas over Q_{tot}, formula $\phi_1 \mathcal{U} \phi_2$ intuitively means that (over some word) ϕ_2 will eventually become true and ϕ_1 is true until this happens. For an LTL

formula ϕ, formula $\Diamond\phi$ means that ϕ becomes eventually true, whereas $\Box\phi$ indicates that ϕ is true at all positions of a word. More expressiveness can be achieved by combining the mentioned operators. For example, $\Diamond\Box\phi$ means that ϕ will eventually become true and then remains true forever, while $\Box\Diamond\phi$ means that ϕ is true infinitely often.

Definition 2.6: A hybrid automaton $\mathcal{HA}$ with discrete state set Q is said to satisfy an LTL formula ϕ over Q_{tot}, if all words $w \in \mathcal{L}(\mathcal{HA})$ satisfy ϕ.

Remark 2.7: In the literature, the definition of a hybrid automaton usually contains a *reset map*, that describes the relation between the final continuous state in the old discrete mode and the initial continuous state in the new discrete mode. So, the corresponding behaviors and languages are more restrictive, and an LTL formula is more easily satisfied. Unfortunately, incorporation of reset maps often leads to decidability problems. In the approach presented in this paper undecidable problems are avoided at the cost of obtaining more conservative results.

III. Piecewise-affine hybrid systems and feedback control automata

The main object of study in this paper are hybrid systems with continuous inputs. The main difference with hybrid automata is that the continuous input may be used to influence the evolution of the hybrid trajectories. For this purpose a so-called feedback control automaton is introduced. In closed-loop with a hybrid system, this yields a hybrid automaton. Given an LTL-formula ϕ, the problem is to construct a feedback control automaton that guarantees that the language generated by the controlled hybrid system satisfies ϕ.

Definition 3.1: A *piecewise-affine hybrid system over polytopes*, with m continuous inputs and arbitrary continuous initial states is a tuple

$$\mathcal{HS} = (Q, Q_0, N, \mathit{Inv}, \mathit{Fac}, \mathcal{S}, U), \tag{5}$$

where:

- Q is a finite set of discrete states,
- Q_0 denotes the set of initial discrete states,
- $N : Q \to \mathbb{N}$ is a map giving the dimension N_q of the continuous state in each mode $q \in Q$,
- $\mathit{Inv} : Q \to 2^{\mathbb{R}^{N_q}}$ is the invariant map, where $\mathit{Inv}(q)$ is a closed full dimensional *polytope* in $\mathbb{R}^{N_q}$,
- $\mathit{Fac} = (\mathit{Fac}_q)_{q\in Q}$ is an assignment of maps $\mathit{Fac}_q : \mathcal{FT}(\mathit{Inv}(q)) \to Q$, $q \in Q$, where $\mathcal{FT}(\mathit{Inv}(q))$ denotes the set of facets of polytope $\mathit{Inv}(q)$,
- $\mathcal{S} = (A_q, B_q, a_q)_{q\in Q}$ is an assignment of affine systems, with $A_q \in \mathbb{R}^{N_q\times N_q}$, $B_q \in \mathbb{R}^{N_q\times m}$ and $a_q \in \mathbb{R}^{N_q}$.
- U is the input set; U is a polytope in $\mathbb{R}^m$.

This definition is interpreted in the following way. Let $u : [t_0, \infty) \to U$ be an input trajectory. Then $(q_i, t_i, x_i)_{i=0}^n$ (with either $n \in \mathbb{N}_0$ or $n = \infty$) is a corresponding hybrid trajectory of $\mathcal{HS}$ if

1) $q_0 \in Q_0$,
2) For all $i \in \{0, 1, \dots, n\}$ the function $x_i : [t_i, t_{i+1}] \to \mathit{Inv}(q_i)$ satisfies the affine differential equation
$$\dot{x}_i(t) = A_{q_i}x_i(t) + B_{q_i}u(t) + a_{q_i}, \tag{6}$$
3) For $i \in \{0, 1, \dots, n-1\}$, x_i leaves $\mathit{Inv}(q_i)$ at time t_{i+1} by crossing a facet $F \in \mathit{Fac}^{\leftarrow}_{q_i}(q_{i+1})$.

In comparison with the definition of a hybrid automaton, the assignment *Fac* replaces both the set of transitions T and the set of guards G. In particular it is assumed that a switch from discrete mode q_i to q_{i+1} occurs, if state x_i leaves $\mathit{Inv}(q_i)$ by crossing a facet $F \in \mathit{Fac}^{\leftarrow}_{q_i}(q_{i+1})$, and that this transition completely depends on the facet that is crossed.

In the sequel we will often omit the adjectives “piecewise-affine” and “over polytopes” for systems of the form (5), and just speak of hybrid systems.

Next we define what a controller is in this hybrid context, and how hybrid feedback should be interpreted.

Definition 3.2: A *feedback control automaton* for a hybrid system $\mathcal{HS}$ is a tuple

$$\mathcal{F} = (S, Q, s_0, \tau, \pi, \mathcal{K}), \tag{7}$$

where

- S is the set of discrete states of automaton $\mathcal{F}$,
- Q is the discrete input set, equal to the set of discrete locations of $\mathcal{HS}$,
- s_0 is the (fixed) initial state of the deterministic automaton $\mathcal{F}$,
- $\tau : S \times Q \to S$ is the memory update function,
- $\mathcal{K} = \cup_{q\in Q}\mathcal{K}_q$ where for each $q \in Q$, $\mathcal{K}_q$ is a set of affine mappings from $\mathit{Inv}(q)$ to U.
- $\pi : S \times Q \to \mathcal{K}$ is the output function, with the property that for all $s \in S$ and $q \in Q$: $\pi(s, q) \in \mathcal{K}_q$.

A feedback control automaton may be considered as an automaton with inputs and outputs. Given a (finite or infinite) input trajectory $q_0, q_1, q_2, \dots$, the automaton produces a state trajectory $s_0, s_1, s_2, \dots$ defined by the transitions

$$s_{i+1} = \tau(s_i, q_i),$$

and a corresponding output sequence $K_{q_0}, K_{q_1}, K_{q_2}, \dots$ of affine mappings, with $K_{q_i} = \pi(s_i, q_i)$.

