Proceedings of the 45th IEEE Conference on Decision & Control
Manchester Grand Hyatt Hotel
San Diego, CA, USA, December 13-15, 2006

An Equivalence Result between Linear Hybrid Automata and Piecewise Affine Systems

Stefano Di Cairano and Alberto Bemporad

Abstract— **In this paper we examine a relationship existing among linear hybrid automata (LHA) and piecewise affine (PWA) systems. While a LHA is an autonomous nondeterministic model, a PWA system is a deterministic model with inputs. By extending continuous-time PWA models to include the dynamics of discrete states and resets, we show in a constructive way that a LHA can be equivalently represented as a PWA system, where equivalent means that the two systems generate the same trajectories. The key idea is to model the uncertainty associated with LHA transitions as an additional vector of input disturbances in the corresponding PWA model. By linking the LHA modelling framework (popular in computer science) with the PWA modelling framework (popular in systems science), our equivalence result allows one to expand the use of several existing control theoretical tools (for stability analysis, optimal control, etc.) developed for PWA models to a much wider class of hybrid systems.**

I. INTRODUCTION

Hybrid systems can be considered as a cross-point between computer science and control theory: they can be seen as a discrete system (an automaton) that reacts to a continuous system (a physical process) by influencing its evolution, or, by switching the point of view, as a physical process whose parameters change according to a discrete dynamics, which in turn is influenced by the process itself.

A large variety of mathematical models have been proposed for hybrid systems with different modelling capabilities and different purposes. In particular, control theorists have mainly focused on piecewise affine (PWA) systems [1], mixed logical dynamical (MLD) systems [2], and other classes of hybrid systems like (extended) linear complementarity systems (ELC/LC) and min-max plus scaling (MMPS) systems (see [3] and the references therein). In parallel, several models have been proposed also by computer scientists, among them hybrid automata (HA) [4] are probably the most powerful model. System theoretical properties of HA were investigated in [5]. Linear hybrid automata (LHA) [4], [6] and timed automata (TA) [7] are also popular in the computer science community. Different models have different purposes, in particular computer scientists are mainly concerned with simulation and verification [8], [9], while control theorists are mainly concerned with stability analysis [10], identification [11], model predictive control [2], and reachability analysis/verification [12].

Equivalence relations between MMPS, ELC, LC, MLD and PWA systems were shown, under mild conditions, in [3].

This work was supported by the European Community through the HYCON Network of Excellence, contract number FP6-IST-511368.

The authors are with Dipartimento di Ingegneria dell'Informazione, Università di Siena, Italy, dicairano,bemporad@dii.unisi.it

A precise relation between HA, LHA and TA was shown in [4]. In this paper we connect the above two sets of equivalent models by showing that, under mild assumptions, a LHA can be represented as a PWA system. We give a constructive proof of this by explicitly showing how an equivalent PWA system of a given LHA can be defined. The constructive procedure is also invertible, namely a PWA system obtained through it can be converted back to the original LHA.

In Section II we introduce LHA models and in Section III we describe PWA models and their extension to include discrete dynamics and resets. The representation of the continuous and discrete LHA dynamics in the PWA formalism is described in Section IV. A discussion of the results concludes the paper in Section V.

A. Notation

In this paper we adopt the formalism typically used in computer science for describing LHA, and the formalism of control theory for PWA systems. Sets of reals and general linear subspaces are indicated with calligraphic letters (such as $\mathcal{X}$). The domains of Boolean variables is $\{0, 1\}$, the set of reals is $\mathbb{R}$, the set of positive reals is $\mathbb{R}_+$, and the set of nonnegative reals is $\mathbb{R}_0$. The symbols $\leq$, $\geq$, $>$, $<$ denote componentwise inequality relations when used between vectors. The symbols $\wedge$ and $\vee$ denote logical *and* and logical *or*, respectively. Predicates are logical functions of pure Boolean variables and of Boolean conditions representing the result of a comparison between real variables. The notation $[P(X_1, \ldots X_r)|X_1 \leftarrow x_1, \ldots X_r \leftarrow x_r]$ is used to evaluate the predicate P when the free variables $X_1, \ldots X_r$ take values $x_1, \ldots x_r$, respectively. For instance given $P(X_1, X_2) = ((X_1 < 1) \wedge (X_2 > 4))$ we obtain $[P(X_1, X_2)|X_1 \leftarrow 0,\ X_2 \leftarrow 5] = \text{TRUE}$ while $[P(X_1, X_2)|X_1 \leftarrow 2,\ X_2 \leftarrow 5] = \text{FALSE}$. Given a matrix H, H^i denotes the i-th row of H; given a vector x, x_i denotes the i-th component of x.

