Event-Driven Optimal Control of Integral Continuous-Time Hybrid Automata

Alberto Bemporad†, Stefano Di Cairano†◇ and Jorge Júlvez‡

***Abstract*— This paper proposes an *event-driven* optimal control strategy for hybrid systems composed of continuous-time integral dynamics with inputs and of finite state machines triggered by endogenous and exogenous events. Endogenous events are caused by continuous states or continuous inputs crossing certain linear thresholds or by the elapse of time intervals, while exogenous events are forced by changes of binary and continuous inputs. The advantages of the proposed strategy are the reduction of the amount of computation required to solve optimal control problems, and the reduction of approximation errors typical of discrete-time approaches. We examine several performance objectives and constraints that lead to mixed-integer linear or quadratic optimization problems and we exemplify the approach on a simple example.**

I. Introduction

Optimal control aims at finding the input profile that generates the optimal behavior of a dynamical system with respect to a given performance index while respecting a set of constraints. For hybrid systems [1], this problem is in general hard to solve if a continuous-time model is considered [2]–[4], while it can be efficiently solved by Mixed Integer Programming (MIP) techniques if the system is modelled in discrete-time [5]. Nevertheless, sampling continuous-time systems introduces errors, that in case of hybrid dynamics might become large, due to discontinuities of the state update equations. The standard approach to reduce this effect is to increase the sampling rate, therefore requiring more samples to control the system during a given time interval. This may not be computationally feasible given that in general the computational complexity to obtain the optimal control profile increases exponentially with respect to the number of sampling steps. Also, when dealing with systems with different operative modes, a uniform (small) sampling time must be chosen basing on the fastest mode, often resulting in oversampling.

In the literature several mathematical models have been proposed for hybrid systems. This paper focuses on a subclass of continuous-time Piecewise Affine (PWA) [6] systems, closely related to Linear Hybrid Automata (LHA) [7], [8]. We propose a different approach to control such systems exploiting preliminary ideas presented in [9] for continuous Petri nets [10]: time-driven control is substituted by event-driven control. In contrast with discrete-time methods, in this approach events do not happen when a fixed amount of time (the sampling period) elapses, but when the mode of the system or the input value changes. Thus, events only happen when something perturbs the linearity of the continuous dynamics. The main advantages of the approach are the following: 1) it minimizes the number of variables required to compute a trajectory of the system during a given time interval; 2) trajectory errors due to mode switches at intersampling instants are eliminated.

We shall better explain the importance of the second one. In a discrete-time model, the mode is assumed constant during the whole sampling period. However, when the real system is continuous-time, this assumption is, in general, no longer true. We call the error generated by this wrong assumption *mode-mismatch error*. Figure 1 reports an example of a large trajectory deviation due to mode-mismatch error. The system is a two-dimensional piecewise affine system defined over polyhedral regions. The initial state is $x_0 = [1\ 1]^T$, the desired final state is $x_f = [6\ 3.5]^T$. A discrete-time model of the system is obtained with sampling period $T_s = 1$. The discrete-time model is able to reach the target state x_f from x_0 in one step by applying the input $u^* = 5$. However, when u^* is applied constantly during the time interval $[0, 1]$ the behavior of the original continuous-time system is different from the discrete-time one. The straight dashed line is the trajectory of the discrete-time model, while the solid line is the corresponding continuous-time trajectory. The value of the mode $i(t)$ is not constant during the whole sampling period and the large difference is due to the fast dynamics in the region containing x_f. The problem can be mitigated by reducing the sampling period, but this generates an useless amount of computation when the system is in the first region, where the dynamics is slow. A better approach would be to change the input exactly when the mode changes, as we detail in the rest of the paper.

Event-driven control approaches were applied successfully to switched piecewise affine systems [11], where the switches between different dynamics are triggered by external decisions and the mode dynamics are autonomous. Switched time optimal control of systems with a similar continuous state dynamics has been proposed in [12]. The approach presented here differs from [12] for two reasons: we consider systems with multiple inputs and constraint dependent switches rather than autonomous systems with controlled switches, and we propose mixed integer programming instead of linear programming and feasible sequence enumeration as the solution technique.

The rest of the paper is structured as follows: Section II presents an abstract model of a continuous hybrid system that can be formulated as an event-driven system. An event-driven

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

†Dipartimento di Ingegneria dell'Informazione, Università di Siena, `bemporad,dicairano@dii.unisi.it`.

‡Dep. Informática e Ingeniería de Sistemas, Centro Politécnico Superior de Ingenieros, Universidad de Zaragoza, `julvez@unizar.es`

◇ Corresponding author.