Given a hybrid system $\mathcal{HS}$, a feedback control automaton for $\mathcal{HS}$ describes a state feedback law for this hybrid system in the following way. The hybrid system $\mathcal{HS}$ starts at time t_0 in discrete location $q_0 \in Q_0$, and with arbitrary continuous initial state $x_0(t_0) = x_0^0 \in \mathit{Inv}(q_0)$; the feedback control automaton is initialized in discrete state s_0. For every $i \in \{0, 1, \dots, n\}$ (with either $n \in \mathbb{N}_0$ or $n = \infty$) the continuous state evolves according to the affine differential equation

$$\dot{x}_i(t) = A_{q_i}x_i(t) + B_{q_i}K_{q_i}(x_i(t)) + a_{q_i}, \quad x_i(t_i) = x_i^0, \tag{8}$$

with $K_{q_i} = \pi(s_i, q_i)$ and x_i^0 an arbitrary element in $\mathit{Inv}(q_i)$. If at time t_{i+1} the continuous state trajectory $x_i(t)$ crosses facet $F \in \mathcal{FT}(\mathit{Inv}(q_i))$, then an instantaneous transition to location $\mathit{Fac}_{q_i}(F)$ of hybrid system $\mathcal{HS}$ occurs. At the same time instant, the control automaton switches to state $s_{i+1} = \tau(s_i, q_i)$, and the affine system $(A_{q_{i+1}}, B_{q_{i+1}}, a_{q_{i+1}})$ at location q_{i+1} is governed by the affine feedback law $K_{q_{i+1}} = \pi(s_{i+1}, q_{i+1})$. Again, the continuous initial state $x_{i+1}(t_{i+1})$ is assumed to be an arbitrary element of $\mathit{Inv}(q_{i+1})$. In this way, the interconnection of hybrid system $\mathcal{HS}$ and feedback control automaton $\mathcal{F}$ turns out to be a hybrid automaton.

Before we can define the closed-loop automaton in a formal way, we first have to specify what the exact meaning of *crossing a facet* is. This is not completely obvious in those cases where the exit point is an element of a lower dimensional face of the state polytope, and belongs to more than one facet. We resolve this issue by adopting the terminology and approach described in [9, Section IV.B].

Definition 3.3: Let P be a full dimensional closed convex polytope in $\mathbb{R}^n$, and let $h : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be an affine mapping. Let F be a facet of P.

(i) The *exit set* of F with respect to the affine dynamics $\dot{x}(t) = h(x(t))$ on polytope P is defined by

$$ES(F, P, h) := \mathrm{cl}\{x \in F \mid n_F^T h(x) > 0\}, \qquad (9)$$

where n_F denotes the outward unit normal vector of F w.r.t. polytope P.

(ii) A trajectory $x : [t_0, t_1] \to P$ that satisfies the differential equation $\dot{x}(t) = h(x(t))$ is said to leave polytope P at time t_1 by *crossing* facet F if

$$x(t_1) \in Dept(P, h) \cap ES(F, P, h). \qquad (10)$$

According to [9, Lemma 4.7] every point $\hat{x} \in Dept(P, h)$ belongs to the exit set of at least one facet. It may happen that a point $\hat{x} \in Dept(P, h)$ on a lower dimensional face of P belongs to the exit sets of more than one facet. It is also possible that such a point belongs to a facet F, but is not an element of $ES(F, P, h)$.

Assumption 3.4: If a trajectory of the continuous state of a piecewise-affine hybrid automaton leaves polytope $Inv(q)$ by *crossing* the facets $F_1, \ldots, F_k$, then the discrete mode will switch instantaneously to a mode that belongs to the set $\{Fac_q(F_i) \mid i = 1, \ldots, k\}$. I.e. for all $q, q' \in Q$, guard $G(q, q')$ is given by

$$G(q, q') = \bigcup_{F \in Fac_q{}^{\leftarrow}(q')} ES(F, Inv(q), h_q). \qquad (11)$$

Assumption 3.4 induces some non-determinism in the switching behavior of the underlying automaton. If a trajectory leaves $Inv(q)$ by crossing several facets at the same time, then the induced transition is not uniquely determined.

The previous considerations lead to the following construction of the closed-loop of a hybrid system $\mathcal{HS}$ and a feedback control automaton $\mathcal{F}$.

Definition 3.5: The closed-loop of a piecewise-affine hybrid system $\mathcal{HS} = (Q, Q_0, N, Inv, Fac, \mathcal{S}, U)$ and feedback control automaton $\mathcal{F} = (S, Q, s_0, \tau, \pi, \mathcal{K})$ for $\mathcal{HS}$ is the hybrid automaton $\mathrm{CL}(\mathcal{HS}, \mathcal{F}) = (S_p, S_{p,0}, T_p, N_p, Inv_p, G_p, \mathcal{S}_p)$ with

- $S_p = Q \times S$ the finite set of discrete states,
- $S_{p,0} = Q_0 \times \{s_0\}$ the set of initial discrete states,
- $N_p : S_p \to \mathbb{N}$ is the map $(q, s) \mapsto N_q$, describing the dimension of the continuous state in mode $(q, s) \in S_p$,
- $Inv_p : S_p \to 2^{\mathbb{R}^{N_q}}$ is the invariant map, with $Inv_p(q, s) = Inv(q)$ for all $(q, s) \in S_p$,
- $\mathcal{S}_p = (h_{(q,s)})_{(q,s) \in Q \times S}$ is the assignment of affine vector fields $h_{(q,s)} : Inv(q) \to \mathbb{R}^{N_q}$, $((q, s) \in Q \times S)$, where $h_{(q,s)}(x) = A_q x + B_q \pi(s, q)(x) + a_q$,
- $T_p \subset S_p \times S_p$ the set of discrete transitions
$$T_p = \begin{array}{l}\{((q, s), (q', s')) \in S_p \times S_p \mid s' = \tau(s, q) \text{ and} \\ \exists F \in Fac_q^{\leftarrow}(q') \text{ s.t. } ES(F, Inv(q), h_{(q,s)}) \neq \varnothing\},\end{array}$$
- Guards G_p associating the union of exit sets
$$G_p((q, s), (q', s')) = \bigcup_{F \in Fac_q{}^{\leftarrow}(q')} ES(F, Inv(q), h_{(q,s)})$$
to each transition $((q, s), (q', s')) \in T_p$.

