Automatica 45 (2009) 1243–1251

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Event-driven optimization-based control of hybrid systems with integral continuous-time dynamics[☆]

S. Di Cairano[a,c], A. Bemporad[a,*], J. Júlvez[b]

[a] Department of Information Engineering, University of Siena, Via Roma 56, Siena, Italy
[b] Department of Software, Technical University of Catalonia, Jordi Girona 1-3, Barcelona, Spain
[c] Powertrain Control Department, Ford Motor Company Research and Advanced Engineering, Dearborn, MI, USA

ARTICLE INFO

Article history:
Received 27 November 2007
Received in revised form
27 June 2008
Accepted 17 December 2008
Available online 13 March 2009

Keywords:
Hybrid systems
Discrete-event systems
Optimal control
Model predictive control
Mixed-integer programming

ABSTRACT

In this paper we introduce a class of continuous-time hybrid dynamical systems called *integral continuous-time hybrid automata* (icHA) for which we propose an event-driven optimization-based control strategy. Events include both external actions applied to the system and changes of continuous dynamics (mode switches). The icHA formalism subsumes a number of hybrid dynamical systems with practical interest, e.g., linear hybrid automata. Different cost functions, including minimum-time and minimum-effort criteria, and constraints are examined in the event-driven optimal control formulation. This is translated into a finite-dimensional mixed-integer optimization problem, in which the event instants and the corresponding values of the control input are the optimization variables. As a consequence, the proposed approach has the advantage of automatically adjusting the attention of the controller to the frequency of event occurrence in the hybrid process. A receding horizon control scheme exploiting the event-based optimal control formulation is proposed as a feedback control strategy and proved to ensure either finite-time or asymptotic convergence of the closed-loop.

1. Introduction

Hybrid systems are capable of modelling complex processes characterized by the coexistence and interaction of discrete and continuous dynamics. The hybrid dynamics are usually described either in discrete-time through difference equations, or in continuous-time through differential equations. The trajectory of a continuous-time hybrid system can be represented as a sequence of continuous evolutions interleaved by discrete events (Lygeros, Johansson, Simic, Zhang, Sastry, 2003), which cause changes in the set of differential equations defining the continuous flow, also called the "modes" of the hybrid system. Such a behavior is exhibited, for instance, by piecewise affine systems, where the coefficients of the linear differential or difference equations defining the continuous dynamics depend on linear inequality conditions over the continuous state and input vectors.

As mode switches introduce discontinuities in the vector fields defining the system dynamics, this may lead to weaker solution concepts for the differential equations, such as the Filippov or Utkin solutions (Van Beek,Pogromsky, Nijmeijer, & Rhooda, 2004), and also to pathological effects, such as Zeno behaviors (Lygeros et al., 2003). The presence of possible resets of the state values after mode switches further complicates the trajectory, as in this case the trajectory itself can become discontinuous. The continuous flow and the instants at which the discrete events occur can be further influenced by exogenous discrete and continuous input signals. From a controller design perspective, when optimal control is applied to continuous-time hybrid systems, the resulting computational problem is hard to solve, since it usually involves nonconvex problems (see (Xu & Antsaklis, 2003) and the references therein).

In the discrete-time setting not only most of the aforementioned subtleties in the characterization of trajectories disappears, but also optimal control problems can be solved very efficiently through mixed-integer programming (MIP) solvers (Bemporad & Morari, 1999). In comparison with the continuous-time case, here the simplification is mainly due to the fact that events (such as mode switches) can only occur at sampling instants. However, mode switches that occur in the intersampling, and hence are not

☆ This paper was not presented at any IFAC meeting. This work was partially supported by the European Community through the HYCON Network of Excellence, contract number FP6-IST-51136, by the Italian Ministry for Education, University and Research (MIUR) under project "Advanced control methodologies for hybrid dynamical systems" (PRIN'2005), and by the Spanish Ministry of Education and Science (Juan de la Cierva fellowship). This paper was recommended for publication in revised form by Associate Editor Bart De Schutter under the direction of Editor Ian R. Petersen.

* Corresponding author. Tel.: + 39 0577 234631; fax: +39 (02) 700 543345.
E-mail addresses: dicairano@ieee.org (S. Di Cairano), bemporad@dii.unisi.it (A. Bemporad), julvez@lsi.upc.edu (J. Júlvez).

0005-1098/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2008.12.011

recognized or delayed, may lead to nonnegligible modelling errors (Di Cairano, 2008, Ch. 5) that we call mode-mismatch. In the discrete-time case, modelling precision can be improved by reducing the sampling period. However, this increases the computational load of the controller. In particular, in a model predictive control (MPC) context (Maciejowski, 2002), the obvious disadvantage is that, for a given time-horizon of prediction, a larger number of control variables is involved in the optimization problem.

To avoid mode-mismatch errors one should change command values exactly when the mode changes, or in other words adopt an *event-driven* control approach. Event-driven approaches have been originally used for control in the framework of discrete-event systems, e.g., finite state machines and Petri nets. Receding horizon control approaches have been successfully applied to some classes of discrete-event systems, max-plus-linear discrete-event systems (De Schutter & van den Boom, 2001), discrete-event systems with real-time constraints (Miao & Cassandras, 2007), and scheduling systems (Cassandras & Mookherjee, 2003).

This paper introduces *integral continuous-time hybrid automata* (icHA), a model paradigm for a special class of continuous-time hybrid dynamical systems for which no a priori information about the timing and order of the events is assumed. It will be shown that the dynamics of an icHA can be expressed with an event-driven formulation, as will be detailed in Section 2. A similar formulation has been used in (Borrelli, Falcone, & del Vecchio, 2007). In Section 3 we formulate different continuous-time optimal control problems of icHA that can be solved via mixed-integer programming techniques despite the event-driven formulation. A similar approach was proposed in (Xu & Antsaklis, 2002), although the system model and the optimization approach are different: in (Xu & Antsaklis, 2002) mode switches are directly controlled and the system dynamics have no other exogenous input signals, that also induces a solution algorithm based on linear programming and enumeration of feasible mode sequences. Finally, the optimal control methods are used in Section 4 to develop an event-driven closed-loop control approach that exploits a model predictive control philosophy, for which conditions for convergence are given.

1.1. Notation

Small letters denote variables or vectors (e.g., v, ξ), superscripts (e.g., v^j) vector components. Subscripts are used for system modes (e.g., A_i, b_i, f_i) and for time instants in a sequence of time instants (t_i). Bold symbols denote finite sequences of variables, e.g., $\mathbf{a} \triangleq \{a(t_i)\}_{i=0}^N$. The sets $\mathbb{R}$ and $\mathbb{Z}$ indicate the set of real numbers and integer numbers, respectively, $\mathbb{Z}_{0+}, \mathbb{R}_{0+}, \mathbb{Z}_{0+}, \mathbb{R}_+$ indicate the sets of nonnegative integer, nonnegative real, positive integer, and positive real numbers, respectively. We use $\mathbb{Z}_{[a,b]}$ where $a, b \in \mathbb{Z}$, $b \geq a$ to indicate the set $\{x \in \mathbb{Z} : a \leq x \leq b\}$. We indicate the origin of a space by $\mathcal{O}$, the interior of a set $\mathcal{X}$ by $\mathrm{Int}(\mathcal{X})$ and its closure by $\mathrm{Cl}(\mathcal{X})$. The binary operator $\setminus$ indicates the difference between sets.