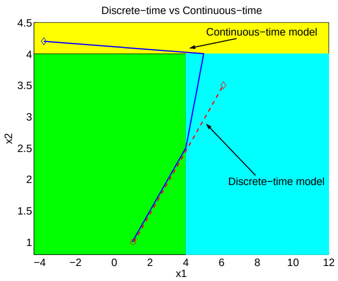

Fig. 1. State trajectory error introduced by mode-mismatch between discrete-time (dashed) and continuous-time (solid)

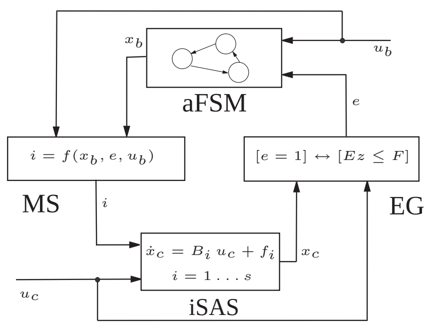

Fig. 2. Integral continuous time Hybrid Automaton (icHA)

formulation for the PWA system is reported in Section III. In Section IV several criteria for optimal control are proposed and Section V presents an example.

II. Integral Continuous Hybrid Automaton

A continuous-time version of the Discrete Hybrid Automaton (DHA) [13] is considered here, in which the continuous dynamics are integral and which will be called *integral continuous (time) Hybrid Automaton* (icHA). As the DHA, the icHA is constituted by four components shown in Figure 2: an integral Switched Affine System (iSAS), an Event Generator (EG), a Mode Selector (MS) and an asynchronous Finite State Machine (aFSM). The continuous-time iSAS represents a collection of possible integral dynamics (i.e., the system modes) for the continuous state,

$$\dot{x}_c(t) = B_{i(t)} u_c(t) + f_{i(t)}, \tag{1}$$

in which $x_c \in \mathbb{R}^{n_c}$ and $u_c \in \mathbb{R}^{m_c}$ are the continuous components of the state and input vectors, respectively, and $i \in \mathcal{I} = \{1, 2, \ldots s\}$ is the system mode. The EG is defined by the linear threshold conditions

$$[e_i^x(t) = 1] \leftrightarrow \left[E_i^x \left[\begin{smallmatrix} x_c(t) \\ t \end{smallmatrix}\right] \leq F_i^x\right], \; i = 1, \ldots n_e^x \tag{2a}$$
$$[e_i^u(t) = 1] \leftrightarrow \left[E_i^u \left[\begin{smallmatrix} u_c(t) \\ t \end{smallmatrix}\right] \leq F_i^u\right], \; i = 1, \ldots n_e^u \tag{2b}$$

where $n_e^x + n_e^u = n_e$ and $e = [e_1^x \ldots e_{n_e^x}^x \; e_1^u \ldots e_{n_e^u}^x]^T \in \{0, 1\}^{n_e}$ is the vector of endogenous binary input variables. The discrete state dynamics are defined by an aFSM

$$x_b(t_{k+1}) = f_{\mathrm{aFSM}}(x_b(t_k), u_b(t_k), e(t_k)), \tag{3}$$

in which $x_b \in \{0, 1\}^{n_b}$ and $u_b \in \{0, 1\}^{m_b}$ are the discrete components of state and input vectors, respectively, $f_{\mathrm{aFSM}} : \{0, 1\}^{n_b + m_b + n_e} \rightarrow \{0, 1\}^{n_b}$ is a Boolean function, t_k is the instant of the k^{th} event, thus $t_k = \inf_{t > t_{k-1}}\{t : (u_b(t), e(t)) \neq (u_b(t_{k-1}), e(t_{k-1}))\}$, and for $t_k < t \leq t_{k+1}$, the binary state is kept constant at $x_b(t) = x_b(t_{k+1})$. The discrete dynamics reacts to events that are asynchronous variations of the endogenous binary inputs e and of the exogenous binary inputs u_b, and the time instants t_k represent the instants in which some of those signals change, causing a switch on the binary state x_b, delayed by an infinitesimal time. The value of $i(t)$ in Equation (1) is assigned by the Mode Selector (MS) through the scalar product:

$$i(t) = [1\, 2 \ldots s] \cdot f_{\mathrm{MS}}(x_b(t), u_b(t), e(t)), \tag{4}$$

where $f_{\mathrm{MS}} : \{0, 1\}^{n_b + m_b + n_e} \rightarrow \{0, 1\}^s$ is a Boolean function and $[1 \ldots 1] \cdot f_{\mathrm{MS}} = 1$, $\forall (x_b(t), u_b(t), e(t)) \in \{0, 1\}^{n_b + m_b + n_e}$.

A. Integral Piecewise Affine Systems

The icHA is an abstract model that can be converted into a computational continuous-time model called integral Piecewise Affine system. Continuous-time Piecewise Affine systems are dynamical models in which the state space is partitioned into regions and in each region the dynamics are defined by affine differential equations.