Note that the discrete part of the closed-loop system is obtained by taking the product of the automata describing the discrete switching of $\mathcal{HS}$ and $\mathcal{F}$, respectively. The continuous open-loop dynamics is simply copied from $\mathcal{HS}$. The discrete state s of $\mathcal{F}$ only plays a role in the choice of the affine control law $\pi(s, q)$. This opens the possibility to apply different feedback laws on the continuous dynamics at one discrete location of the hybrid system $\mathcal{HS}$.

According to Definition 2.4 the words generated by the hybrid trajectories $((q_i, s_i), t_i, x_i)_{i=0}^n$ of $\mathrm{CL}(\mathcal{HS}, \mathcal{F})$ are elements of $(Q \times S)^\omega_{\text{tot}}$. Since we want to influence the words generated by the hybrid system $\mathcal{HS}$ by application of a feedback control automaton $\mathcal{F}$, only the first component (from Q_{tot}) in every element of these words is of interest.

Definition 3.6: Let $\rho = ((q_i, s_i), t_i, x_i)_{i=0}^n$ be a hybrid trajectory of the closed-loop of hybrid system $\mathcal{HS}$ and feedback control automaton $\mathcal{F}$. Then the word $w_{\text{cl}}(\rho)$ generated by ρ w.r.t. to the closed-loop system is an infinite sequence $w_{\text{cl}}(\rho) = (w_i)_{i \in \mathbb{N}_0} \in Q^\omega_{\text{tot}}$ defined in the following way:

(1) If $\rho = ((q_i, s_i), t_i, x_i)_{i=0}^n$ is a *finite* sequence, then $w_i := q_i$ for $i \leq 0 \leq n$ and $w_i := \xi(q_n)$ for $i > n$.
(2) If $\rho = ((q_i, s_i), t_i, x_i)_{i=0}^\infty$ is an *infinite* sequence, then $w_i := q_i$ for all $i \in \mathbb{N}_0$.

Definition 3.7: Let $\mathcal{HS}$ be a piecewise-affine hybrid system, and let $\mathcal{F}$ be a feedback control automaton for $\mathcal{HS}$. Then the language generated by $\mathcal{HS}$ in closed-loop with $\mathcal{F}$ is defined by

$$\mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F})) := \{w_{\text{cl}}(\rho) \mid \rho \in \mathcal{B}(\mathrm{CL}(\mathcal{HS}, \mathcal{F}))\} \qquad (12)$$

Hence, $\mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F}))$ is the projection of the full closed-loop language $\mathcal{L}(\mathrm{CL}(\mathcal{HS}, \mathcal{F}))$ from $(Q \times S)^\omega_{\text{tot}}$ to Q^ω_{tot}.

In this paper we study how feedback control automata can be used to influence the language of a hybrid system:

Problem 3.8: Let $\mathcal{HS}$ be a piecewise-affine hybrid system with discrete state set Q, and let ϕ be an LTL-formula over Q_{tot} that describes the control objective that we want to achieve. The problem is to find a feedback control automaton $\mathcal{F}$ for $\mathcal{HS}$ such that all $w \in \mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F}))$ satisfy ϕ.

IV. Transition systems and control automata

Instead of solving Problem 3.8 directly, we translate it from a hybrid into a purely discrete-event setting, and first consider the same control problem for discrete event systems with inputs.

Definition 4.1 (Transition system): A finite (non-deterministic) transition system is a tuple $\mathcal{T} = (Q, Q_0, \Sigma, \delta)$, where Q is a finite set of states, $Q_0 \subseteq Q$ is the set of initial states, Σ is a finite input alphabet, and $\delta : Q \times \Sigma \to 2^Q$ is a (non-deterministic) transition function.

Whenever Q_0 is not specified, we assume that $Q_0 = Q$. For a given state $q \in Q$, the set of available (feasible) inputs is denoted by Σ_q, i.e. $\Sigma_q = \{\sigma \in \Sigma \mid |\delta(q, \sigma)| \geq 1\}$. In the sequel we assume that $\Sigma_q \neq \varnothing$ for all $q \in Q$. If at state $q \in Q$ an input $\sigma \in \Sigma_q$ is applied, then the state will switch to an element of the set $\delta(q, \sigma)$. Obviously, a transition system $\mathcal{T}$ is *deterministic* if it has only one initial state and its transition function satisfies $|\delta(q, \sigma)| = 1$, $\forall q \in Q,\ \sigma \in \Sigma_q$.

Definition 4.2: Let $\mathcal{T} = (Q, Q_0, \Sigma, \delta)$ be a transition system, and let $w = (q_i)_{i \in \mathbb{N}_0} \in Q^\omega$ and $\sigma = (\sigma_i)_{i \in \mathbb{N}_0} \in \Sigma^\omega$ be infinite sequences of states and inputs, respectively. Then σ is called an *admissible input word* for the *(state) word* w if the following conditions are satisfied:

(i) $\forall i \in \mathbb{N}_0 : \ \sigma_i \in \Sigma_{q_i}$,

(ii) $\forall i \in \mathbb{N}_0 : \ q_{i+1} \in \delta(q_i, \sigma_i)$.

The language $\mathcal{L}(\mathcal{T})$ of all state words generated by transition system $\mathcal{T}$ is given by $\mathcal{L}(\mathcal{T}) = \{w \in Q^\omega \mid w_0 \in Q_0,$ and there exists an admissible input word σ for $w\}$.

A control objective for a transition system $\mathcal{T}$ may be formulated as a requirement on the set of all (state) words. E.g., given an LTL-formula ϕ over Q, one may want to guarantee that all words $w \in \mathcal{L}(\mathcal{T})$ satisfy formula ϕ. Usually, the realization of such a control objective requires the design of an automaton that generates suitable input values $\sigma_i \in \Sigma_{q_i}$ based on knowledge of the finite state sequence $q_0, q_1, q_2, \ldots, q_i$.