II. LINEAR HYBRID AUTOMATA

A linear hybrid automaton (LHA) [4], [13] is a tuple

$$\mathcal{H} = \{X, V, E, flow, inv, init, jump, event, \Sigma\}, \qquad (1)$$

where $X = \{X_1, X_2, \ldots X_n\}$ is the (ordered) collection of continuous (real-valued) states, the couple (V, E) defines a graph in which $V = \{v_1, v_2, \ldots, v_l\}$ is the set of vertices (the discrete states of the LHA), each one representing a *control mode*, and E is the set of directed edges, representing the way control modes are allowed to switch (i.e., the discrete dynamics). Each of the vertex labelling functions $flow$, inv and $init$ assigns a predicate to each control mode, defining the allowed continuous state evolutions, the *invariant set* in which the continuous states must remain when in that mode[1], and the allowed values for initial states, respectively. Functions $jump$ and $event$ are edge labelling functions: $jump$ defines the conditions for changing the control mode, $event$ associates events from the finite set Σ to control mode switches. The free variables in the predicates of $init$ and inv are from X, the ones of $flow$ are from the set of derivatives $\dot{X} = \{\dot{X}_1, \dot{X}_2, \ldots, \dot{X}_n\}$ of continuous states. Resets of continuous states are defined by the predicates assigned by the *jump* function to the vertices. The free variables on such predicates are from $X \cup X'$, where X' is the set of values of state variables after a discrete transition. If the predicate $[jump(v_1, v_2)|X \leftarrow \underline{x}, X' \leftarrow \overline{x}] = \text{TRUE}$ then the transition from the control mode v_1 to the control mode v_2 when the continuous state is $\underline{x}$ and is reset to $\overline{x}$ is allowed. For linear hybrid automata the *init*, *inv*, *flow* and *jump* predicates are the conjunction of linear inequalities[2]. Given a generic predicate P acting on a finite set Z of free variables, for any assignment of the variables in $Z_1 \subset Z$ there may exist more than one assignment of the variables in $Z \setminus Z_1$ such that P holds. From a system theoretical point of view, such an ambiguity may map into nondeterministic state evolutions.

The *flow* function associates to each control mode j the conjunction of predicates

$$\bigwedge_{h=1}^{r_j} \left[\underline{p}_h^{(j)} \leq \sum_{k=1}^{N} q_{h,k}^{(j)} \cdot \dot{X}_k \leq \overline{p}_h^{(j)} \right], \quad (2)$$

where r_j is the number of ranges defining the flow. The predicate takes value TRUE for the values $\dot{x} \in \dot{X}$, $v_j \in V$ such that the inequality (2) is satisfied. Predicate (2) could also contain strict inequalities, that we skip here for compactness of notation.

The set of initial states $(X_0 \times V_0) \subseteq (\mathbb{R}^n \times V)$ contains couples (x_0, v_i) such that $[init(v_i)|X \leftarrow x_0] = \text{TRUE}$. The state of the LHA evolves in the following way. From an initial state (x_0, v_i) at time $\underline{t}_0$ such that $[init(v_i)|X \leftarrow x_0] = \text{TRUE}$, the continuous state evolves for $t \in T_0 = [\underline{t}_0, \overline{t}_0]$ in such a way that $\forall t \in T_0, [flow(v_i)|\dot{X} \leftarrow \dot{x}(t)] \wedge [inv(v_i)|X \leftarrow x(t)] = \text{TRUE}$. The first clause is a condition on feasibility of the dynamics, the second on feasibility of the state trajectory. Let the instant $\overline{t}_0 = \underline{t}_1$ be the control switch instant so that $\exists e = (v_i, v_j) \in E : [jump(e)|X \leftarrow x(\overline{t}_0), X' \leftarrow x(\underline{t}_1)] = \text{TRUE}$ and $[inv(v_j)|X \leftarrow x(\underline{t}_1)] = \text{TRUE}$. Then the evolution proceeds from $x(\underline{t}_1)$ through a continuous flow for $t \in T_1 = [\underline{t}_1, \overline{t}_1]$, when a new switch occurs. Thus, the evolution of the linear hybrid automaton is obtained as a sequence of epochs $T_i = [\underline{t}_i, \overline{t}_i]$, ($\underline{t}_i = \overline{t}_{i-1}$), of continuous evolutions interleaved by discrete events, at which the control mode changes and the continuous state is reset, introducing discontinuities on the continuous state dynamics. Such a sequence of epochs $\mathcal{T} = [T_0, T_1, \ldots]$ is called a *time trajectory*.

[1]Note that the *invariant set* defined for LHA is not "invariant" in the system theoretical sense, namely the dynamical system is not supposed to remain in the set indefinitely.

[2]As pointed out in [6, footnote 7], a disjunction of predicates can be implemented by splitting control modes and transitions.

III. Piecewise Affine Systems

Piecewise affine (PWA) systems [1], [10] are dynamical systems defined by the relations

$$\dot{x}_c(t) = A_{i(t)} x_c(t) + B_{i(t)} u_c(t) + f_{i(t)}, \quad (3a)$$

$$i(t) \;:\; H_{i(t)} x_c(t) + J_{i(t)} u_c(t) \leq K_{i(t)}, \quad (3b)$$

$$\tilde{H}_{i(t)} x_c(t) + \tilde{J}_{i(t)} u_c(t) < \tilde{K}_{i(t)}. \quad (3c)$$

where $x_c(t) \in \mathbb{R}^{n_c}$ is the state vector at time t, and $u_c(t) \in \mathbb{R}^{m_c}$ is the input vector[3]. The index $i(t) \in \mathbb{I} \triangleq \{1, \ldots, s\}$ labels the *active mode* of the system, which is uniquely determined by the condition $\left[\begin{smallmatrix} x(t) \\ u(t) \end{smallmatrix}\right] \in P_{i(t)}$, where the polyhedral region $P_{i(t)} \subseteq \mathbb{R}^{n_c+m_c}$ is defined by inequalities (3b)–(3c). A PWA system is well-posed if $P_i \cap P_j = \emptyset$, $\forall i \neq j$, and $\bigcup_{i \in \mathbb{I}} P_i \subseteq \mathbb{R}^{n_c+m_c}$. In the discrete-time case, $\dot{x}_c(t)$ is replaced by $x_c(t+1)$ in (3a). In this paper we only focus on continuous-time PWA models.