Relational operators (such as $\leq$) over vectors are intended componentwise. For matrices, $Q \geq 0$ indicates positive semidefiniteness, $\|z\|_\infty^Q \triangleq \max_i |(Qz)^i|$, $\|z\|_1^Q \triangleq \sum_i |(Qz)^i|$, and $\|z\|_2^Q \triangleq z^{\mathrm{T}}Qz$ where $Q > 0$. The symbol I indicates the identity matrix of appropriate dimensions, while we use the bold $\mathbf{0}$ and $\mathbf{1}$ to indicate matrices and vectors entirely composed of zeros and ones, respectively. The symbol $\rightarrow$ denotes logical implication (*if*) and the symbol $\leftrightarrow$ logical equivalence (*iff*).

Given a signal $a(\cdot) : \mathbb{R}_{0+} \rightarrow \mathbb{A}$ we indicate its continuous-time trajectory by $a(s)$, $s \in \mathbb{R}_{0+}$, and its event-driven trajectory by $a(j)$, where $j \in \mathbb{Z}_{0+}$, $a(j) = a(t_j)$ and t_j is the time instant at which the jth event occurs. When considering event-driven predictive control, the predicted trajectory of a signal $a(\cdot)$ from state $x(t)$ at time t along event steps $\mathbb{Z}_{[r,s]}$ are indicated by $\mathbf{a}(x(t)) = [a_{x(t)}(r), \ldots, a_{x(t)}(s)]$.

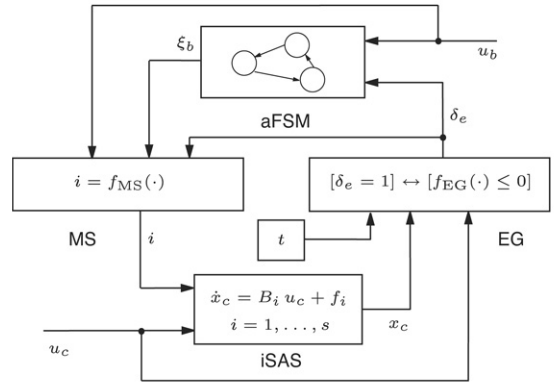


Fig. 1. Integral continuous-time hybrid automaton (icHA).

2. Integral continuous-time hybrid automaton

We consider a continuous-time version of the discrete hybrid automaton (DHA) proposed in (Torrisi & Bemporad, 2004), denoted as *integral continuous(-time) hybrid automaton* (icHA), where discrete-time affine dynamics are replaced by integral continuous-time dynamics. Similarly to the DHA, the icHA consists of the four components reported in Fig. 1: the integral switched affine system (iSAS), the event generator (EG), the mode selector (MS) and the asynchronous finite state machine (aFSM). The iSAS represents a collection of continuous-time integral dynamics for the continuous states

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

where $x_c \in \mathbb{R}^{n_c}$ 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. As will be detailed later, the main reason for restricting the attention on integral dynamics instead of more general linear dynamics is computational. Nonetheless, the class of continuous-state dynamics (1) has been widely exploited for modelling and verification of hybrid systems (Henzinger, 1996; Henzinger, Ho, & Wong-Toi, 1997; Xu & Antsaklis, 2002), as it is powerful enough to model many practical problems. In fact, given a nonlinear (possibly discontinuous) dynamical model $\dot{x}_c = f(x_c(t), u_c(t))$, model (1) can be interpreted as a piecewise zero-order approximation of the state-transition function with respect to the state vector x_c and a first-order approximation with respect to the input vector u_c. Similarly to the piecewise affine (PWA) approximation $\dot{x}_c = A_i x_c + B_i u_c + f_i$, the icHA dynamics can be made arbitrarily close to a given nonlinear dynamics by increasing the number of modes.

The EG defines the *endogenous* binary inputs δ_e according to the linear threshold conditions

$$[\delta_{e_x}^i(t) = 1] \leftrightarrow \left[E_i^x \begin{bmatrix} x_c(t) \\ t \end{bmatrix} \leq F_i^x\right], \quad i \in \mathbb{Z}_{[1,n_e^x]}, \tag{2a}$$

$$[\delta_{e_u}^i(t) = 1] \leftrightarrow \left[E_i^u u_c(t) \leq F_i^u\right], \quad i \in \mathbb{Z}_{[1,n_e^u]}, \tag{2b}$$

and we get $\delta_e \triangleq [\delta_{e_x}^1 \ldots \delta_{e_x}^{n_e^x}\, \delta_{e_u}^1 \ldots \delta_{e_u}^{n_e^u}]^{\mathrm{T}} \in \{0, 1\}^{n_e}$, $n_e \triangleq n_e^x + n_e^u$, as the vector of endogenous binary input variables. The icHA may be also excited by *exogenous* binary input signals $u_b \in \{0, 1\}^{m_b}$ entering the mode selector and the asynchronous finite state machine.

Definition 2.1. $\mathcal{PC}_{(m_c,m_b)}$ is the set of piecewise constant functions $u(t) \triangleq \begin{bmatrix} u_c(t) \\ u_b(t) \end{bmatrix}$, $u : \mathbb{R} \rightarrow \mathbb{R}^{m_c} \times \{0, 1\}^{m_b}$ such that $u(t)$ is constant for all $t \in [t_k, t_{k+1})$, $k \in \mathbb{Z}_{0+}$, where $t_0 < t_1 < \cdots$ is a sequence of time instants. $\square$

Discrete-time samples held constant between sampling instants by a zero-order holder are a special case of piecewise constant functions, corresponding to equally spaced time instants. General piecewise constant functions can approximate arbitrarily well any given piecewise continuous function, provided that the number of intervals can go to infinity and the interval length can go to 0.

Assumption 2.1. In what follows it is assumed that $u : \mathbb{R} \to \mathbb{R}^{m_c} \times \{0,1\}^{m_b} \in \mathcal{PC}_{(m_c,m_b)}$, [1] i.e., the input trajectory is piecewise constant along time. □

Definition 2.2. An *event* occurs whenever an endogenous input δ_e or an exogenous input $\begin{bmatrix} u_c \\ u_b \end{bmatrix}$ changes value. Given the initial time $t_0 \in \mathbb{R}$, the event instants are defined recursively as

$$t_k \triangleq \min_{t>t_{k-1}} \{t : (u_c(t), u_b(t), \delta_e(t)) \neq (u_c(t_{k-1}), u_b(t_{k-1}), \delta_e(t_{k-1}))\}, \tag{3}$$

where clearly $t_j < t_{j+1}$ for all $j \in \mathbb{Z}_{0+}$. We assume that the minimum in (3) always exists. □

The Boolean state $\xi_b \in \{0,1\}^{n_b}$ is defined as

$$\xi_b(t) \triangleq x_b(t_k) \quad \text{for } t_{k-1} \leq t < t_k, \tag{4}$$

$$x_b(t_{k+1}) = f_{\text{aFSM}}(x_b(t_k), u_b(t_k), \delta_e(t_k)), \tag{5}$$

and $f_{\text{aFSM}} : \{0,1\}^{n_b+m_b+n_e} \to \{0,1\}^{n_b}$ is the Boolean function defining the transitions of the asynchronous finite state machine. The Boolean state $\xi_b(t)$ remains constant, $\xi_b(t) = x_b(t_k)$, during the whole interval $t_{k-1} \leq t < t_k$. At the event instant t_k, the Boolean state switches to the new value $f_{\text{aFSM}}(x_b(t_k), \delta_e(t_k), u_b(t_k))$, and remains at that value for $t_k \leq t < t_{k+1}$. While we are assuming that the transitions of the aFSM are instantaneous, delays can be easily modelled by introducing additional events and states. In (5) discrete-state transitions can occur at any time instant, not only at multiples of a given sampling period as for DHA.