Definition 4.3 (Control automaton): A control automaton $\mathcal{A}$ for a transition system $\mathcal{T} = (Q, Q_0, \Sigma, \delta)$ is a tuple $\mathcal{A} = (S, Q, s_0, \tau, \pi, \Sigma)$, with:

- S is a finite set of states,
- s_0 is the (deterministic) initial state,
- Q is the input set, and is equal to the set of states of $\mathcal{T}$,
- $\tau : S \times Q \to S$ is the memory update function,
- Σ is the output set, and is equal to the input set of $\mathcal{T}$,
- $\pi : S \times Q \to \Sigma$ is the output function, with the property that $\pi(s, q) \in \Sigma_q$ for all $(s, q) \in S \times Q$.

Given initial state $s_0 \in S$ and applying the word $w = (q_i)_{i \in \mathbb{N}_0} \in \mathcal{L}(\mathcal{T})$, the automaton produces a state word $(s_i)_{i \in \mathbb{N}_0}$ satisfying $s_{i+1} = \tau(s_i, q_i)$. If the corresponding output word $(\sigma_i)_{i \in \mathbb{N}_0} \in \Sigma^\omega$ given by $\sigma_i = \pi(s_i, q_i)$, serves as an admissible input word for $w \in \mathcal{L}(\mathcal{T})$, we obtain the following interconnection of $\mathcal{T}$ and $\mathcal{A}$:

Definition 4.4: The *interconnection* $\mathrm{IC}(\mathcal{T}, \mathcal{A})$ of the (non-deterministic) transition system $\mathcal{T} = (Q, Q_0, \Sigma, \delta)$ and the control automaton $\mathcal{A} = (S, Q, s_0, \tau, \pi, \Sigma)$ is defined as the non-deterministic automaton $(S_p, S_{p,0}, \delta_p)$, where

- $S_p = Q \times S$ is the finite state set,
- $S_{p,0} = Q_0 \times \{s_0\}$ is the set of initial states,
- $\delta_p : S_p \to 2^{S_p}$ is the (non-deterministic) transition function $\delta_p(q, s) = \{(\hat{q}, \hat{s}) \in Q \times S \mid \hat{q} \in \delta(q, \pi(s, q))$ and $\hat{s} = \tau(s, q)\}$.

Again, since control automaton $\mathcal{A}$ is the object to be designed, we are primarily interested in the words $w \in Q^\omega$, that are generated by the transition system $\mathcal{T}$, interconnected with control automaton $\mathcal{A}$.

Definition 4.5: The language $\mathcal{L}(\mathcal{T}, \mathrm{IC}(\mathcal{T}, \mathcal{A}))$ of all state words that are generated by transition system $\mathcal{T}$, after interconnection with control automaton $\mathcal{A}$, is defined by

$$\begin{aligned} &\mathcal{L}(\mathcal{T}, \mathrm{IC}(\mathcal{T}, \mathcal{A})) := \{(q_i)_{i \in \mathbb{N}_0} \in Q^\omega \mid \exists (s_i)_{i \in \mathbb{N}_0} \in S^\omega \text{ s.t.} \\ &(q_0, s_0) \in S_{p,0}, \text{and } \forall i \in \mathbb{N}_0 : (q_{i+1}, s_{i+1}) \in \delta_p(q_i, s_i)\}. \end{aligned} \tag{13}$$

In the setting of transition systems and control automata, Problem 3.8 corresponds to:

Problem 4.6: Let $\mathcal{T} = (Q, Q_0, \Sigma, \delta)$ be a transition system, and let ϕ be an LTL-formula over Q. Find a control automaton $\mathcal{A} = (S, Q, s_0, \tau, \pi, \Sigma)$ such that all words $w \in \mathcal{L}(\mathcal{T}, \mathrm{IC}(\mathcal{T}, \mathcal{A}))$ satisfy ϕ.

For solving Problem 4.6, constructive methods exist, based on interconnection with a suitable Büchi automaton ([2]). The main idea of this paper is to transform hybrid control Problem 3.8 into Problem 4.6 for a related transition system, obtained by abstraction, to design a control automaton in the discrete-event setting, and to transform this control automaton back into a suitable feedback control automaton for the original hybrid system.

V. Abstraction of a hybrid system to a control transition system

Let q be one particular discrete mode of a piecewise-affine hybrid system $\mathcal{HS}$. If at mode q an admissible affine control law $K : \mathit{Inv}(q) \to U$ is applied to the continuous state x_q, then the set of possible transitions from mode q is restricted by the behavior of the closed-loop dynamics

$$\dot{x}_q(t) = h_q(x_q(t)), \tag{14}$$

on the polytope $\mathit{Inv}(q)$, where $h_q(x) = A_q x + B_q K(x) + a_q$ denotes the affine vector field of the closed-loop system. In particular, the set of possible transitions depends on the set of exit sets that are intersected by the departure set $\mathit{Dept}(\mathit{Inv}(q), h_q)$. We define the set $\mathit{NextState}(q, K)$ as the set of all discrete states to which the system may switch if controller K is applied to the continuous dynamics at location q. Again we distinguish two cases: switching to an other location in finite time, and remaining in the same location forever. Formally stated, *NextState* is a mapping from $\bigcup_{q \in Q}(\{q\} \times \mathcal{K}_q)$ to $2^{Q_{\text{tot}}}$ defined by

$$\begin{cases} \text{For } q' \in Q : q' \in \mathit{NextState}(q, K) \text{ if and only if} \\ \qquad \exists F \in \mathit{Fac}_q^{\leftarrow}(q') \text{ s.t. } \mathit{ES}(F, \mathit{Inv}(q), h_q) \neq \varnothing, \\ \\ \text{For } q' \in Q_{\text{inv}} : \ q' \in \mathit{NextState}(q, K) \text{ if and only if} \\ \qquad q' = \xi(q) \text{ and } \exists x_q^0 \in \mathit{Inv}(q) \text{ s.t. solution } x_q(t) \text{ of} \\ \qquad \text{(14) with initial state } x_q(t_0) = x_q^0 \text{ satisfies} \\ \qquad x_q(t) \in \mathit{Inv}(q) \text{ for all } t \geq t_0. \end{cases} \tag{15}$$

Note that $\mathit{ES}(F, \mathit{Inv}(q), h_q) \neq \varnothing$ if and only if there exists of a solution of differential equation (14) that leaves $\mathit{Inv}(q)$

by crossing facet F. Hence, *NextState*(q, K) is a nonempty subset of $Q \cup \{\xi(q)\}$.