Given an initial state x_0, an initial instant $\underline{t}_0$ and an input function $u_c : [\underline{t}_0, t_f] \to \mathbb{R}^{m_c}$, the PWA system (3) evolves as follows. Let i_0 be the active mode at $\underline{t}_0$, that is $i_0 \in \mathbb{I}$ such that (3b), (3c) are satisfied for $i(t) = i_0$, $x_c(t) = x_0$, $t = \underline{t}_0$ and $u_c = u_c(\underline{t}_0)$. For $t \in [\underline{t}_0, \overline{t}_0)$ where $\overline{t}_0 > \underline{t}_0$ is the smallest instant such that (3b), (3c) are not satisfied with $i = i_0$ (or, alternatively, for $t \in [\underline{t}_0, \overline{t}_0]$ where $\overline{t}_0 > \underline{t}_0$ is the largest instant up to which (3b), (3c) are satisfied non-stop with $i = i_0$), the state evolves according to the dynamics $\dot{x}_c(t) = A_{i_0} x_c(t) + B_{i_0} u_c(t) + f_{i_0}$. Thus, at time $\overline{t}_0 = \underline{t}_1$ the mode switches to the new active mode index $i_1 \in \mathbb{I}$ such that (3b), (3c) are satisfied for the current state and input vectors. Note that the trajectory $x_c(t)$ is continuous, since mode switches only introduce discontinuities in the state derivatives (no resets are considered so far).

For discrete-time PWA models, Boolean states, inputs and outputs are considered in [14], resulting in discrete-state transitions that only occur at multiples of the sampling period. The variables of the system in (3) become $x(t) = \left[\begin{smallmatrix} x_c(t) \\ x_b(t) \end{smallmatrix}\right] \in \mathbb{R}^{n_c} \times \{0,1\}^{n_b}$, $u(t) = \left[\begin{smallmatrix} u_c(t) \\ u_b(t) \end{smallmatrix}\right] \in \mathbb{R}^{m_c} \times \{0,1\}^{m_b}$, and the Boolean state update function is modelled as a mode-dependent constant (either 0 or 1, given by the truth-table of the state-update function of Boolean states). In [15] resets have been introduced for discrete-time PWA systems by exploting the equivalence with MLD systems [3].

Due to system trajectory discontinuities, the incremental form is more correct than the differential one to represent continuous-time PWA systems with discrete dynamics and resets (the trajectory is not differentiable at time instants $\overline{t}_0$, $\overline{t}_1$, $\ldots$). By extending the analysis from the discrete-time PWA case, the update of Boolean states, determined by a transition of the associated asynchronous finite state machine, can be defined in continuous-time as $x_b(t + dt) =$

[3]An output vector $y_c(t) = C_{i(t)} x_c(t) + D_{i(t)} u_c(t) + g_{i(t)} \in \mathbb{R}^{p_c}$ is also usually defined in dynamics (3).

$T^b_{i(t)}$, where dt is an infinitely small time interval and $T^b_{i(t)}$ is a mode-dependent binary vector. Particular care must be taken into account when using such a Boolean state update equation to avoid multiple switches at the same time instant. In fact, the system mode i depends on the Boolean state x_b, which in turn depends on i, with an infinitesimal delay. In [16], [17], the discrete dynamics have been introduced as an event-driven asynchronous finite state machine by exploiting the concept of events, associated to mode switches and changes of input values. The effect of such an approach is that both the discrete state and the system mode switch at event instants and remain constant until the next event.

Resets associated to mode switches can be modelled in the same way. We define additional *reset modes* $\mathbb{I}_r = \{s+1 \dots s_r\}$ [16] which are added to the normal *evolution modes* $\mathbb{I} = \{1 \dots s\}$ and we include additional terms in the continuous state dynamics equation (3a). The system evolves as follows. At time $\bar{t}$, let the state be $\bar{x}$, the input $\bar{u} = u(\bar{t})$, and assume the active mode $j \in \mathbb{I}$ becomes inactive. Then a reset mode $h \in \mathbb{I}_r$ is activated, the state is immediately reset to a value $\hat{x}$ which forces also the mode to change to an evolution mode $k \in \mathbb{I}$. In the incremental form the continuous state dynamics can be modelled as

$$\begin{aligned} x_c(t+dt) = (A_{i(t)}x_c(t) + B_{i(t)}u_c(t) + f_{i(t)})dt + \\ S_{i(t)}x_c(t) + R_{i(t)}u_c(t) + T_{i(t)}, \end{aligned} \tag{4}$$

where $A_i,\ B_i,\ f_i$ are zero for $i = s+1, \dots, s_r$, $R_i,\ T_i$ are zero and S_i is the identity for $i = 1, \dots, s$. Note that the first term in (4) is responsible for the continuous part of the state trajectory, while the remaining terms are responsible for discontinuities. For the ease of notation we reformulate the system dynamics using the "+" operator that indicates "immediately after" (i.e., $z^+(t) = z(t+dt)$). We obtain