Definition 1: An integral Piecewise Affine (iPWA) systems is a PWA system that has integral dynamics in each mode:

$$\begin{aligned}
\dot{x}_c(t) \; &= \; B_{i(t)} u_c(t) + f_{i(t)}, && \text{(5a)}\\
x_b(t^+) \; &= \; g_{i(t)}, && \text{(5b)}\\
i(t) \in \mathcal{I} \; &: \; H_{i(t)}^x x(t) + J_{i(t)}^x u_b(t) + W_{i(t)}^x t \leq K_{i(t)}^x, && \text{(5c)}\\
& \; H_{i(t)}^u u_c(t) + J_{i(t)}^u u_b(t) + W_{i(t)}^u t \leq K_{i(t)}^u. && \text{(5d)}
\end{aligned}$$

The partition of the full state $x(t) = [x_c(t) \;\; x_b(t)]^T$ is polyhedral and depends also on the binary input u_b. Constraints (5c) and (5d) involve only x and u_c, respectively, because of definition (2) of endogenous binary inputs. Note that the binary state update function (5b), in which $x_b(t^+) = \lim_{\tau \rightarrow 0^+} x_b(t + \tau)$, follows by embedding (3) in the inequalities (5c) and (5d), so that the binary state is obtained as a mode-dependent constant [14]. We assume that there is an infinitesimal delay between the evaluation of the mode and the update of the binary state, thus the discrete-state trajectory is left-continuous. This is not reductive since in any physical systems the components whose state is modelled by x_b have a finite trigger time (consider for instance CMOS logic gates). The translation of icHA into a PWA form results in an iPWA system; such a translation can be performed as in [15], by enumerating all the feasible combinations of discrete valued variables and by finding the corresponding polyhedral cells.

III. Event-driven model of integral Continuous Hybrid Automata

In this paper event-driven optimal control for icHA/iPWA systems is proposed, in which both the input values and the durations between two consecutive events are decision variables, under possible operating constraints. As a result, the number of different control values per time unit is not constant as in discrete-time control, but it changes with the event rate. As a new input is computed only when an event occurs, the computation effort is certainly smaller than the discrete-time approach. Furthermore, the mode-mismatch error is eliminated, since when the mode changes the input is recomputed, according to a possibly new dynamical mode.

Model (5) must be converted into a suitable form that can be tackled by Mixed Integer Programming (MIP) techniques [16]. Let $\mathcal{PC}_{(m_c,m_b)}$ be the set of piecewise constant functions $u=\left[\begin{smallmatrix}u_c\\u_b\end{smallmatrix}\right]$, $u:\mathbb{R}\rightarrow\mathbb{R}^{m_c}\times\{0,1\}^{m_b}$ such that $u(t)$ is constant $\forall t\in[t_k,t_{k+1})$, $k=0,1,\ldots$ where $t_0<t_1<...$ is a sequence of time instants. In the following we will assume the input function $u_c:\mathbb{R}\rightarrow\mathbb{R}^{m_c}\in\mathcal{PC}_{(m_c,0)}$[1].

A. Event-Driven Dynamics of Continuous States

Consider a PWA system whose continuous state dynamics are defined by $\dot{x}_c(t)=A_{i(t)}x_c(t)+B_{i(t)}u(t)+f_{i(t)}$ and consider a time interval $[t_0,t_f]$ in which the system does not switch mode; $\forall t\in[t_0,t_f]$ the trajectory is defined by $\dot{x}_c(t)=Ax_c(t)+Bu_c(t)+f$, where $A=A_{i(t_0)}$, $B=B_{i(t_0)}$, $f=f_{i(t_0)}$ are computed by (5c) and (5d). The closed form solution for $x_c(t),\ t\in[t_0,t_f]$ is well known.

Consider a signal $u_c:\mathbb{R}\rightarrow\mathbb{R}^{m_c}$ such that $u_c\in\mathcal{PC}$, let $t_0<t_1<\ldots<t_h=t_f$ and let $\{u_k\}_{k=0}^{h-1}$ be the associated constant values. The state value at t_f is

$$x_c(t_f)=e^{A(t_f-t_0)}x_c(t_0)+\sum_{k=0}^{h-1}\int_{t_k}^{t_{k+1}}e^{A(t_f-\tau)}\mathrm{d}\tau\,Bu_k+\int_{t_k}^{t_{k+1}}e^{A(t_f-\tau)}\mathrm{d}\tau\,f,\quad (6)$$