The function *NextState* is an important building block for the abstraction of a hybrid system to a transition system. In this setting, the exact form of the applied affine control law is not important. Instead one is interested in the set of discrete states to which the system can switch in exactly one step. This observation may be used to restrict the choice of possible feedbacks to a finite number of feedbacks: if two (or more) different feedbacks lead to the same discrete behavior, it is sufficient to consider only one of these feedback laws. Formally, one may describe this by the following partial ordering.

Definition 5.1: Let $q \in Q$ and let K_1 and K_2 be two admissible feedback laws for system (14) on *Inv*(q). Then

(i) $K_1 \preceq_q K_2$ iff *NextState*$(q, K_1) \subseteq$ *NextState*(q, K_2),
(ii) $K_1 \sim_q K_2$ iff $K_1 \preceq_q K_2$ and $K_2 \preceq_q K_1$.

For the solution of Problem 3.8 it is sufficient to consider only controllers that are minimal with respect to the partial ordering $\preceq_q$.

Proposition 5.2: Let $\mathcal{HS}$ be a piecewise-affine hybrid system, and let $\mathcal{F}_1 = (S, Q, s_0, \tau, \pi_1, \mathcal{K})$ and $\mathcal{F}_2 = (S, Q, s_0, \tau, \pi_2, \mathcal{K})$ be two feedback control automata for $\mathcal{HS}$ with the same discrete behavior (but with different outputs). Then

$$\forall q \in Q, \forall s \in S : \pi_1(s, q) \preceq_q \pi_2(s, q), \qquad (16)$$
$$\implies$$
$$\mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F}_1)) \subseteq \mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F}_2)).$$

In particular, if condition (16) is satisfied and all words $w \in \mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F}_2))$ satisfy the LTL-formula ϕ, then all words $w \in \mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F}_1))$ satisfy ϕ.

Corollary 5.3: If in the setting of Proposition 5.2

$$\forall q \in Q, \forall s \in S : \pi_1(s, q) \sim_q \pi_2(s, q),$$

then hybrid system $\mathcal{HS}$ generates in closed-loop with $\mathcal{F}_1$ the same set of words as in closed-loop with $\mathcal{F}_2$, i.e.

$$\mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F}_1)) = \mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F}_2)).$$

Obviously, $\sim_q$ describes an equivalence relation on the set $\mathcal{K}_q$ of admissible controllers for the affine system (A_q, B_q, a_q) on polytope *Inv*(q). The number of equivalence classes is bounded above by $2^{|Q|+1}$, the cardinality of the set of subsets of $Q \cup \{\xi(q)\}$. Note however that subsets $\tilde{Q}$ of $Q \cup \{\xi(q)\}$ may exist, for which the corresponding equivalence class does not exist, because the set $\{K \in \mathcal{K}_q \mid$ *NextState*$(q, K) = \tilde{Q}\}$ is empty.

The partial ordering $\preceq_q$ may be extended in a straightforward way to the set of equivalence classes $\mathcal{K}_q / \sim_q$. So, if $\mathrm{EqCl}_q(K_i)$, $i = 1, 2$ denotes the equivalence class $\{K \in \mathcal{K}_q \mid$ *NextState*$(q, K) =$ *NextState*$(q, K_i)\}$ of control law K_i, then $\mathrm{EqCl}_q(K_1) \preceq_q \mathrm{EqCl}_q(K_2)$ if and only if

$$\forall K \in \mathrm{EqCl}_q(K_1) \forall L \in \mathrm{EqCl}_q(K_2) : \textit{NextState}(q, K) \subseteq \textit{NextState}(q, L).$$

An element (equivalence class) $E_{q,1}$ of $\mathcal{K}_q / \sim_q$ is called *minimal* if for all $E_{q,2} \in \mathcal{K}_q / \sim_q$ we have

$$E_{q,2} \preceq_q E_{q,1} \implies E_{q,2} = E_{q,1}.$$

Since $\mathcal{K}_q / \sim_q$ is a finite and partially ordered set, it follows that $\mathcal{K}_q / \sim_q$ has a nonempty subset of minimal elements (see e.g. [7, p. 39]), denoted by $\mathcal{K}_{q,\min}$. A feedback control automaton $\mathcal{F} = (S, Q, s_0, \tau, \pi, \mathcal{K})$ is called *minimal* if

$$\forall s \in S \, \forall q \in Q \, \exists E_{(q,s)} \in \mathcal{K}_{q,\min} : \pi(s, q) \in E_{(q,s)}.$$

Corollary 5.4: Let $\mathcal{HS}$ be a piecewise-affine hybrid system, and let ϕ be an LTL formula. If there exists a feedback control automaton $\mathcal{F}$ that solves Problem 3.8, then there also exists a *minimal* feedback control automaton that solves Problem 3.8.

Apparently, it is sufficient to restrict the attention in location q to continuous control laws from the equivalence classes in $\mathcal{K}_{q,\min}$.

Given a hybrid system $\mathcal{HS}$ we now define a transition system corresponding to $\mathcal{HS}$ by applying a state feedback to the continuous dynamics in each discrete location, and subsequently abstracting from the continuous dynamics.

Definition 5.5: Let $\mathcal{HS}$ be a piecewise-affine hybrid system with discrete state set Q. Then the *control transition system CTS*$(\mathcal{HS})$ *corresponding to* $\mathcal{HS}$ is defined as the transition system

$$\mathrm{CTS}(\mathcal{HS}) = (Q_{\mathrm{tot}}, Q_0, \hat{\mathcal{K}}_{\min}, \Delta),$$

where $Q_{\mathrm{tot}} = Q \cup Q_{\mathrm{inv}}$ is the finite set of discrete states, $Q_0 \subset Q$ is the set of initial states, and $\hat{\mathcal{K}}_{\min} := (\cup_{q \in Q} \mathcal{K}_{q,\min}) \cup \{\epsilon\}$ is the set of all inputs. Here $\mathcal{K}_{q,\min}$ is the set of all feasible inputs for state $q \in Q$, and the symbol ϵ is the only feasible input for states $q \in Q_{\mathrm{inv}}$. Finally, $\Delta : Q_{\mathrm{tot}} \times \hat{\mathcal{K}}_{\min} \to 2^{Q_{\mathrm{tot}}}$ is the (non-deterministic) transition function given by

$$\Delta(q, E) = \begin{cases} \{q\} & \text{if } q \in Q_{\mathrm{inv}} \text{ and } E = \epsilon, \\ \textit{NextState}(q, K) & \text{if } q \in Q \text{ and } E \in \mathcal{K}_{q,\min}, \\ & \quad \text{with } K \text{ an arbitrary element of } E, \\ \varnothing & \text{otherwise.} \end{cases} \qquad (17)$$