$$\begin{aligned} \dot{x}_c(t) &= A_{i(t)}x_c(t) + B_{i(t)}u_c(t) + f_{i(t)}, && \text{(5a)}\\ x_c^+(t) &= \dot{x}_c dt + S_{i(t)}x_c(t) + R_{i(t)}u_c(t) + T_{i(t)}, && \text{(5b)}\\ x_b^+(t) &= T^b_{i(t)}, && \text{(5c)}\\ i(t) &: \quad H_{i(t)} \begin{bmatrix} x_c(t) \\ x_b(t) \end{bmatrix} + J_{i(t)}u_c(t) \le K_{i(t)}, && \text{(5d)}\\ & \quad\ i(t) \in \mathbb{I} \cup \mathbb{I}_r. \end{aligned}$$

The complete formulation would also require nonstrict inequalites in the mode selection (5d), we skip them here for compactness of notation. Clearly if the state trajectory is continuous (5) reduces to (3). Note also that possible Boolean exogenous inputs $u_b(t) \in \{0,1\}^{m_b}$ could be taken into account in (5d) to model externally-forced mode switches, but they are not needed in this paper to model LHA in PWA form.

IV. Translation of LHA in PWA Form

In this section we show that the LHA model (1) is a special case of the PWA model (5). For a given LHA generating trajectories $(x(t), v(t))$, we will construct a PWA system generating a trajectory (x_c, x_b), where $x(t) = x_c(t)$ and a proper binary encoding of $v(t)$ coincides with $x_b(t)$, $\forall t \in \mathbb{R}$.

A. Continuous and discrete states

The continuous states x_c of the equivalent PWA model are the continuous states x of the LHA, with dimension $n_c = n$.

The control modes $V = \{v_1, \dots, v_l\}$ map into Boolean states $x_b = \{0,1\}^{n_b}$. To do so, we introduce the encoding $cod : V \to \{0,1\}^{n_b}$ which associates to each $\bar{v} \in V$ a value $\bar{x}_b \in \{0,1\}^{n_b}$. The inverse $cod^{-1}\{0,1\}^{n_b} \to V$ may be a partial function (i.e., it may be undefined for some values on its domain). Thus, $\forall v \in V$, $x_b = cod(v)$ and $\forall x_b \in \{0,1\}^{n_b}$ such that $cod^{-1}(x_b)$ is defined, $v = cod^{-1}(x_b)$. A convenient choice for the function cod is the "one-hot" coding, namely the i-th vertex v_i is associated with $x_b = e_i$, where e_i is the i-th column of the identity matrix of order l. We assume such an encoding for the rest of this paper, and hence $n_b = l$.

When evolving in control mode v_j, the continuous state x of the LHA must satisfy the invariant condition $inv(v_j)$, namely $x \in \mathcal{IS}(j)$, where $\mathcal{IS}(j)$ is the *invariant set* for control mode v_j, that is the set of all $x \in \mathbb{R}^n$ such that $[inv(v_j)|X \leftarrow x] =$ TRUE. Since the inv predicate is defined by clauses constituted by linear inequalities, $\mathcal{IS}(j)$ is the polyhedron described by the inequalities

$$L_j x_c \le M_j, \tag{6}$$

where $L_j \in \mathbb{R}^{k_j \times n}$ and $M_j \in \mathbb{R}^{k_j}$, and k_j is the number of inequalities describing a (minimal) hyperplane representation of $\mathcal{IS}(j)$ (we avoid distinguishing between strict and nonstrict inequalities in (6) for compactness of notation). The continuous dynamics associated to v_j are defined by the zeroth-order linear differential inclusions (2). By introducing a constrained input $u_c(t) \in \mathbb{R}^n$ that models the uncertainty associated with the actual value of state derivatives, we transform (2) into

$$\begin{aligned} \dot{x}_{ci}(t) &= u_{ci}(t) && \text{(7a)}\\ \underline{p}_h^{(j)} &\le \sum_{k=1}^{N} q_{h,k}^{(j)} \cdot u_{ck} \le \overline{p}_h^{(j)}, \quad h = 1 \dots r_j, && \text{(7b)} \end{aligned}$$

which can be expressed as

$$\begin{aligned} \dot{x}_c(t) &= u_c(t) && \text{(8a)}\\ \underline{p}_j &\le Q_j u_c(t) \le \overline{p}_j, && \text{(8b)} \end{aligned}$$

where $Q_j \in \mathbb{R}^{r_j \times n}$ is the matrix whose (h,k)-th element is $q_{h,k}^{(j)}$ and $\underline{p}_j,\ \overline{p}_j \in \mathbb{R}^{r_j}$ are vectors whose h-th components are $\underline{p}_h^{(j)}$ and $\overline{p}_h^{(j)}$, respectively. Summarizing, in the j-th control mode the dynamics are described by

$$\begin{aligned} \dot{x}_c(t) &= u_c(t) && \text{(9a)}\\ x_b^+(t) &= cod(v_j) && \text{(9b)}\\ & \underline{p}_j \le Q_j u_c(t) \le \overline{p}_j, && \text{(9c)}\\ & L_j x_c(t) \le M_j, && \text{(9d)}\\ & -\alpha \le x_b(t) - cod(v_j) \le \alpha && \text{(9e)} \end{aligned}$$

until the mode eventually switches, where α is any vector of positive scalars smaller than $\frac{1}{2}$, and hence (9e) represents the condition $x_b(t) = cod(v_j)$.