which is an affine function of the input values u_k, but is nonlinear with respect to the time instants t_k. Hence, the optimal control problem for a system whose dynamics are defined by (6) and where both u_k and t_k are decision variables is in general nonconvex [2]. If the interest is restricted to systems that have an integral dynamics ($A=0$), Equation (6) becomes $x_c(t_f)=x_c(t_0)+\sum_{k=0}^{h-1}B(t_{k+1}-t_k)u(t_k)+\sum_{k=0}^{h-1}(t_{k+1}-t_k)f(t_k)$, since $u\in\mathcal{PC}$. Consider now an iPWA system, choose the input function $u\in\mathcal{PC}$ in such a way that the mode switches occur only at the instants t_k and consider instants t_k as the instants at which events occur. Thus, the state trajectory in the interval $[t_0,t]$, where $t=t_h$, is

$$x_c(t)=x_c(t_0)+\sum_{k=0}^{h-1}\Big(B_{i(t_k)}(t_{k+1}-t_k)u_c(t_k)+f_i(t_k)(t_{k+1}-t_k)\Big),\quad (7)$$

where $k\in\{0,1,\ldots\}$ is the event counter. The events can be endogenous or exogenous. The endogenous events are due to switches on system dynamics, while the exogenous events are forced on purpose by the controller. In both cases, at the event instants the input levels can change. The event-based formulation of the state update equation (7) becomes

$$\begin{aligned} x_c(k+1) &= x_c(k)+B_{i(k)}q(k)u_c(k)+f_{i(k)}q(k) && (8a)\\ t(k+1) &= t(k)+q(k) && (8b)\end{aligned}$$

where k is the event counter, $x(k)=x(t_k)$, $q(k)$ is the time interval between events k and $k+1$, and $u(k)$ is the input value during such an interval. The additional state variable $t(k)$ represents time expressed as a function of the event index, thus $t(k)=t_k$ is the time instant at which the k^{th} event occurs. Equation (8a) is still nonlinear because of the product between variables $q(k)$ and $u_c(k)$, but as in [9] it is possible to introduce a new vector $v(k)=q(k)u_c(k)$ representing the whole effect of the input between two consecutive events ($v(k)$ is the integral over time period $q(k)$ of the input $u_c(k)$). Thus the controlled variables are the input integral $v(k)$ and the input duration $q(k)$. The input $u_c(k)=\frac{v(k)}{q(k)}$ applied to the system is computed from them.

The relations between q and v come from the constraints that define the partitions (5c), (5d), from possible operating constraints on states and inputs, and from the cost function that will be specified in the sequel.

B. Generation of Endogenous Events

Assume the system is in mode $\bar{\imath}$ during $[t_k,t_{k+1})$. Suppose for the sake of simplicity that mode $\bar{\imath}$ is associated with the polyhedral region

$$H_{\bar{\imath}}x_c(t)\leq K_{\bar{\imath}},\quad (9)$$

a particular case of (5c), (5d) in which inequalities involve only continuous states (the general case can be obtained straightforwardly). The event occurring at t_{k+1} is generated when one of the inequalities in (9) is violated, thus all of them must be satisfied on $[t_k,t_{k+1})$:

$$(i(t)=\bar{\imath})\Rightarrow H_{\bar{\imath}}x_c(t)\leq K_{\bar{\imath}},\forall t\in[t_k,t_{k+1}).\quad (10)$$

Because of the integral dynamics of iPWA systems, constraint (10) can be enforced in the event-driven model (15) by adding the mixed logical/continuous constraint

$$(i(k)=\bar{\imath})\Rightarrow H_{\bar{\imath}}(x_c(k)+B_{\bar{\imath}}v(k)+f_{\bar{\imath}}q(k))\leq K_{\bar{\imath}}+\varepsilon_{\bar{\imath}}\mathbf{1}.\quad (11)$$

where $\varepsilon_{\bar{\imath}}>0$ is an arbitrarily small tolerance, e.g. the machine precision, and $\mathbf{1}=[1\ldots1]^T$. Constraint (11) implies that if the mode at t_k is $\bar{\imath}$, at t_{k+1} the system state is within the region associated with mode $\bar{\imath}$, enlarged by $\varepsilon_{\bar{\imath}}$.

The tolerance $\varepsilon_{\bar{\imath}}$ can be seen as the mathematical instrument to slightly enlarge region $\bar{\imath}$, thus allowing the system to cross the borders that define such a region. Consider a single inequality $\bar{h}x_c\leq\bar{k}$. The effect of adding $\varepsilon_{\bar{\imath}}$ at the right-hand side of (11) is to move outwards the halfspace by $\frac{\varepsilon_{\bar{\imath}}}{\|\bar{h}\|_2}$. Thus, the maximal crossing distance in each axis

[1] For the sake of notation the subscript (m_c,m_b) will be dropped in the rest of the paper when the value of (m_c,m_b) is clear from the context.

direction is $\frac{\varepsilon_{\bar{\imath}}}{(h\cdot\phi_l)}$, where $\phi_l, l=1,\ldots,n_c$ are the unitary axis-directed vectors.