Transition function Δ in (17) is well-defined. If $q \in Q_{\mathrm{inv}}$ this is obvious, and if $q \in Q$ and $E \in \mathcal{K}_{q,\min}$, then for every pair of admissible feedback laws $K_1, K_2 \in E$ we know that $K_1 \sim_q K_2$, hence *NextState*$(q, K_1) =$ *NextState*(q, K_2). Furthermore, transition system $\mathrm{CTS}(\mathcal{HS})$ satisfies the condition that for all $q \in Q_{\mathrm{tot}}$ the set of feasible inputs Σ_q is nonempty.

In the dynamics of control transition system $\mathrm{CTS}(\mathcal{HS})$, the discrete states in Q_{inv} only play a limited role: they have only one admissible input ϵ, which results in a self transition. Therefore Q_{inv} may be considered as a cemetery: once a location in Q_{inv} is reached, it is impossible to leave it.

In the next section we will state the main results of this paper. They describe the relationship between the control of a hybrid system $\mathcal{HS}$ (by a feedback control automaton) and the control of its corresponding control transition system

$\mathrm{CTS}(\mathcal{HS})$ (by a control automaton): given a control objective in terms of an LTL-formula ϕ, Control Problem 3.8 for hybrid system $\mathcal{HS}$ is solvable if and only if Control Problem 4.6 for control transition system $\mathrm{CTS}(\mathcal{HS})$ is solvable. Furthermore, we will describe how a feedback control automaton for $\mathcal{HS}$ solving Control Problem 3.8 may be used to construct a control automaton solving Control Problem 4.6 and vice versa.

VI. Temporal logic control for hybrid systems and control transition systems

Let $\mathcal{HS}$ be a piecewise-affine hybrid system, with corresponding control transition system $\mathrm{CTS}(\mathcal{HS})$. First we describe the construction of a control automaton for $\mathrm{CTS}(\mathcal{HS})$ from a *minimal* feedback control automaton for $\mathcal{HS}$.

Definition 6.1: Let $\mathcal{F} = (S, Q, s_0, \tau, \pi, \mathcal{K})$ be a minimal feedback control automaton for piecewise-affine hybrid system $\mathcal{HS}$, and let $\mathrm{CTS}(\mathcal{HS})$ be the control transition system corresponding to $\mathcal{HS}$. The *control automaton* $\mathrm{CA}(\mathcal{F})$ for $\mathrm{CTS}(\mathcal{HS})$ corresponding to $\mathcal{F}$ is defined as

$$\mathrm{CA}(\mathcal{F}) = (S, Q_{\text{tot}}, s_0, f_1(\tau), f_2(\pi), \hat{\mathcal{K}}_{\text{min}}), \qquad (18)$$

where S denotes the state set, s_0 is the initial state, $Q_{\text{tot}} = Q \cup Q_{\text{inv}}$ is the input set, and $\hat{\mathcal{K}}_{\text{min}}$ is the output set, and with memory update function $f_1(\tau) : S \times Q_{\text{tot}} \to S$ given by

$$f_1(\tau)(s,q) = \begin{cases} \tau(s,q) & \text{if} \quad q \in Q, \\ s & \text{if} \quad q \in Q_{\text{inv}}, \end{cases} \qquad (19)$$

and output function $f_2(\pi) : S \times Q_{\text{tot}} \to \hat{\mathcal{K}}_{\text{min}}$ given by

$$f_2(\pi)(s,q) = \begin{cases} \mathrm{EqCl}_q(\pi(s,q)) & \text{if} \quad q \in Q, \\ \epsilon & \text{if} \quad q \in Q_{\text{inv}}. \end{cases} \qquad (20)$$

Note that $\mathrm{CA}(\mathcal{F})$ is a well-defined control automaton for $\mathrm{CTS}(\mathcal{HS})$. Since the feedback control automaton $\mathcal{F}$ is minimal, we know that for all $(s,q) \in S \times Q$ the feedback law $\pi(s,q)$ is an element of the equivalence class $\mathrm{EqCl}_q(\pi(s,q)) \in \mathcal{K}_{q,\text{min}}$, so $f_2(\pi)(s,q) = \mathrm{EqCl}_q(\pi(s,q)) \in \mathcal{K}_{q,\text{min}}$ is a feasible input to control transition system $\mathrm{CTS}(\mathcal{HS})$ at state $q \in Q$. Similarly, if $(s,q) \in S \times Q_{\text{inv}}$ then $f_2(\pi(s,q)) = \epsilon$ is a feasible input to $\mathrm{CTS}(\mathcal{H})$ at state $q \in Q_{\text{inv}}$.

The first main result states that the application of feedback control automaton $\mathcal{F}$ on hybrid system $\mathcal{HS}$ yields the same closed-loop language as the interconnection of control transition system $\mathrm{CTS}(\mathcal{HS})$ with control automaton $\mathrm{CA}(\mathcal{F})$.

Theorem 6.2: Let $\mathcal{HS}$ be a piecewise-affine hybrid system, and let $\mathrm{CTS}(\mathcal{HS})$ be the control transition system corresponding to $\mathcal{HS}$. Let $\mathcal{F}$ be a minimal feedback control automaton for $\mathcal{HS}$, and let $\mathrm{CA}(\mathcal{F})$ be the corresponding control automaton for $\mathrm{CTS}(\mathcal{HS})$. Then

$$\begin{aligned} &\mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F})) = \\ &\quad \mathcal{L}(\mathrm{CTS}(\mathcal{HS}), \mathrm{IC}(\mathrm{CTS}(\mathcal{HS}), \mathrm{CA}(\mathcal{F}))). \end{aligned} \qquad (21)$$

In Theorem 6.2 a control automaton for transition system $\mathrm{CTS}(\mathcal{HS})$ is constructed, based on a minimal feedback automaton $\mathcal{F}$ for hybrid system $\mathcal{HS}$. It is also possible to proceed in the opposite direction. In the next definition, a given control automaton for control transition system $\mathrm{CTS}(\mathcal{HS})$ is transformed into a minimal feedback control automaton for the original hybrid system $\mathcal{HS}$.