The different operating modes of the system represented by the variable $i(t)$ are selected by the mode selector (MS) function through the scalar product

$$i(t) = [1\,2\ldots s] \cdot f_{\text{MS}}(\xi_b(t), u_b(t), \delta_e(t)), \tag{6}$$

where $f_{\text{MS}} : \{0,1\}^{n_b+m_b+n_e} \to \{0,1\}^s$ is a Boolean function satisfying the mutual exclusivity relation $\mathbf{1}^\mathrm{T} \cdot f_{\text{MS}}(\xi_b(t), u_b(t), \delta_e(t)) = 1$, for all $(\xi_b(t), u_b(t), \delta_e(t)) \in \{0,1\}^{n_b+m_b+n_e}$. Note that when u_b and δ_e are constant, the Boolean state and the mode are also constant.

2.1. Event-driven representation of icHA

The icHA model (1)–(6) is next converted to a computationally-oriented event-driven representation.

Let $\mathbf{t} = \{t_j\}_{j=0}^{h-1}$ be a finite sequence of ordered time instants $t_0 < t_1 < \cdots < t_{h-1}$. Let $u_c : \mathbb{R} \to \mathbb{R}^{m_c}$ be a piecewise constant function with breakpoints at the instants in $\mathbf{t}$ and $\mathbf{u} = \{u_c(t_k)\}_{k=0}^{h-1}$ the corresponding sequence of levels. For the affine dynamics $\dot{x}_c(t) = Ax_c(t) + Bu(t) + f$, the state value at t_j, $j \in \mathbb{Z}_{[0,h-1]}$, is a nonlinear function of $\mathbf{t}$ and $\mathbf{u}$. However in the case of integral dynamics ($A = \mathbf{0}$), $x_c(t_j) = x_c(t_0) + \sum_{k=0}^{j-1} (B(t_{k+1} - t_k)u_c(t_k) + (t_{k+1} - t_k)f(t_k))$. Consider now a system with switched integral dynamics (1), and let the set of mode-switch instants $\{t : i(t-\varepsilon) \neq i(t+\varepsilon), \varepsilon \to 0\} \subseteq \mathbf{t}$. Since in any time interval $[t_k, t_{k+1})$ the mode is constant, we obtain that the system dynamics can be rewritten as the first-order linear difference equations

$$x_c(k+1) = x_c(k) + B_{i(k)}v_c(k) + f_{i(k)}q(k) \tag{7a}$$

$$t(k+1) = t(k) + q(k) \tag{7b}$$

where k is the event counter, $x_c(k) = x_c(t_k)$, $t(k) = t_k$, $i(k) = i(t_k)$, $q(k)$ is the time interval between event instants t_k and t_{k+1}, $v_c(k) = q(k)u_c(k)$ is the integral over time period $q(k)$ of the input $u_c(k) = u_c(t_k)$. Note that in (7b) time t is treated as an additional state variable. The controlled variables are the input integral $v_c(k)$ and the input level duration $q(k)$, from which the actual input $u_c(k) = \frac{v_c(k)}{q(k)}$ to be applied to the continuous-time system is immediately recovered.

The event generator becomes

$$[\delta_{e_x}^i(k) = 1] \leftrightarrow \left[E_i^x \begin{bmatrix} x_c(k) \\ t(k) \end{bmatrix} \leq F_i^x\right], \quad i \in \mathbb{Z}_{[1,n_e^x]} \tag{8a}$$

$$[\delta_{e_u}^i(k) = 1] \leftrightarrow \left[E_i^u v_c(k) \leq F_i^u q(k)\right], \quad i \in \mathbb{Z}_{[1,n_e^u]}, \tag{8b}$$

where $\delta_e(k) = \delta_e(t_k)$, and $\delta_e(t) = \delta_e(t_k)$, for all $t \in [t_k, t_{k+1})$ by the definition of t_k in (3). The dependence on time becomes a dependence on a state variable, because of (7b), and (8b) is obtained from (2b) by multiplying by $q(k)$ both sides of the right clause. The mode selector equation becomes

$$i(k) = [1\,2\ldots s] \cdot \tilde{f}_{\text{MS}}(x_b(k), u_b(k), \delta_e(k)), \tag{9}$$

where $i(t) = i(k)$, for all $t \in [t_k, t_{k+1})$, as a consequence of the event definition, and $\tilde{f}_{\text{MS}}(x_b(k), u_b(k), \delta_e(k)) = f_{\text{MS}}(f_{\text{aFSM}}(x_b(k), u_b(k), \delta_e(k)), u_b(k), \delta_e(k))$ because of (6) and of the definition of $\xi_b(t)$. Finally, Eq. (5) is already defined with respect to the event instants.

Eq. (5), (7), (8), (9) define the behavior of the icHA in an event-driven representation. However, to account for (3), the following condition must be ensured:

$$\left[(\delta_e(k), u_c(k), u_b(k)) = (\bar{\delta}_e, \bar{u}_c, \bar{u}_b)\right] \to \left[(\delta_e(t), u_c(t), u_b(t)) = (\bar{\delta}_e, \bar{u}_c, \bar{u}_b), \forall t \in [t_k, t_{k+1})\right]. \tag{10}$$

We consider two different cases: (i) the value of u_c or u_b changes, so that an event is externally forced, for instance by a controller, (ii) the value of δ_e changes, i.e., an endogenous event occurs. Since we will focus on an event-driven control design, in which exogenous events are generated on purpose by the controller, we can assume that $u_c(t)$, $u_b(t)$ are constant between event instants and simply restrict condition (10) to

$$\left[\delta_e(k) = \bar{\delta}_e\right] \to \left[\delta_e(t) = \bar{\delta}_e,\ \forall t \in [t_k, t_{k+1})\right], \tag{11}$$

i.e., the current mode is kept until the next endogenous event. Moreover, as $\delta_{e_u}^i$ variables in (2b) can change only when the input changes, they can be dealt with as for externally forced events, namely by appropriate selection of the control inputs. Thus, it is indeed sufficient to enforce (11) for the variables in (2a).