The tolerance introduces a tiny mode-mismatch error that can be reduced as much as desired by taking a small enough ε_i, without increasing the solution complexity. The following proposition proves that (11) forces the state to remain in the region obtained by enlarging region $\bar{\imath}$ by $\varepsilon_{\bar{\imath}}$, up to the next event.

Proposition 1: If inequality (9) is satisfied for $x_c(k)$, and $v(k)$ and $q(k)$ are such that (11) is satisfied, then

$$H_{\bar{\imath}} x_c(t) \leq K_{\bar{\imath}}, +\varepsilon_{\bar{\imath}}\mathbf{1},\ \forall t\in[t_k, t_{k+1}]. \tag{12}$$

Proof: Since $x_c(t_k)=x_c(k)$ and $\varepsilon_{\bar{\imath}}>0$, (12) is satisfied for $t=t_k$. Since $x_c(t_{k+1})=x_c(k+1)=x_c(k)+B_{\bar{\imath}}v(k)+f_{\bar{\imath}}q(k)$, if (11) is satisfied, then also (12) is satisfied for $t=t_{k+1}$. Finally, (12) holds $\forall t\in[t_k,t_{k+1}]$ because the state trajectory $x_c(t),\ t\in[t_k,t_{k+1}]$ is a linear function of time and the inequality (12) defines a convex set. ■

As a consequence of Proposition 1, the state trajectory belongs to region $\bar{\imath}$ during $[t_k, t_{k+1}-\sigma]$, where $\sigma>0$ depends on $\varepsilon_{\bar{\imath}}$, and $\sigma\to 0$ as $\varepsilon_{\bar{\imath}}\to 0$.

At t_{k+1} the system might switch mode or maintain the same mode. The second case happens when the inequality (9) is still satisfied at t_{k+1}, which means that an exogenous event occurred; otherwise $i(t_{k+1})\neq i(t_k)$ and the event was an endogenous one.

C. Operating Constraints

Operating constraints on the state can be embedded in the set of constraints (9). Input bounds $\underline{u}\leq u_c(t)\leq\overline{u}$ can be written as the linear constraints

$$\underline{u}q(k)\leq v(k)\leq\overline{u}q(k). \tag{13}$$

Different input bounds for different modes can be enforced as $[i(k)=\bar{\imath}]\to[\underline{u}_{\bar{\imath}}q(k)\leq v(k)\leq\overline{u}_{\bar{\imath}}q(k)]$, where $\underline{u}_{\bar{\imath}}$ and $\overline{u}_{\bar{\imath}}$ are the input upper and lower bounds while in mode $\bar{\imath}$. Additional operating constraints may be imposed on time intervals between two events

$$\underline{q}\leq q(k)\leq\overline{q}. \tag{14}$$

For example a finite $\overline{q}$ imposes a maximum time for each control action, in order to prevent the system from running in open loop for too long. A minimum duration $\underline{q}$ ensures a minimum time interval between two events (and thus between two mode switches), therefore avoiding undesirable effects such as high frequency chattering and Zeno behaviors [17].

D. Discrete Dynamics

When the event-driven model is obtained from model (5) the discrete state dynamics is directly defined by the polyhedral partition (5c), (5d) and by the mode-dependent constant vectors in (5b) (see [14]). If the discrete dynamics is formulated as an event-driven FSM it can be reformulated as a set of mixed-integer (in)equalities as described in [13]. Such (in)equalities can be directly embedded in the optimization problem described later in Section IV, or used to compute a PWA representation of the form (15) defined below by using the approach of [15].

The endogenous events generated by discrete state transitions are expressed exactly as the ones in Equation (11), since discrete-state transitions are also triggered by linear thresholds on states and inputs, and by exogenous events caused by changes of binary inputs u_b.

E. Event-Driven PWA Formulation

By collecting (5), (11), (13), (14) an event-driven integral PWA system can be written as:

$$\begin{aligned}
x_c(k+1) &= x_c(k)+B_{i(k)}v(k)+f_{i(k)}q(k) && \text{(15a)}\\
t(k+1) &= t(k)+q(k) && \text{(15b)}\\
x_b(k+1) &= g_{i(k)} && \text{(15c)}\\
i(k)\ &:\ H_{i(k)}\begin{bmatrix} x(k)\\ t(k)\end{bmatrix}+J_{i(k)}\begin{bmatrix} v(k)\\ q(k)\\ u_b(k)\end{bmatrix}\leq K_{i(k)}. && \text{(15d)}
\end{aligned}$$

Equations (15a) and (15c) define the evolution of continuous and discrete states, respectively, in the event-driven framework. Equation (15b) defines the time evolution and (15d) defines the polyhedral partition after the intersection with operating and event-generation constraints. Note that k is the event counter and $t(k)$ is the corresponding time instant.