Definition 6.3: Let $\mathcal{HS}$ be a piecewise-affine hybrid system, with discrete state set Q, and let $\mathrm{CTS}(\mathcal{HS})$ be the corresponding control transition system. Let $\mathcal{A} = (S, Q_{\text{tot}}, s_0, \tau, \pi, \hat{\mathcal{K}}_{\text{min}})$ be a control automaton for $\mathrm{CTS}(\mathcal{HS})$, and $\rho = (\rho_q)_{q \in Q}$ a tuple of selection functions $\rho_q : \mathcal{K}_{q,\text{min}} \to \mathcal{K}_q$ with the property that $\rho_q(E) \in E$ for all $E \in \mathcal{K}_{q,\text{min}}$. Then the minimal *feedback control automaton* $FCA(\mathcal{A}, \rho)$ *for* $\mathcal{HS}$ *corresponding to* $\mathcal{A}$ *and* ρ is defined by

$$\mathrm{FCA}(\mathcal{A}, \rho) := (S, Q, s_0, g_1(\tau), g_2(\pi), \mathcal{K}),$$

with state set S, deterministic initial state s_0, input set Q, memory update function $g_1(\tau) : S \times Q \longrightarrow S$, given by $g_1(\tau)(s,q) = \tau(s,q)$, output set $\mathcal{K}$, and output function $g_2(\pi)$ given by

$$g_2(\pi) : S \times Q \longrightarrow \mathcal{K} : \; g_2(\pi)(s,q) = \rho_q(\pi(s,q)). \qquad (22)$$

The feedback control automaton $\mathrm{FCA}(\mathcal{A}, \rho)$ is well-defined, because for every $(s,q) \in S \times Q$, the feedback law $g_2(\pi)(s,q)$ belongs to $\mathcal{K}_q$. The next result can be considered as a reflection of Theorem 6.2.

Theorem 6.4: Let $\mathcal{HS}$ be a piecewise-affine hybrid system, and let $\mathrm{CTS}(\mathcal{HS})$ be the control transition system corresponding to $\mathcal{HS}$. If $\mathcal{A}$ is a control automaton for $\mathrm{CTS}(\mathcal{HS})$, and $\rho = (\rho_q)_{q \in Q}$ is a is a tuple of selection functions $\rho_q : \mathcal{K}_{q,\text{min}} \to \mathcal{K}_q$ with the property that $\rho_q(E) \in E$ for all $E \in \mathcal{K}_{q,\text{min}}$, then

$$\begin{aligned} &\mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathrm{FCA}(\mathcal{A}, \rho))) = \\ &\quad \mathcal{L}(\mathrm{CTS}(\mathcal{HS}), \mathrm{IC}(\mathrm{CTS}(\mathcal{HS}), \mathcal{A})). \end{aligned} \qquad (23)$$

By combining Theorem 6.2 and Theorem 6.4, the design of a minimal feedback control automaton for a hybrid system $\mathcal{HS}$ may be reduced to the design of a control automaton for the control transition system $\mathrm{CTS}(\mathcal{HS})$ corresponding to $\mathcal{HS}$.

Theorem 6.5: Let $\mathcal{HS}$ be a piecewise-affine hybrid system with discrete state set Q, and let $\mathrm{CTS}(\mathcal{HS})$ be the corresponding control transition system. Let ϕ be an LTL-formula over Q_{tot} that specifies the control objective. Let $\mathrm{Spec}(\phi)$ denote the subset of Q_{tot}^{ω}, that contains all sequences $(q_i)_{i \in \mathbb{N}_0} \in Q_{\text{tot}}^{\omega}$ that satisfy LTL-formula ϕ.

- *(i)* There exists a feedback control automaton $\mathcal{F}$ for $\mathcal{HS}$ such that $\mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F})) \subset \mathrm{Spec}(\phi)$ if and only if there exists a control automaton $\mathcal{A}$ for the control transition system $\mathrm{CTS}(\mathcal{HS})$ such that $\mathcal{L}(\mathrm{CTS}(\mathcal{HS}), \mathrm{IC}(\mathrm{CTS}(\mathcal{HS}, \mathcal{A}))) \subset \mathrm{Spec}(\phi)$.
- *(ii)* If the minimal feedback control automaton $\mathcal{F}$ is such that $\mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathcal{F})) \subset \mathrm{Spec}(\phi)$, then the interconnection of $\mathrm{CTS}(\mathcal{HS})$ and $\mathrm{CA}(\mathcal{F})$ satisfies $\mathcal{L}(\mathrm{CTS}(\mathcal{HS}), \mathrm{IC}(\mathrm{CTS}(\mathcal{HS}, \mathrm{CA}(\mathcal{F})))) \subset \mathrm{Spec}(\phi)$.

Let $\rho = (\rho_q)_{q \in Q}$ be a tuple of selection functions $\rho_q : \mathcal{K}_{q,\text{min}} \to \mathcal{K}_q$.

- *(iii)* If control automaton $\mathcal{A}$ guarantees that $\mathcal{L}(\mathrm{CTS}(\mathcal{HS}), \mathrm{IC}(\mathrm{CTS}(\mathcal{HS}, \mathcal{A}))) \subset \mathrm{Spec}(\phi)$, then application of the minimal feedback control automaton $\mathrm{FCA}(\mathcal{A}, \rho)$ to hybrid system $\mathcal{HS}$ yields $\mathcal{L}(\mathcal{HS}, \mathrm{CL}(\mathcal{HS}, \mathrm{FCA}(\mathcal{A}, \rho))) \subset \mathrm{Spec}(\phi)$.