Let the mapping $\text{cod}(\cdot) : \{0,1\}^{n_e^x} \to \mathbb{Z}_{0+}$ associate a nonnegative integer number d to each allowed value of vector $\delta_e^x = [\delta_{e_x}^1 \ldots \delta_{e_x}^{n_e^x}]^\mathrm{T}$ defined in (2a). For example d may be the integer whose binary encoding is δ_{e_x}. Define the matrix $\bar{E}^x(d)$ and the vector $\bar{F}^x(d)$ by collecting the rows in the inequalities of the EG (8) which are satisfied for e^x such that $\text{cod}(\delta_{e_x}) = d$. Define $\hat{E}^x(d)$, $\hat{F}^x(d)$ by collecting as rows the inequalities of the EG (8), which are not satisfied for δ_{e_x} such that $\text{cod}(\delta_{e_x}) = d$. Hence (11) is equivalently replaced by

$$\left[\text{cod}(\delta_{e_x}(k)) = d\right] \to \left[\bar{E}^x(d)\begin{bmatrix} x \\ t \end{bmatrix} \leq \bar{F}^x(d), \hat{E}^x(d)\begin{bmatrix} x \\ t \end{bmatrix} > \hat{F}^x(d)\right], \quad \forall t \in [t_k, t_{k+1}). \tag{12}$$

[1] For the simplicity of notation the subscript (m_c, m_b) will be dropped in the rest of the paper when the values of m_c and m_b are clear from the context.

As an example, consider two thresholds $[\delta^1_{e_x} = 1] \leftrightarrow [x \le 0]$, $[\delta^2_{e_x} = 1] \leftrightarrow [x \le 1]$. The matrices associated to $\delta^x_e = [0\ 1]^{\mathrm{T}}$, where $\mathrm{cod}(\delta^x_e) = 1$, are $\bar{E}^x(1) = 1$, $\bar{F}^x(1) = 1$, collecting the second threshold condition (satisfied), and $\hat{E}^x(1) = 1$, $\hat{F}^x(1) = 0$.

Proposition 2.1. *Let* $\mathrm{cod}(\tilde{\delta}_{e_x}) = d$, *and let* $\bar{E}^x(d)$, $\hat{E}^x(d)$ *and* $\bar{F}^x(d)$, $\hat{F}^x(d)$ *be the associated matrices and vectors, respectively, obtained by collecting the rows in (8) which are either satisfied* ($\bar{E}^x$, $\bar{F}^x$) *or not satisfied* ($\hat{E}^x$, $\hat{F}^x$) *when* δ_{e_x} *is such that* $\mathrm{cod}(\delta_{e_x}) = d$. *In the case of integral dynamics, if* $u_c(t)$ *is constant for* $t \in [t_k, t_{k+1})$, *(11) is guaranteed by the mixed-logical constraint*

$$\left[\mathrm{cod}(\delta^x_e(k)) = d\right] \to \left[\begin{bmatrix}\bar{E}^x(d)\\ -\hat{E}^x(d)\end{bmatrix}\begin{bmatrix}x(k+1)\\ t(k+1)\end{bmatrix} \le \begin{bmatrix}\bar{F}^x(d)\\ -\hat{F}^x(d)\end{bmatrix} + \varepsilon\mathbf{1}\right], \tag{13}$$

for $\varepsilon > 0$, $\varepsilon \to 0^+$.

Proof. Because of the integral dynamics, the state trajectory for $t \in [t_k, t_{k+1})$ is the line $x(t) = x(k) + \gamma(t - t(k))$, where $\gamma = \frac{x(k+1)-x(k)}{t(k+1)-t(k)}$. Condition (12) and condition (13) define two polyhedra, $\mathcal{P}$ and $\mathcal{P}_\varepsilon$, respectively. Since $\varepsilon > 0$, $\mathcal{P}_\varepsilon \supset \mathcal{P}$ and if $\varepsilon \to 0$, $\mathcal{P}_\varepsilon \to \mathcal{P}$. Condition (13) ensures that $x(k+1) \in \mathcal{P}_\varepsilon$ and (12) ensures that $x(k) \in \mathcal{P} \subset \mathcal{P}_\varepsilon$, hence for all $t \in [t_k, t_{k+1}]$, $x(t) \in \mathcal{P}_\varepsilon$. Thus, by linearity of the trajectory there exists $\varrho(\varepsilon)$ such that $x(t) \in \mathcal{P}$, for all $t \in [t_k, t_{k+1} - \varrho(\varepsilon)]$. Since for $\varepsilon \to 0$, $\mathcal{P}_\varepsilon \to \mathcal{P}$, then for $\varepsilon \to 0$ also $\varrho(\varepsilon) \to 0$. □

The effect of adding ε in Eq. (13) is to expand the polyhedron $\mathcal{P}$. This is necessary for allowing $\delta^x_e(k+1) \ne \delta^x_e(k)$, otherwise the system would be constrained to remain always in the same mode. The value of $\varrho(\varepsilon)$ is the time the trajectory spends in $\mathcal{P}_\varepsilon \setminus \mathcal{P}$. When $\mathcal{P}_\varepsilon \to \mathcal{P}$, $\varrho(\varepsilon) \to 0$.

Note that, within a given mode $i(k)$, $x(k+1)$ is an affine function of $x(k)$, $q(k)$, and $v(k) \triangleq \begin{bmatrix}v_c(k)\\ v_b(k)\end{bmatrix}$, where $v_b(k) = u_b(k)$, so that (13) can be reformulated as a set of mixed-integer inequalities on $x(k)$, $q(k)$, $v(k)$, $\delta_e(k)$ (Williams, 1993). Indeed, Eqs. (5), (7), (8), (9), (13) represent a DHA that can be converted into the *event-driven MLD* (eMLD) system

$$x(k+1) = Ax(k) + B_1 w(k) + B_2\delta_e(k) + B_3 z(k) + B_5, \tag{14a}$$

$$t(k+1) = t(k) + q(k), \tag{14b}$$

$$E_2\delta_e(k) + E_3 z(k) \le E_1 w(k) + E_4 x(k) + E_5 + E_6 t(k), \tag{14c}$$

where $w(k) \triangleq \begin{bmatrix}v(k)\\ q(k)\end{bmatrix}$, for instance using the tool HYSDEL (Torrisi & Bemporad, 2004). Differently from the standard discrete-time MLD system (Bemporad & Morari, 1999), in the eMLD (14) k is an event counter, while time t is an additional state variable.

Remark 2.1. Discontinuities of the continuous state trajectory in the form of state resets $x_c(t_{k+1}) = \Phi_i x_c(t_k) + \mu_i$ can be included as follows. To model resets one must add *reset modes* $i \in \{s+1, \ldots, s_r\}$, modify (7a) into $x_c(k+1) = (\Phi_i x_c(k) + \mu_i) + B_i v(k) + f_i q(k)$, and (7b) into $t(k+1) = t(k) + \Gamma_i q(k)$, where in modes $i = \{1, \ldots, s\}$, $\Phi_i = I$, $\mu_i = \mathbf{0}$, and $\Gamma_i = 1$, while in reset modes $i = \{s+1, \ldots, s_r\} B_i = \mathbf{0}$, $f_i = \mathbf{0}$, and $\Gamma_i = \mathbf{0}$.

When a reset occurs Eq. (11) does not apply, as $t_{k+1} = t_k$. In this case, assuming that two consecutive resets cannot occur, the continuous-time trajectory $x_c(t)$ may be discontinuous in t_k, and is defined as follows. Let j, be the mode immediately before the reset mode i. Then, $\lim_{t \to t_k^-} x_c(t) = x_c(k-1) + B_j v(k-1) + f_j q(k-1)$, $x_c(t_k) = \Phi_i(x_c(k-1) + B_j v(k-1) + f_j q(k-1)) + \mu_i$, and $\lim_{t \to t_k^+} x_c(t) = x_c(t_k)$. In other words, we define the state trajectory $x_c(t)$ as right continuous.