B. Discrete transitions

So far we have described the dynamics of the continuous states of the LHA under the assumption that the control mode remains constant. In this case there is a one-to-one relation between the control modes of the LHA and the partitions of the equivalent PWA system, which are defined by the linear conditions (9c)–(9e). Dynamics (9) describes the trajectories of the continuous states of the LHA except for a zero-measure set of time instants, namely the set of switching instants $T = \{\underline{t}_0, \ldots, \underline{t}_n, \ldots\}$. In order to translate the discrete dynamics we need to introduce additional modes in the PWA system to select the successor discrete state, whenever a discrete transition occurs.

Assumption 1: $\forall (v_i, v_j) \in (V \times V)$, $jump(v_i, v_j) \equiv enab(v_i, v_j) \wedge res(v_i, v_j)$, where the free variables in *enab* are from X while the ones in *res* are from $X \cup X'$.

Assumption 2: $\forall j \in V$, $\forall x : [enab(i,j)|X \leftarrow x] =$ TRUE, $\exists \bar{x} : [inv(j)|X \leftarrow \bar{x}] \wedge [res(i,j)|X \leftarrow x, X' \leftarrow \bar{x}]$.

Assumption 1 states that the $jump$ predicate can be decomposed into two predicates, the first concerning the enabling of the discrete transition, the second concerning the reset after such a transition. The first predicate depends only on the actual state, the second depends on the actual and successor state. Since $jump$ is composed by linear inequalities, the same will be for both $enab$ and res. Note that Assumption 1 is usually satisfied for realistic systems, where the enabling of discrete transitions does not depend on the state after the transitions. Assumption 2 requires that when a transition is enabled, there exists at least one feasible successor state after the reset. Even this condition is quite natural for real systems, and, if not satisfied, usually the model of the system should be revised. However, while Assumption 1 is needed because of the transformation mechanism we introduce next, Assumption 2 is only needed to ensure that the system does not reach a deadlock.

We say that a transition $e = (v_i, v_j)$ is enabled at a state x if $[enab(v_i, v_j)|X \leftarrow x] =$ TRUE. Accordingly we define the enabling set of the transition from v_i to v_j as $\mathcal{ES}(i,j) = \{x_c \in \mathcal{IS}(i) : [enab(v_i, v_j)|X \leftarrow x_c] = \text{TRUE}\}$. A discrete transition will occur at any time instant t such that the transition is enabled, generating the following behavior. Let the system start from a state in which the discrete transition $\bar{e} = (v_i, v_j)$ is disabled, and evolve into a state at which $\bar{e}$ becomes enabled. Then the discrete transition will occur at any time instant, until $\bar{e}$ becomes disabled again.

In PWA systems mode switches are deterministic events that occur when the system state crosses the boundaries of the currently active region. A nondeterminism can be obtained by introducing further additional inputs acting as disturbances on region boundaries. By letting the disturbance variables change arbitrarily, the switching hyperplanes of the PWA systems move, causing transitions nondeterminism.

Example 1: An intuitive example of this behavior is shown in Figure 1 for a one-dimensional system, with state $x_c \in \mathbb{R}$ and initial value $x_c(t_0) = 0$. The state evolves in region v_0 with $\dot{x}_c = c > 0$ (c is a given scalar, so no additional inputs u_c are needed here to represent a differential inclusion). The control mode can switch $\forall x_c \in [x_m, x_M]$, and the invariant set for this mode is $\mathcal{IS} = \{x_c \in \mathcal{X} : 0 \le x_c \le x_M\}$. Let us introduce an additional input $w \in \mathbb{R}$. The partitions of the extended PWA system are defined in the lifted (x, w)-space $\mathbb{R}^2$. The region j in the extended PWA system is defined by $P_j = \{(x,w) \in \mathbb{R}^2 : 0 \le x \le x_m + w, 0 \le w \le x_M - x_m\}$. Let $w(t_0) = \bar{w}$ and assume that $w(t)$ remains constant up to next switching instant, that occurs when $x = x_m + \bar{w}$. For different values of $\bar{w}$, the mode switch occurs at different state values (see the two trajectories A and B shown in Figure 1), covering the whole range of possible values $[x_m, x_M]$ as w spans $[0, x_M - x_m]$. When P_j is projected back onto the state space $\mathbb{R}$, the partition can be decomposed into a region (thin line) in which the system certainly does not switch and a region (thick line) in which the system will switch at some time. ■