Theorem 6.5 can be applied to translate Control Problem 3.8 for hybrid system $\mathcal{HS}$ into the purely discrete-event Control Problem 4.6 for control transition system $\mathrm{CTS}(\mathcal{HS})$. In general, the latter will be easier to solve, and any technique that is available in the discrete-event setting may be used. Once a suitable control automaton $\mathcal{A}$ for $\mathrm{CTS}(\mathcal{HS})$ has been found, Theorem 6.5 describes how it can be transformed into a feedback control automaton $\mathrm{FCA}(\mathcal{A}, \rho)$ for $\mathcal{HS}$, that realizes the required control objective in the hybrid setting.

Remark 6.6: If Control Problem 4.6 for $\mathrm{CTS}(\mathcal{HS})$ is *not* solvable, it is obvious that the corresponding Control Problem 3.8 for hybrid system $\mathcal{HS}$ with *arbitrary* continuous initial states is not solvable either. However, if the continuous initial states of $\mathcal{HS}$ are *fixed* by reset maps (cf. Remark 2.7), it may happen that there still exists a feedback control automaton for this modified situation, because the language generated by a hybrid automaton with fixed continuous initial states is a subset of the language of the same hybrid automaton with arbitrary continuous initial states. This conservativeness is inherent to our approach based on hybrid systems with *arbitrary* continuous initial states. On the other hand, any solution for the case of arbitrary continuous initial states remains a solution if the continuous initial states are fixed by reset maps.

Note that the solution strategy proposed in this paper will only work if the following two issues are resolved:

1) Construction of control transition system $\mathrm{CTS}(\mathcal{HS})$ from hybrid system $\mathcal{HS}$. Although Definition 5.5 contains a formal description, it is not completely constructive, because in every location $q \in Q$ full knowledge of the set $\mathcal{K}_{q,\min}$ of minimal equivalence classes in $\mathcal{K}_q/\sim_q$ is required. The question is, how this information can be obtained from an implementation of the function *NextState*.
2) A constructive solution to the discrete-event Control Problem 4.6 is needed.

Ad 1: For given $q \in Q$ and $K \in \mathcal{K}_q$, the set *NextState*(q, K) is easily computed: it suffices to check the direction of the affine vector field of the closed-loop system at the vertices of polytope *Inv*(q) and to check for a fixed point in *Inv*(q) (see e.g. [8],[9]). However, if $\mathcal{K}_q$ contains infinitely many elements is is impossible to carry out these computations for all admissible affine feedbacks. Instead one has to verify for each subset $\tilde{Q}$ of the *finite* set $Q \cup \{\xi(q)\}$, whether there exists a continuous control law $K \in \mathcal{K}_q$ such that *NextState*$(q, K) = \tilde{Q}$. This question is a so-called control-to-facet problem (see [9]), that has been solved for piecewise-affine systems on simplices. Also for other types of dynamics (partial) solutions exist (see e.g. [3]).

Ad 2: Let ϕ be an LTL-formula that describes the control objective for control transition system $\mathrm{CTS}(\mathcal{HS})$. Then there exists a so-called *Büchi automaton* $\mathcal{B}$ ([2]) that accepts words $w \in Q^\omega$ if and only if w satisfies LTL-formula ϕ. If $\mathcal{B}$ is a *deterministic* Büchi automaton, it may be interconnected with the (non-deterministic) control transition system. On this product automaton a so-called Büchi game has to be solved. If a winning strategy exists, then the original control problem is solvable. Furthermore, based on the winning strategy, a control automaton for the control transition system $\mathrm{CTS}(\mathcal{HS})$ may be constructed.

The implementation issues mentioned above will be elaborated in more detail in a different paper on hybrid feedback control. There a complete construction will be given for the class of piecewise-affine hybrid systems on simplices.

VII. Concluding remarks

In this paper we have developed a strategy for solving control problems on piecewise-affine hybrid systems with continuous inputs, for which the control objective is specified by an LTL-formula on the words generated by the system. The main idea was to abstract the hybrid system to a control transition system, try to solve the control problem in this purely discrete setting, and to translate the control automaton that was obtained for the abstraction, into a feedback control automaton for the original hybrid system. In this paper, the emphasis was on exact specification of different types of systems, their behavior, the language generated by these systems, and feedback interconnections of systems. In a different paper the authors plan to elaborate this strategy into a constructive algorithm, and to provide some illustrations of its applicability.

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, The algorithmic analysis of hybrid systems. *Theoret. Comput. Sci.*, 138: 3–34, 1995.

[2] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, Cambridge, MA, 2007.

[3] C. Belta and L.C.G.J.M. Habets, Controlling a class of nonlinear systems on rectangles. *IEEE Trans. Aut. Control*, 51: 1749–1759, 2006.

[4] E.M. Clarke, O. Grumberg, and D.A. Peled, *Model Checking*. MIT Press, Cambridge, MA, 1999.

[5] P. Collins, A trajectory-space approach to hybrid systems. In *Proc. Int. Symposium on Math. Theory of Networks and Systems*. Leuven, 2004.

[6] P.J. Collins, *Hybrid Trajectory Spaces*. Report MAS-R0501, Centrum voor Wiskunde en Informatica, Amsterdam, 2005.

[7] H. Gericke, *Theorie der Verbände*. Hochschultaschenbücher Vol. 38/38a, 2. Auflage, Bibliografisches Institut, Mannheim, 1967.

[8] L.C.G.J.M. Habets and J.H. van Schuppen, A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica*, 40: 21–35, 2004.

[9] L.C.G.J.M. Habets, P.J. Collins, and J.H. van Schuppen, Reachability and control synthesis for piecewise-affine hybrid systems on simplices. *IEEE Trans. Aut. Control*, 51: 938–948, 2006.

[10] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya, What's decidable about hybrid automata? *J. Comput. Syst. Sci*, 57: 97–124, 1998.

[11] J. Lygeros, K.H. Johansson, S. Sastry, and M. Egerstedt, On the existence of executions of hybrid automata. In *Proc. 38th IEEE Conf. Decision and Control*, pages 2249–2254, IEEE Press, New York, 1999.

[12] J.C. Willems, Models for dynamics. In U. Kirchgraber and H.O. Walther (Eds.), *Dynamics Reported*, Vol. 2, pp. 171–269, B.G. Teubner, Stuttgart/Wiley, Chichester, 1989.

[13] J.C. Willems, Paradigms and puzzles in the theory of dynamical systems. *IEEE Trans. Aut. Control*, 36: 259–294, 1991.