In accordance with the above definition of resets, in eMLD models these are instantaneous, contrary to resets in discrete-time MLD models that are constrained to last one sampling interval (Torrisi & Bemporad, 2004). An equivalent event-driven PWA system (ePWA) of the eMLD (14) can be obtained by using the algorithm in (Bemporad, 2004). See (Di Cairano, 2008) for further details.

2.2. Modelling capabilities

Even though the piecewise integral dynamics of the continuous state of icHA has some limitations, the icHA cover several popular model classes.

Linear hybrid automata (LHA), known to be useful for both control (Wong-Toi, 1997) and verification purposes (Henzinger et al., 1997), can be modelled by icHA, as intuitively shown by the example in Section 3.3. A formal proof of the capabilities of icHA to model linear hybrid automata can be obtained by resorting to an equivalent piecewise affine representation of the icHA system (Di Cairano, 2008) and by exploiting the results in (Di Cairano & Bemporad, 2006) on the relations between linear hybrid automata and piecewise affine systems. In (Bemporad, Di Cairano, & Júlvez, 2006) an example of formal verification of a LHA performed through its equivalent icHA formulation was given.

Another class of systems that can be modelled as icHA is the class of continuous Petri nets (CPN) (Silva & Recalde, 2002), that relaxes classical Petri nets allowing continuous values for transition firings, to alleviate combinatorial state explosion. The dynamics of a timed continuous Petri net under finite server semantics is piecewise integral, hence the class of continuous Petri nets is contained in the class of icHA. The application to continuous Petri net control of the techniques proposed here has been presented in (Julvez, Bemporad, Recalde, & Silva, 2004).

Finally as mentioned in Section 2, piecewise integral dynamics can approximate nonlinear dynamics arbitrarily well, possibly at the price of increasing the number of system modes, as in the case of piecewise affine systems.

3. Event-driven optimal control

Consider the *event-driven optimal control problem* for the icHA (1)–(6)

$$\min_{\mathbf{x},\mathbf{t},\mathbf{q},\mathbf{v}} J(\mathbf{x}, \mathbf{t}, \mathbf{q}, \mathbf{v}) \tag{15a}$$

$$\text{s.t.}\quad \text{icHA dynamics (1)–(6)} \tag{15b}$$

$$g(x(k), t(k), q(k), v(k)) \le \mathbf{0},\ k \in \mathbb{Z}_{[0,N-1]} \tag{15c}$$

$$g_N(x(N), t(N)) \le \mathbf{0} \tag{15d}$$

$$x(0) = x_0, \quad t(0) = t_0, \tag{15e}$$

where $J(\cdot)$ is a convex function of $(\mathbf{x}, \mathbf{t}, \mathbf{q}, \mathbf{v})$, $\mathbf{t} = \{t(k)\}_{k=0}^N$ are the time instants at which events occur, $\mathbf{x} = \{x(k)\}_{k=0}^N$ are the corresponding state values, $\mathbf{q} = \{q(k)\}_{k=0}^{N-1}$ are the durations of the time intervals between two consecutive events, $q(k) = t(k+1) - t(k)$, $\mathbf{v} = \{v(k)\}_{k=0}^{N-1}$ are the input integrals computed on the corresponding intervals $\{[t(k), t(k+1))\}_{k=0}^{N-1}$, x_0 is a given initial state, and t_0 a given initial time. Constraints (15c) and (15d) represent possible additional constraints in the optimal control problem and N is the event-based optimal control horizon, i.e., the number of events considered in the optimal control formulation.

By formulating the icHA dynamics (15b) as the equivalent eMLD model (14), problem (15) can be solved by mixed-integer programming (MIP) for different objective functions (15a) and constraints (15c) and (15d) as detailed in Section 3.1. The cost function and the constraints determine the type of problem to be solved. As described below, all such problems are formulated in a way that results in mixed-integer linear or quadratic programming (MILP, MIQP) problems, for which efficient and reliable solvers exist.

3.1. Cost function

Similar to the discrete-time case, the cost function in (15a) can be defined as

$$J(\mathbf{x}, \mathbf{t}, \mathbf{q}, \mathbf{v}) \triangleq F(x(N), t(N)) + \sum_{k=0}^{N-1} L(x(k), t(k), q(k), v(k)). \quad (16)$$

For instance, one can set $L(x, t, v, q) \triangleq \|x - \hat{x}\|_p^{Q_1} + \|t - \hat{t}\|_p^{Q_2} + \|v - \hat{v}\|_p^{R_1} + \|q - \hat{q}\|_p^{R_2}$, $F(x, t) \triangleq \|x - \hat{x}\|_p^{Q_N} + \|t - \hat{t}\|_p^{Q_T}$, $p \in \{1, 2, \infty\}$, where "ˆ" denotes a given reference value for the corresponding vector. From the general formulation (15) we can derive specific forms of the cost function (15a) and the associated constraints (15c) and (15d).

It may be required that the system state reaches a desired target state $\hat{x}$ after N events, $x(N) = \hat{x}$, or, in a softened form, that it gets very close to it by setting $F(x(N), t(N)) = \rho \|x(N) - \hat{x}\|_\infty$, where $\rho \in \mathbb{R}_+$ is a large weight. In alternative, one can consider a convex desired target set $\mathcal{X}_N$ and impose the constraint $x(N) \in \mathcal{X}_N$. A target time $\hat{t}$ or set $\mathcal{T}_N$ can be formulated similarly.

The minimum-time criterion looks for the sequence $(\mathbf{x}, \mathbf{t}, \mathbf{q}, \mathbf{v})$ that minimizes the time needed to bring the system from x_0 to the final state $\hat{x}$. This can be obtained by setting $L(x(k), t(k), q(k), v(k)) = |q(k)| = q(k)$, $F(x(N), t(N)) = 0$ in (16).

The minimum-effort criterion looks for minimizing the intensity of the command input $u(t)$. If the ℓ_1-norm of the input function is used, we obtain $J(\mathbf{x}, \mathbf{t}, \mathbf{q}, \mathbf{v}) = \int_0^{t_N} \|u(t)\| \mathrm{d}t = \sum_{k=0}^{N-1} \int_{t(k)}^{t(k+1)} \|u(t)\|_1 \mathrm{d}t$. Since u is constant in each period $[t(k), t(k+1))$, it is enough to set $L(x(k), t(k), q(k), v(k)) = \|v(k)\|_1$, $F(x(N), t(N)) = 0$ in (16).