Consider a LHA in control mode v_j and let $\mathcal{IS}(j)$ be defined as in (6). Suppose that the control switches from v_j to v_i are enabled for $i = 1, \ldots, l_j$, and that, for simplicity, the enabling condition is the single linear inequality $h_{j,i} x_c > k_{j,i}$. We associate the *holding set* $\mathcal{HS}(j)$ to each control mode v_j, which is the set of continuous states such that all the linear inequalities in $enab(v_j, v_i)$, $\forall i = 1 \ldots l_j$, are false. Then, $\mathcal{HS}(j)$ is defined by the linear inequalities

$$h_{j,i} x_c \le k_{j,i}, \qquad i = 1 \ldots l_j, \tag{10a}$$

$$L_j x_c \le M_j. \tag{10b}$$

The continuous dynamics cannot change while $x_c \in \mathcal{HS}(j)$. However, $\mathcal{HS}(j)$ is in general only a subset of the region of the PWA system where mode j is active, because otherwise a transition would be forced to occur as soon as one of enabling conditions becomes true, which is not consistent with the general LHA semantics[4]. A discrete transition will occur at some time t such that $x_c(t) \in \mathcal{IS}(j) \setminus \mathcal{HS}(j)$. In order to represent such a nondeterminism in the exact transition by using a deterministic PWA model[5], again we introduce additional continuous input variables $\{w_{j,i}\}_{i=1}^{l_j}$ to relax constraints (10a) in $h_{j,i} x \le k_{j,i} + w_{j,i}$. The effect of $w_{j,i} \in [0, +\infty)$ is to enlarge the halfspace of a band having width $\frac{w_{j,i}}{\|h_{j,i}\|_2}$. Note that we have lifted the dimension of the PWA partition, thus obtaining polyhedra that are not overlapping. The projection $\mathcal{PS}(j)$ back onto the x-space of the set

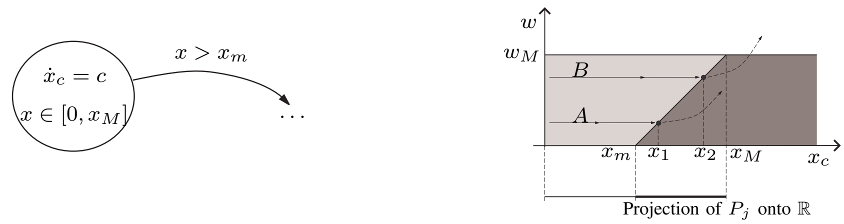
$$h_{j,i} x \le k_{j,i} + w_{j,i}, \quad i = 1 \ldots l_j \tag{11a}$$

$$L_j x \le M_j, \ w_{i,j} \ge 0 \tag{11b}$$

is clearly $\mathcal{IS}(j)$. Note that by fixing $w_{j,i} = \bar{w}_{j,i}$, (11) becomes a set in the x-space contained in between $\mathcal{HS}(j)$ (for $\bar{w}_{j,i} = 0$) and $\mathcal{IS}(j)$ (for $\bar{w}_{j,i} \to \infty$ or large enough, see Eq. (14) below).

[4] In verification tools based on linear hybrid or timed automata such as HyTech or UPPAAL it is often possible to specify transitions that occur as soon as they become enabled. Those are called *urgent events*. In case the regions of the PWA translation are defined as in (10), the PWA model represents an LHA in which all the transitions are urgent events.

[5] One could define PWA models with overlapping regions to represent the nondeterminism, but this is not in the spirit of this paper.



(a) LHA with an uncertain transition

(b) Equivalent PWA representation

Fig. 1. Uncertain LHA transitions can be represented by a PWA system with additional disturbance inputs

We now define in the lifted PWA space the discrete dynamics equivalent to the one of the LHA. Introduce $l_j + 1$ PWA modes associated to the LHA control mode v_j

$$P_{j,0} \;=\; \{(x_c, x_b, w) \in \mathbb{R}^{n+l+l_j} : (11) \text{ hold for } x = x_c \text{ and } -\alpha \leq x_b - cod(v_j) \leq \alpha\} \tag{12}$$

and for $f = 1, \ldots, l_j$

$$P_{j,f} \;=\; \{(x_c, x_b, w) \in \mathbb{R}^{n+l+l_j} :$$

$$h_{j,f} x_c > k_{j,f} + w_{j,f}, \tag{13a}$$

$$h_{j,i} x_c \leq k_{j,i} + w_{j,i},\ i = 1, \ldots, l_j,\ i \neq f \tag{13b}$$

$$L_j x_c \leq M_j,\ w_{i,j} \geq 0 \tag{13c}$$

$$-\alpha \leq x_b - cod(v_j) \leq \alpha\}. \tag{13d}$$

The above sets $P_{j,f}$, $f = 0, \ldots, l_j$ are polyhedral non-overlapping cells associated to the original control mode v_j of the LHA. Cell $P_{j,0}$ represents the situation in which the current mode remains active, while $P_{j,f}$ corresponds to the occurrence of the transition enabled by constraint $h_{j,f} x_c > k_{j,f}$, for $f = 1, \ldots, l_j$.

The definition of sets $P_{j,f}$ allows the PWA system to represent discrete transitions (v_j, v_f) that occur at any state value $x_c \in \mathcal{ES}(j, f)$. Consider a transition (v_j, v_f) occurring when $\bar{x}_c \in \mathcal{ES}(j, f)$. Let $\bar{w} \in \mathbb{R}_0^{l_j}$ be a vector such that $(\bar{x}_c, cod(v_j), \bar{w}_i) \in P_{j,f}$. Let $\underline{t}$ be the time instant at which the evolution in control mode v_j begins and $\bar{t} > \underline{t}$ be the time instant such that $x_c(\bar{t}) = \bar{x}_c$. Thus, assuming that $x_c(t) \in \mathcal{IS}(j)$ and $(x_c(t), x_b(t), w_j(t)) \in P_{j,0}$, $\forall t \in [\underline{t}, \bar{t}]$, if $w_j(\bar{t}) = \bar{w}$, at time $\bar{t}$ the system enters $P_{j,f}$.