F. Resets

Discontinuities of the continuous state trajectory can be introduced by resets. Additional *reset modes* $i\in\{s+1,\ldots,s_r\}$ are included, (15a) is modified into $x_c(k+1)=(E_i x_c(k)+h_i)+B_{i(k)}v(k)+f_{i(k)}q(k)$, and (15b) into $t(k+1)=t(k)+G_i q(k)$. In modes $i=\{1\ldots s\}$, $E_i=I_{(n_c\times n_c)}$ (where I is the identity matrix), $h_i=\mathbf{0}_{(n_c\times 1)}$ and $G_i=1$, while in reset modes $i=\{s+1\ldots s_r\}$ $B_i=\mathbf{0}_{(n_c\times m_c)}$, $f_i=\mathbf{0}_{(n_c\times 1)}$ and $G_i=0$. Note that resets are instantaneous.

IV. Event-Driven Optimal Control

Consider the optimal control problem

$$\begin{aligned}
\min_{q,v}\quad & J(x,t,q,v) && \text{(16a)}\\
\text{s.t.}\quad & \text{system dynamics (15)} && \text{(16b)}\\
& g(x,t,q,v)\leq 0\,, && \text{(16c)}\\
& x(0)=x_0,\quad t(0)=t_0. && \text{(16d)}
\end{aligned}$$

where J is a convex function of (x,t,q,v), $t=\{t(k)\}_{k=0}^N$ are the time instants at which the events occur, $x=\{x(k)\}_{k=0}^N$ are the corresponding state values, $q=\{q(k)\}_{k=0}^{N-1}$ are the durations of the time intervals between two consecutive events and $v=\{v(k)\}_{k=0}^{N-1}$ are the input integrals during $[t_k,t_{k+1})$. Constraint (16c) represents possible additional constraints in the optimal control problem and N is the number of allowed events.

By converting model (15) into an equivalent MLD model [15], problem (16) can be solved by mixed integer programming for different objective functions (16a) and constraints (16c).

Typically it is required that the system state reaches a desired target state x_f after (at most) N events,

$$x(N) = x_f. \tag{17}$$

Sometimes, to prevent infeasibility of the optimal control problem, it is useful to soften (17) by adding instead

$$\rho \|x(N) - x_f\|_\infty \tag{18}$$

in (16a), where ρ is a large weighting constant. In alternative, one can consider a convex desired target set $\mathcal{X}_f$ and impose the constraint $x(N) \in \mathcal{X}_f$ (either as a hard or a soft constraint). A target time can be formulated similarly.

The objective criterion in (16a) defines how the target state must be reached. Minimum time, minimum effort, and minimum displacement criteria are considered in this paper. The *Minimum-Time* criterion looks for the sequence (q, v) that minimizes the time needed to bring the system from x_0 to the final state x_f. This can be obtained by minimizing

$$J(x, t, q, v) = \sum_{k=0}^{N-1} q(k). \tag{19}$$

Note that the v variables are not involved in (19), but still are involved in (17).

The *Minimum-Effort* criterion looks for the sequence (q, v) that brings the system to the final state x_f or target set $\mathcal{X}_f$ with the minimum control effort. Let $J(x, t, q, v) = \int_0^{t_N} |u(t)|\mathrm{d}t$ be the L_1 norm of the input function, which provides a measure of the effort to control the system. With respect to the event instants $\{t_k\}_{k=0}^N$, $\|u(\cdot)\|_{L_1} = \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} |u(t)|\mathrm{d}t$, and since u is constant in each period $[t_k, t_{k+1})$, we obtain

$$J(x, t, q, v) = \|u(\cdot)\|_{L_1} = \sum_{k=0}^{N-1} q(k)|u(k)| = \sum_{k=0}^{N-1} |v(k)| \tag{20}$$

where $u(k)$ is the control level chosen at the k^{th} event, $q(k) = t(k+1) - t(k)$ is the period duration, and $v(k)$ is the input integral as defined in Section III-A.

The *Minimum-Displacement* criterion looks for the trajectory that brings the state to the target state x_f (or to the target set $\mathcal{X}_f$), while minimizing the maximum displacement from a desired continuous state trajectory r. This amounts to minimize the L_∞ norm of the state deviation

$$\begin{aligned} &J(x, t, q, v) = \|x_c(\cdot) - r(\cdot)\|_{L_\infty} = \\ &\max_{t \in [t_0, t_f]} |x_c(t) - r(t)|_\infty = \max_{k \in [0, N]} |x_c(k) - r(k)|_\infty, \end{aligned} \tag{21}$$

where $t_f = t(N)$, $r \in \mathcal{PC}_{n_c}$ and the changes of the value of r are registered as exogenous events. A special case is $r(t) = x_f$, $t \in [t_0, t_f]$. The last equality in (21) is a consequence of the following proposition.