The minimum-displacement criterion looks for the trajectory that minimizes the largest deviation from a desired continuous state trajectory $\hat{x}_c(\cdot)$, that we assume piecewise linear and continuous (a special case is $\hat{x}_c(\cdot) \equiv \hat{x}_c$):

$$J(\mathbf{x}, \mathbf{t}, \mathbf{q}, \mathbf{v}) = \max_{t \in [t(0), t(N)]} \|x_c(t) - \hat{x}_c(t)\|_\infty. \quad (17)$$

Proposition 3.1. *Let $x_c(t)$, for all $t \in [t_0, t_N]$, be the trajectory of continuous states of an icHA system with no resets, $t_0 < t_1 < \cdots < t_N$ be the event instants, assume that $\hat{x}_c(t)$ is linear over each $[t_i, t_{i+1})$, $i \in \mathbb{Z}_{[0,N-1]}$ and continuous over $[t_0, t_N]$. Then $\max_{t \in [t_0, t_N]} \|x_c(t) - \hat{x}_c(t)\|_\infty = \max_{k=0,\ldots,t_N} \{\|x_c(t(k)) - \hat{x}_c(t(k))\|_\infty\}$.*

Proof. In the absence of resets, state trajectories of icHA are continuous, so $\|x_c(\cdot) - \hat{x}_c(\cdot)\|_\infty$ is continuous, being the composition of continuous functions $(\|\cdot\|_\infty, x_c(\cdot), \hat{x}_c(\cdot))$, and therefore the maximum over $[t_0, t_N]$ is well defined. Moreover, function $\|x_c(\cdot) - \hat{x}_c(\cdot)\|_\infty$ is a convex function of time t on $[t(k), t(k+1)]$, being the composition of a convex function (the infinity norm) with linear functions (the state trajectory of the icHA and $\hat{x}_c$ between two consecutive switches), and thus it attains its maximum either at $t(k)$ or at $t(k+1)$. Hence $\max_{t \in [t(0), t(N)]} \|x_c(t) - \hat{x}_c(t)\|_\infty = \max_{0 \le k \le N-1} \{\max_{t \in [t(k), t(k+1)]} \|x_c(t) - \hat{x}_c(t)\|_\infty\} = \max_{0 \le k \le N-1} \{\max\{\|x_c(t(k)) - \hat{x}_c(t(k))\|_\infty, \|x_c(t(k+1)) - \hat{x}_c(t(k+1))\|_\infty\}\} = \max_{0 \le k \le N} \{\|x_c(t(k)) - \hat{x}_c(t(k))\|_\infty\}$. □

While (17) is not in the form (16), its equivalent form $\max_{0 \le k \le N} \{\|x_c(t(k)) - \hat{x}_c(t(k))\|_\infty\}$ still leads to a mixed-integer linear optimal control problem.

3.2. Operating constraints

The constraints in (15c) and (15d) can involve quite general mixed-integer linear constraints on states, event instants, and inputs. By using convexity arguments and continuity of $x_c(\cdot)$ similar to the ones used in the proof of Proposition 3.1, state constraints $\underline{h} \le Hx(t) \le \overline{h}$, $t \in [t(0), t(N)]$ can simply be enforced through constraints on the state at event instants only

$$\underline{h} \le Hx(k) \le \overline{h}, \quad k \in \mathbb{Z}_{[0,N]}. \quad (18)$$

Similarly, input bounds $\underline{u} \le u_c(t) \le \overline{u}$, $t \in [t(0), t(N))$ can be rewritten immediately as the linear constraints

$$\underline{u}q(k) \le v(k) \le \overline{u}q(k), \quad k \in \mathbb{Z}_{[0,N-1]}. \quad (19)$$

Different input bounds for different modes can be enforced by the logical constraint $[i(k) = \bar{\imath}] \to [\underline{u}_{\bar{\imath}}q(k) \le v(k) \le \overline{u}_{\bar{\imath}}q(k)]$, where $\underline{u}_{\bar{\imath}}$ and $\overline{u}_{\bar{\imath}}$ are upper and lower bounds of the input while in mode $\bar{\imath}$. Note that, differently from the standard discrete-time optimal control problem where constraint satisfaction is guaranteed only *pointwise in time*, in the event-driven approach state constraints are enforced *continuously on time*.

Additional operating constraints may be imposed on time intervals between two consecutive events

$$\underline{q} \le q(k) \le \overline{q}, \quad k \in \mathbb{Z}_{[1,N-1]}. \quad (20)$$

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 until the next event. 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 control-induced Zeno behaviors (Lygeros et al., 2003). In more detail, if constraint (20) is enforced, no solution that chatters with period smaller than $\underline{q}$ is generated by the control strategy. Constraint softening may be used to avoid infeasibility when chattering cannot be avoided, thus resulting in the largest possible chattering period. Furthermore, (20) ensures that the time-length of the optimization problem is at least $N\underline{q}$. Finally, the lower bound $\underline{q}$ can be used to account for the computation time of the optimal control problem (15). The minimum dwell time may be used there to guarantee that there is enough time to update the control input before the next event occurs.

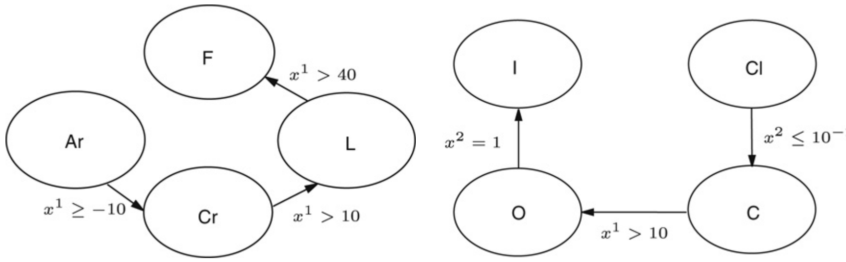
Constraints on event instants $\underline{t}_k \le t(k) \le \overline{t}_k$, $k \in \mathbb{Z}_{[1,N]}$ can be enforced, since time is a state variable on the optimization problem (15). These may be combined with state constraints to enforce that at a time instant during a given interval, the state value is within a given range.

3.3. Numerical example

Consider the well-known "train–gate" benchmark for hybrid systems (Henzinger et al., 1997), commonly used for verification purposes, slightly modified and proposed here as a control problem.

The system is composed of a train and a gate. The control objective is to let the train cross the gate as fast as possible in a safe condition, i.e., the gate must be closed when the train crosses it. The well-known linear hybrid automaton describing the system model is shown in Fig. 2. The discrete state of the system is composed of the discrete state of the train and of the gate, whose dynamics are described by the automata in Fig. 2. The train can be in an *arriving (Ar)*, *crossing (Cr)*, *leaving (L)*, or *far (F)* situation, depending on its position, the gate can be *open (O)*, *closing (Cl)*, *closed (C)* or *idle (I)*.

The continuous states of the system are the train position x^1 and the gate opening percentage x^2, where $x^2 = 0$ means completely



(a) Train Automaton. (b) Gate Automaton.

Fig. 2. Automata that describe the Train–Gate system.

Table 1
Numerical results of the time-discretization approach (e.d. = event-driven (5), t.u. = time units).

N	T_s (t.u.)	T_{cpu} (s)	T_{viol}(t.u.)	$T_{viol}^{(sup)}$(t.u.)
5	20.00	0.29	3.82	40.00
6	16.66	0.86	21.83	33.33
10	10.00	10.93	9.82	20.00
15	6.66	640.21	6.55	13.33
e.d.	–	0.26	0	0

closed while $x^2 = 1$ means completely open. Instead of by the differential inclusions (Henzinger et al., 1997) used for verification, we define the dynamics by $\dot{x} = u(t) + f_i$, where $f_i(t) = \begin{bmatrix} f^1(t) \\ 0 \end{bmatrix}$, $u(t) = \begin{bmatrix} u^1(t) \\ u^2(t) \end{bmatrix}$ and $\underline{u}_i^j \leq u^j \leq \overline{u}_i^j$, $j = 1, 2$, and i is the system mode. The system modes are the cartesian product of the states of the train and the gate automata, hence 16 in total. The numerical values can be found in (Di Cairano, 2008, Sec. 5.2.3).