Remark 1: Variables $w_{j,i}$ could be upper-bounded without affecting the definition of sets $P_{j,f}$, $f = 0, \ldots, l_j$. In fact, by solving the linear programming problem,

$$w_{j,i}^* = \sup_x \quad h_{j,i} x_c - k_{j,i} \tag{14a}$$

$$\text{subject to} \quad L_j x_c \leq M_j \tag{14b}$$

and by letting the upper-bound be $\bar{w}_{j,i} = w_{j,i}^*$ (with possibly $\bar{w}_{j,i} = +\infty$ if the set $\mathcal{IS}(j)$ is unbounded), the constraint $w_{j,i} \geq 0$ can be equivalently replaced by $0 \leq w_{j,i} \leq \bar{w}_{j,i}$. The supremum is taken in (14) to account for possible non-closed sets $\mathcal{IS}(j)$. ■

Remark 2: The above approach can be extended to more complex enabling conditions, provided that they can be expressed by the conjunction of linear inequalities. In this case variables $w_{j,i}$ become vectors with one component for each linear inequality in the predicate $enab(v_j, v_i)$. Note also that the enabling conditions $h_{j,i} x_c > k_{j,i}$ can be generalized to any combination of strict and nonstrict inequalities. ■

Example 2: Consider the LHA with two continuous states and four modes represented in Figure 2(a). Let the LHA control mode be 0 and assume it can switch to modes 1, 2 or 3. Thus, the lifted PWA system is defined in the 9-dimensional space $\mathbb{R}^{2+4+3}$ after introducing variables $w_{0,1}$, $w_{0,2}$ and $w_{0,3}$. Figure 2(b) shows the polyhedral partition projected onto the continuous state-space $\mathbb{R}^2$. In the holding set $\mathcal{HS}$ (white) the system cannot switch. The sets S_1 [S_2, S_3] correspond to $\mathcal{ES}(0, 1)$ [$\mathcal{ES}(0, 2)$, $\mathcal{ES}(0, 3)$] in which the system can either continue its evolution in mode 0 or switch to control mode 1 [2,3]. The value of the continuous state when the switch occurs depends on the value taken by w_1 [w_2, w_3]. In the darker set $S_{1,3}$ the system can either continue evolving in mode 0 or switch to control mode 1 or 3, depending on the value taken by variables $w_{0,1}$ and $w_{0,3}$. Note that even if the system behavior looks nondeterministic in the projected two-dimensional space, in the lifted space the behavior is deterministic with respect to both the "jump/not jump" decision and to the control mode after the jump. ■

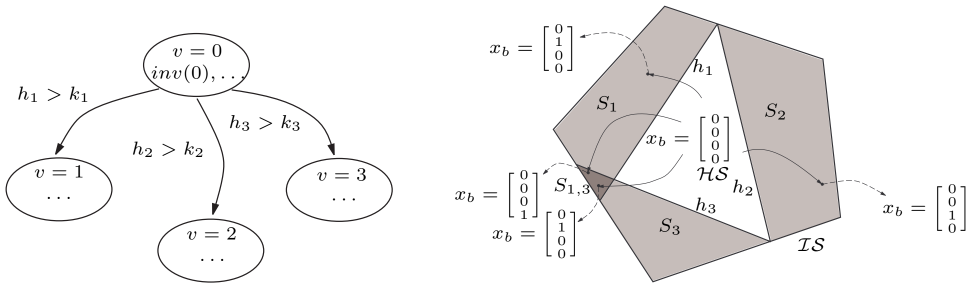
When the active region is $P_{j,0}$, the Boolean state x_b remains constantly equal to $cod(v_j)$, while when the system is in one of the regions $P_{j,f}$, $f = 1, \ldots, l_j$, the Boolean state changes. In a PWA model the successor Boolean state $x_b^+(t)$ can be easily defined by a region-dependent constant [14]. Let v_f be the Boolean state after the transition $e = (v_j, v_f)$ enabled by $h_{j,f} x_c > k_{j,f}$ occurs. Thus, we define the following discrete state-update piecewise constant dynamics

$$x_b^+(t) = \begin{cases} \text{cod}(v_j) & \text{if} \quad (x_c(t), x_b(t), w(t)) \in P_{j,0} \\ \text{cod}(v_f) & \text{if} \quad (x_c(t), x_b(t), w(t)) \in P_{j,f}, \\ & \qquad f = 1, \ldots, l_j. \end{cases} \tag{15}$$

The cells $P_{j,i}$ are also used to model resets:

$$x_c^+(t) \;=\; S_{j,f} x_c + T_{j,f} \text{ if } (x_c(t), x_b(t), w(t)) \in P_{j,f}, \quad j = 1, \ldots, n_b, \quad f = 0, \ldots, l_j \tag{16}$$

where $S_{j,i}$, $T_{j,i}$ define the reset condition and $S_{j,0} = I$, $T_{j,0} = 0$. In this case Equation (16) defines deterministic resets as affine functions of the state. Nondeterministic LHA resets can be represented by a technique similar to the one used in (7), thus by adding an additional input vector u_r which models the reset uncertainty.