Proposition 2: Let $x_c(t)$, $\forall t \in [t_0, t_f]$, be the continuous state trajectory of an iPWA system, $t_0 < t_1 < \ldots < t_h = t_f$ be the switching instants[2] and $r(t)$ be piecewise constant over $[t_0, t_f]$. Then $\max_{t \in [t_0, t_f]} \|x_c(t) - r(t)\|_\infty = \max_{k \in [0, h-1]} \{\|x_c(t_k) - r(t_k)\|_\infty\}$.

Proof: $\max_{t_0 \le t \le t_f} \|x_c(t) - r(t)\|_\infty = \max_{0 \le k \le h-1} \{ \max_{t \in [t_k, t_{k+1}]} \|x_c(t) - r(t)\|_\infty\}$. Function $\|x_c(t) - r(t)\|_\infty$ is convex on $[t_k, t_{k+1}]$, since it is the composition of a convex function (the norm defined over a bounded polyhedron) with a linear function (the state trajectory of the iPWA system between two consecutive switches). Thus, it attains its maximum value at the border, either at t_k or at t_{k+1}. Since the state values at the mode switch instants t_k are contained in the set of state values at the event instants $\{x(k)\}_{k=0}^N$, also the last equality in (21) holds. ∎

V. Numerical Example

In this section we consider "train–gate" example [8], a well known benchmark for hybrid systems, commonly used for verification purposes but proposed here as a control problem. We define the dynamics by non-autonomous differential equations $\dot{x}_i(t) = u(t) + f_i$, with $a_i - f_i \le u_i \le b_i - f_i$, instead of the differential inclusions of [8] used for verification. Both examples were modelled in HYSDEL [13] (see `http://www.dii.unisi.it/~dicairano/sources/eventdriven.html`). We performed the tests on a Pentium Xeon 2.8 GHz with 1 GByte RAM running Cplex 9.0 and Matlab 6.5.

Ar	$x_1 \le -10$	$f_1 = 0.5$	$-0.05 \le u_1 \le 0.05$
Cr	$-10 < x_1 \le 10$	$f_1 = 0.42$	$-0.08 \le u_1 \le 0.08$
L	$x_1 > 10$	$f_1 = 0.8$	$-0.05 \le u_1 \le 0.05$
O	$10^{-2} < x_2 \le 1$	$f_2 = 0$	$-0.06 \le u_2 \le 0.045$
Cl	$0 \le x_2 \le 10^{-2}$	$f_2 = 0$	$-0.06 \le u_2 \le 0.045$

TABLE I

Characteristics of the train–gate dynamics

The gate can be either (completely or partially) *open (O)*, or *closed (Cl)*. A train can be in an *arriving (Ar)*, *crossing (Cr)* or *leaving (L)* situation, depending on its position. In these three situations the whole train–gate system is in an *active (Ac)* situation, while when the train is "far" from the gate the system is *idle (I)* and the train speed $\dot{x}_1$ is constant.

The system is modelled as an iPWA system (5). The automaton that defines the system active-idle behavior has two binary states (I,Ac) and an asynchronous exogenous binary input, *"approaching" (app)*. The transition from the idle state to the active state is the consequence of an app signal, that also sets the train position to the initial value $x_1 = -20$. The transition $Ac \rightarrow I$ is fired when $x_1 \ge 40$.

The iPWA system characteristics are reported in Table I, where x_1 is the train position and x_2 is the gate position. The regions of the continuous state space where the binary state is Ac are obtained by partitioning the axis x_1 in three intervals, representing the train modes Ar, Cr and L, and the axis x_2, relative to the gate, in two intervals representing O and Cl. Where the binary state is I, there is only one partition in the

[2] In general the number of mode switches h is smaller than the total number of events N because of exogenous events.

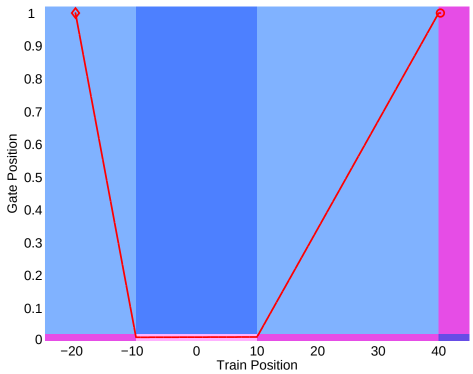

Fig. 3. Train–gate trajectory with minimum time control

continuous state space in which the dynamics are $\dot{x}_1 = 1$ and $\dot{x}_2 = 0$ (constant train speed and gate position blocked).

Consider the following problem: from the initial state (x_{0_c}, x_{0_b}) with $x_{0c} = [-20\ 1]^T$ and $x_{0_b} = Ac$, the train must safely cross the gate in minimum time. The minimum time criterion (19) with target state $x_f = ([40.3\ 1]^T, I)$ was applied with $N = 6$, with the safety constraint imposing that the trajectory never crosses the region (Cr, O).