Consider the following problem: from the initial state $\begin{bmatrix} x_{0_c} \\ x_{0_b} \end{bmatrix}$ with $x_{0_c} = \begin{bmatrix} -20 \\ 1 \end{bmatrix}$ and $x_{0_b} = (Ar, Cl)$, the train must safely cross the gate in minimum time. For control purposes we have modelled the system as an icHA (1)–(6). The minimum time criterion $L(x(k), t(k), q(k), v(k)) = |q(k)| = q(k)$, $F(x(N), t(N)) = 0$ with target state was $\hat{x} = \begin{bmatrix} 40.3 \\ 1 \\ (F, I) \end{bmatrix}$ which was applied with $N = 5$, under the safety constraint imposing that the discrete state (Cr, O) is never reached. The target state $\hat{x}$ is reached in 93.8 time units, with a trajectory that does not cross the unsafe region $(\{x \in \mathbb{R}^2 : 10 \leq x^1 \leq 10, 10^{-2} \leq x^2 \leq 1\})$. The CPU-time to compute the optimal trajectory is 0.26 s.[2]

Table 1 reports the results of the event-driven control compared to a discrete-time solution of the same problem. The time horizon of the discrete-time approach is 100 time units. The minimum-time criterion, which cannot be perfectly enforced in a discrete-time approach, is replaced by minimizing the tracking error between the current and target states. In Table 1, N is the number of time steps, T_s is the sampling period, T_{cpu} is the computation time. In Table 1, T_{viol} indicates the time period during which the current discrete-time solution violates the safety constraint. The violations occur in the intersampling period. $T_{viol}^{(sup)} = 2\,T_s$ is the upper bound on T_{viol}, due to the fact that the safety constraints can be violated up to one sampling period when entering the gate and up to one sampling period when leaving from it. By decreasing T_s, also $T_{viol}^{(sup)}$ decreases, ensuring a better system safety. However, the number of required control steps increases for a fixed time-horizon, and hence the complexity of the optimization problem increases as shown in Table 1.

2 All the simulations presented in this paper have been performed on a Pentium IV-M 2.0 GHz with 1 GByte RAM running Cplex 9.0 and Matlab 6.5.

4. Event-driven model predictive control

The event-driven optimal control approach of Section 3 is an open-loop one. In this section we introduce an event-driven closed-loop strategy based on Model Predictive Control (MPC) concepts (Maciejowski, 2002).

Given an icHA and its eMLD translation obtained as described in Section 2.1, consider again the optimal control problem (15). The event-driven Model Predictive Control (eMPC) strategy is defined as follows:

1. Let N be the event horizon; at a generic time t set $t_0 = t$, $x_0 = x(t)$ in (15).
2. Solve (15). Let $\mathbf{v}^*(x(t)) \triangleq [v_{x(t)}^*(0), \ldots, v_{x(t)}^*(N-1)]$ be the sequence of optimal input integral values, $\mathbf{q}^*(x(t)) \triangleq [q_{x(t)}^*(0), \ldots, q_{x(t)}^*(N-1)]$ be the sequence of input action durations, $\mathbf{x}^*(x(t)) \triangleq [x_{x(t)}^*(1), \ldots, x_{x(t)}^*(N)]$ be the predicted state values at event instants, and $\mathbf{t}^*(x(t)) \triangleq [t_{x(t)}^*(1), \ldots, t_{x(t)}^*(N)]$ be the corresponding time instants at which the events occur, computed from initial state $x(t)$ and initial time t.
3. Compute the input value $u_c(t) = \frac{v_{x(t),c}^*(0)}{q_{x(t)}^*(0)}$, and apply $u(t) = \begin{bmatrix} u_c(t) \\ v_{x(t),b}^*(0) \end{bmatrix}$ during the time interval $[t, t + q_{x(t)}^*(0)]$ to the icHA.[3]
4. Set $t_0 = t + q_{x(t)}^*(0)$, $x_0 = x(t + q_{x(t)}^*(0))$ in (15) and go to Step 2.

The actual state $x(t + q_{x(t)}^*(0))$ at the end of each control action may be different from the predicted one $x_{x(t)}^*(1)$ because of external disturbances and modelling errors. In fact, also the time instant at which the optimization problem is repeated may be different from the scheduled instant $t + q_{x(t)}^*(0)$. By the closed-loop nature of the eMPC approach, the current state (and time) are measured or estimated again and a new updated optimal input sequence is computed.

4.1. An example of eMPC

Consider a system having two continuous states x^1 and x^2, and two state thresholds $[\delta_{e_x}^1 = 1] \leftrightarrow [x^1 \leq 0]$, $[\delta_{e_x}^2 = 1] \leftrightarrow [x^2 \leq 0]$, so that the system has four modes. Each mode corresponds to an orthant of the Cartesian plane, where $i = 1$ corresponds to the positive orthant and the other orthants are numbered clockwise. The system has two inputs $-50 \leq u^1 \leq 50$ and $-50 \leq u^2 \leq 50$, and the vectors and matrices that define Eq. (1) for $i = 1, \ldots, 4$ are $f_1 = f_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $f_2 = f_3 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$, $B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1.4 \end{bmatrix}$, $B_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1.5 \end{bmatrix}$, $B_3 = \begin{bmatrix} 0 & 0 \\ 0 & 1.15 \end{bmatrix}$, $B_4 = \begin{bmatrix} 1 & 0 \\ 0 & 2.3 \end{bmatrix}$. Moreover, there are additional constraints on the inputs: when in mode $i = 1$ it must hold that

3 Different strategies may be proposed here, for example applying $u_c(t)$ during the interval $[t, t + \max\{q_{x(t)}^*(0), T_s\}]$, where T_s is a given minimum time interval (possibly covering more than one optimal event instants) to prevent out-of-time computation problems due to a small duration $q_{x(t)}^*(0)$.

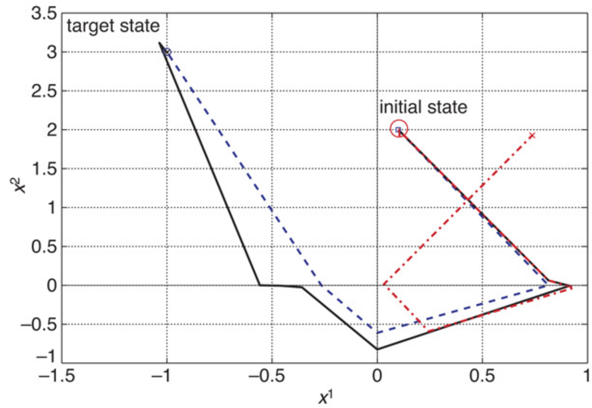


Fig. 3. Event-driven MPC example, nominal (dashed) and disturbed (solid). Disturbed open-loop event-driven optimal control (dash-dotted), same disturbance profile.