(a) LHA switching behavior (b) PWA switching Behavior

Fig. 2. Uncertain transition representation in the extended PWA

C. Equivalent PWA reformulation

The PWA dynamics can be obtained by collecting the dynamics (9a), (16) of the continuous states, the dynamics (15) of the discrete states, the constraints in (9c), and the constraints in (12), (13), defining the switching conditions on x_c, x_b and w. Hence, the PWA model in the form (5) equivalent to the given LHA is

$$\left\{\begin{array}{rcl} \dot{x}_c & = & u_c \\ x_c^+ & = & S_\ell x_c + T_\ell \\ x_b^+ & = & b_\ell \end{array}\right. \quad \text{if} \quad \left\{\begin{array}{l} (x_c(t), x_b(t), w(t)) \in P_{j,i} \\ \underline{p}_j \preceq Q_j u_c \preceq \overline{p}_j \\ w_{j,i} \geq 0, \end{array}\right.$$
$$\ell = (j,i) \qquad j = 1, \ldots, l,\ i = 0, \ldots, l_j \text{ such that } (j,i) \in E \tag{17}$$

where for all $j = 1, \ldots, l$ we have $S_{(j,0)} = I$, $T_{(j,0)} = 0$, $b_{(j,0)} = cod(v_j)$, and $S_{(j,i)} = S_{j,i}$, $T_{(j,i)} = T_{j,i}$, $b_{(j,i)} = cod(v_i)$ for $i = 1, \ldots, l_j$. Only modes $\ell = (j,i) \in E$ are defined in (17), hence the number of partitions is card(V)+card(E).

The PWA system (17) is equivalent to the given LHA because for every state trajectory of the latter there exists a profile of w, u_c producing the same state trajectory through (17). Additional elements of the LHA, the $event$ function and the Σ set are mainly used for verification purposes, for associating variables to the transition, and are not considered here. They can be added as Boolean outputs associated to the sets $P_{j,i}$, which model the discrete transitions. The $init$ set can be defined by additional constraints on the initial state of the PWA model, possibly introducing an *initial mode*.

V. Discussion

In this paper we have proposed a way for representing a linear hybrid automaton as a piecewise affine system with discrete dynamics and resets. This result creates a bridge between the hybrid models exploited in computer science and hybrid models exploited in control theory. The practical advantages are related to the possibility of exploiting different models in different design phases of a hybrid control system. The LHA can be recovered from the equivalent PWA system (17) by performing the operations described in the previous sections backwards. The problem of generating a LHA from a generic PWA system (5) will be investigated in future research, based on the insight provided by this paper.

References

[1] E. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Trans. Automatic Control*, vol. 26, no. 2, pp. 346–358, April 1981.

[2] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[3] W. Heemels, B. de Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–1091, July 2001.

[4] T. Henzinger, "The theory of hybrid automata," in *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS '96)*, New Brunswick, New Jersey, 1996, pp. 278–292.

[5] J. Lygeros, K. Johansson, S. Simic, J. Zhang, and S. Sastry, "Dynamical properties of hybrid automata," *IEEE Trans. Automatic Control*, vol. 48, pp. 2–17, Jan. 2003.

[6] T. Henzinger, P. Ho, and H. Wong-Toi, "HyTech: A model checker for hybrid systems," *International Journal on Software Tools for Technology Transfer*, vol. 1, no. 1–2, pp. 110–122, 1997.

[7] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[8] T. Henzinger, P. Ho, and H. Wong-Toi, "A user guide to HyTech," in *Tools and Algorithms for Construction and Analysis of Systems*, 1995, pp. 41–71.

[9] J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, Y. Wang, and C. Weise, "New Generation of UPPAAL," in *Int. Workshop on Software Tools for Technology Transfer*, June 1998.

[10] M. Johannson and A. Rantzer, "Computation of piece-wise quadratic Lyapunov functions for hybrid systems," *IEEE Trans. Automatic Control*, vol. 43, no. 4, pp. 555–559, 1998.

[11] A. Juloski, W. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J. Niessen, "Comparison of four procedures for the identification of hybrid systems." in *Hybrid Systems: Computation and Control*, 2005, pp. 354–369.

[12] E. Asarin, O. Bournez, T. Dang, and O. Maler, "Reachability analysis of piecewise-linear dynamical systems," in *Hybrid Systems: Computation and Control*. Springer-Verlag, 2000, pp. 20–31.

[13] R. Alur, C. Courcoubetis, T. Henzinger, and P.-H. Ho, "Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems," in *Hybrid Systems*. Springer-Verlag, 1993, pp. 209–229.

[14] A. Bemporad, "Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form," *IEEE Trans. Automatic Control*, vol. 49, no. 5, pp. 832–838, 2004.

[15] F. Torrisi and A. Bemporad, "HYSDEL — A tool for generating computational hybrid models," *IEEE Trans. Contr. Systems Technology*, vol. 12, no. 2, pp. 235–249, Mar. 2004.

[16] A. Bemporad, S. Di Cairano, and J. Júlvez, "Event-driven optimal control of integral continuous-time hybrid automata," in *Proc. 44th IEEE Conf. on Decision and Control*, Seville, Spain, 2005, pp. 1409–1414.

[17] ——, "Event-based model predictive control and verification of integral continuous-time hybrid automata," in *Hybrid Systems: Computation and Control*. Springer-Verlag, 2006, pp. 93–107.