Figure 3 shows the partitions of the continuous-state space (plus additional regions indicating the thresholds triggering the transition $Ac \to I$) and the optimal trajectory generated by the event-driven approach for the above stated problem. The target state x_f is reached in 93.8 time units, with a trajectory that does not cross the unsafe region. The CPU-time to compute the optimal trajectory is 1.23 seconds.

N	t_s	t_{cpu} (sec.)	t_{viol} (time units)	$t_{viol}^{(sup)}$ (time units)
5	20.00	0.45	3.82	40.00
6	16.66	1.46	21.83	33.32
10	10.00	8.58	9.82	20.00
15	6.66	66.77	6.55	13.32
20	5.00	747.14	4.82	10.00

TABLE II

NUMERICAL RESULTS OF THE TIME-DISCRETIZATION APPROACH APPLIED TO THE TRAIN–GATE SYSTEM.

Table II reports the results of a time-discretization approach on the same problem. The time horizon is 100 time units and the minimum time criterion is applied by using state error cost. In Table II N is the number of time steps, t_s is the sampling period, t_{cpu} is the computation time. Note that by the approach of this paper the constraints are satisfied along the whole trajectory, while in the time-discretization approach they can be enforced only at the sampling instants. Thus, t_{viol} is the period in which the discrete-time solution violates the safety constraint, and $t_{viol}^{(sup)} = 2\,t_s$ is its upper bound.

VI. Conclusions

In this paper we have proposed an event-driven approach for optimally controlling a class of hybrid systems whose continuous-state dynamics are integral piecewise affine. The advantages are the reduction of the computations required for solving optimal control problems and and the avoidance of mode-mismatch errors. It may seem too reductive to consider only iPWA dynamics. However such dynamics have been widely studied for many purposes and a rich model library exists. As an example piecewise integral dynamics have been used in the context of linear hybrid automata (LHA) [7], for verification purposes. Moreover, complex dynamics can be approximated by piecewise affine integral ones [8]. In a future work we will investigate the relation between LHA and icHA in deeper details, as well as the application of our techniques to MIP-based verification.

References

[1] P. J. Antsaklis, “A brief introduction to the theory and applications of hybrid systems,” *Proc. IEEE, Special Issue on Hybrid Systems: Theory and Applications*, vol. 88, no. 7, pp. 879–886, July 2000.

[2] X. Xu and P. J. Antsaklis, “Results and perspectives on computational methods for optimal control of switched systems,” in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, no. 2623. Springer-Verlag, 2003, pp. 540–555.

[3] C. G. Cassandras, D. L. Pepyne, and Y. Wardi, “Optimal control of a class of hybrid systems,” *IEEE Trans. Automatic Control*, vol. 46, no. 3, pp. 3981–415, 2001.

[4] M. S. Shaikh and P. E. Caines, “On the optimal control of hybrid systems: Optimization of trajectories, switching times, and location schedules,” in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, no. 2623. Springer-Verlag, 2003, pp. 466–481.

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

[6] 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.

[7] T. A. Henzinger, “The theory of hybrid automata,” in *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, New Brunswick, New Jersey, 1996, pp. 278–292.

[8] T. A. Henzinger, P. H. 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.

[9] J. Júlvez, A. Bemporad, L. Recalde, and M. Silva, “Event-driven optimal control of continuous Petri nets,” in *Proc. 43th IEEE Conf. on Decision and Control*, Paradise Island, Bahamas, 2004, pp. 69–74.

[10] H. Alla and R. David, “Continuous and Hybrid Petri nets,” *Journal of Circuits, Systems, and Computers*, vol. 8, no. 1, pp. 159–188, 1998.

[11] C. Seatzu, D. Corona, A. Giua, and A. Bemporad, “Optimal control of continuous-time switched affine systems,” *IEEE Trans. Automatic Control*, 2003, accepted for publication as a regular paper.

[12] X. Xu and P. J. Antsaklis, “A linear programming approach to time optimal control of integrator switched systems with state constraints,” in *Proc. 41th IEEE Conf. on Decision and Control*, 2002, pp. 2168–2173.

[13] F. D. 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.

[14] G. Ferrari-Trecate, F. A. Cuzzola, and M. Morari, “Lagrange stability and performance analysis of discrete-time piecewise affine systems with logic states,” *Int. J. Control*, vol. 76, no. 16, pp. 1585–1598, 2003.

[15] 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.

[16] ILOG, Inc., *CPLEX 9.0 User Manual*, Gentilly Cedex, France, 2004.

[17] J. Zhang, K. H. Johansson, J. Lygeros, and S. Sastry, “Dynamical systems revisited: hybrid systems with zeno execution,” in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2000, vol. 1790, pp. 451–464.