$u^2 \geq -2$, $[i = 2] \to [u^2 \leq -0.5]$, $[i = 3] \to [u^2 \leq 2]$, $[i = 4] \to [u^1 \leq 2]$ and $[i = 4] \to [-0.5 \leq u^2 \leq 2]$.

The controller must steer the state of the system from $x_0 = \left[\begin{smallmatrix} 0.1 \\ 2 \end{smallmatrix}\right]$ to $\hat{x} = \left[\begin{smallmatrix} -1 \\ 3 \end{smallmatrix}\right]$ while minimizing function (16) with norm $p = \infty$, $Q_1 = \left[\begin{smallmatrix} 10 & 0 \\ 0 & 10 \end{smallmatrix}\right]$, $Q_2 = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$, $R_1 = \left[10^{-3} \quad 10^{-3}\right]$, $R_2 = 1$, and $0.1 \leq q \leq 50$. We have set $\hat{q} = 0.1$, $\hat{v} = u_\infty \hat{q}$, where $u_\infty = \left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right]$. The closed-loop system is perturbed by input-additive disturbances, so that the continuous state dynamics is $\dot{x}(t) = B_{i(t(k))}\left(u(t(k)) + \nu(k)\right) + f_{i(t(k))}$, for all $t \in [t(k), t(k+1))$, where $\nu(k)$, $k \in \mathbb{Z}_{0+}$, is a sequence of time-uncorrelated stochastic vectors with independent components uniformly distributed in $[-0.1, 0.1]$.

Fig. 3 reports the continuous-time trajectories generated by the eMPC controller with a prediction horizon of 4 events, and reiterated for 8 consecutive steps. In the undisturbed case (dashed line) the closed-loop eMPC strategy trajectory coincides with the open-loop optimal one; four control actions, corresponding to four mode switches, are required to bring the system to the target state. In the presence of disturbances the eMPC (solid line) is able to counteract them, and to still bring the system close to $\hat{x}$, even if a larger number of control actions with respect to the undisturbed case is required. The trajectory obtained by the open-loop optimal policy under the effect of the same disturbance realization is plotted as a dash-dotted line, showing that the effects of the disturbance are not negligible, due to the switching nature of the system.

4.2. Conditions for convergence of closed-loop eMPC

In this section we prove closed-loop convergence properties of eMPC by resorting to classical techniques employed in nonlinear and hybrid MPC analysis (Alamir & Bornard, 1994; Bemporad & Morari, 1999; Lazar, Heemels, Bemporad, & Weiland, 2007; Mayne, Rawlings, Rao, & Scokaert, 2000) and specializing them to the present continuous-time and event-driven context.

In the following analysis we consider the case of terminal equality constraint $\mathcal{X}_N = \{\hat{x}\}$ and $\mathcal{T}_N = \{\hat{t}\}$ or $\mathcal{T}_N = \mathbb{R}_{0+}$. The approach can be extended to the case of terminal polyhedral sets, see (Di Cairano, 2008, Sec. 5.4.2).

Definition 4.1. A state value $\bar{x} = \left[\begin{smallmatrix} \bar{x}_c \\ \bar{x}_b \end{smallmatrix}\right]$ is an equilibrium for the icHA in mode $\bar{\imath}$ if and only if there exists a steady-state input value $\bar{u}_\infty = \left[\begin{smallmatrix} \bar{u}_{c,\infty} \\ \bar{u}_{b,\infty} \end{smallmatrix}\right]$ and $\bar{\delta}_{e,\infty}$ such that: (i), $B_{\bar{\imath}}\,\bar{u}_{c,\infty} + f_{\bar{\imath}} = 0$; (ii), $\bar{\delta}_{e,\infty} = f_{EG}(\bar{x}_c, \bar{u}_{c,\infty}, t)$, for all $t \geq 0$, where f_{EG} is the event generator (2); (iii), $\bar{x}_b = f_{aFSM}(\bar{x}_b, \bar{u}_{b,\infty}, \bar{\delta}_{e,\infty})$; (iv), $\bar{\imath} = [1 \ldots s] \cdot f_{MS}(\bar{x}_b, \bar{u}_{b,\infty}, \bar{\delta}_{e,\infty})$.

Definition 4.1 requires that the input $\bar{u}_\infty$ maintains the continuous state, the discrete state, and the mode constant. For instance the target state $\left[\begin{smallmatrix} -1 \\ 3 \end{smallmatrix}\right]$ in the example of Section 4.1 is an equilibrium for mode $i = 4$ with steady-state input $\bar{u}_\infty = \left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right]$.

4.2.1. Terminal equality constraints

When the terminal constraints $x(N) = \hat{x}$, $t(N) = \hat{t}$ are included, the terminal cost F is removed from (16). Let $\chi(k) \triangleq \left[\begin{smallmatrix} x(k) \\ t(k) \end{smallmatrix}\right]$ denote the full state of the eMLD system at a generic time $t(k)$. Let $J^*(\chi(k))$ be the optimal cost of Problem (15), $\mathbf{X}^*(\chi(k)) = [\chi^*_{\chi(k)}(1), \ldots, \chi^*_{\chi(k)}(N)]$ and $\mathbf{w}^*(\chi(k)) = [w^*_{\chi(k)}(0), \ldots, w^*_{\chi(k)}(N-1)]$ be[4] the optimal state and optimal input trajectories, respectively, where $w^*_{\chi(k)}(i) = \left[\begin{smallmatrix} v^*_{\chi(k)}(i) \\ q^*_{\chi(k)}(i) \end{smallmatrix}\right]$. For the sake of notation, denote by $\chi(k+1) = G(\chi(k), w(k))$ the state update function (14).

Theorem 4.1. *Let $\mathcal{X}_N = \{\hat{x}\}$, $\mathcal{T}_N = \{\hat{t}\}$, and let $\underline{q} = 0$. Let $\hat{x}$ be an equilibrium with corresponding steady-state input $\hat{u}$, let $\hat{q} = 0$, $\hat{v} = \left[\begin{smallmatrix} 0 \\ \hat{u}_b \end{smallmatrix}\right]$, and assume that $(\hat{x}, \hat{t}, \hat{q}, \hat{v})$ satisfies constraints (15c) and (15d). Let L be a function such that $L(\hat{x}, \hat{t}, \hat{q}, \hat{v}) = 0$ and $L(x, t, q, v) \geq \psi\left(\left\|\left[\begin{smallmatrix} x-\hat{x} \\ t-\hat{t} \end{smallmatrix}\right]\right\|\right)$, where $\psi : \mathbb{R}_{0+} \to \mathbb{R}_{0+}$ is a nondecreasing function, $\psi(0) = 0$, $\psi(\alpha) \in \mathbb{R}_+$, for all $\alpha \in \mathbb{R}_+$. If Problem (15) is feasible for the initial state $x_0 = x(0)$ and $t_0 = 0$, then it is recursively feasible, i.e., it is feasible for all $\left[\begin{smallmatrix} x(k+1) \\ t(k+1) \end{smallmatrix}\right] = G(\chi(k), w^*_{\chi(k)}(0))$, $k \in \mathbb{R}_+$, and moreover $\lim_{t \to \hat{t}} x(t) = \hat{x}$.*

Proof. We first prove feasibility. Assume Problem (15) admits a solution at event step $k = 0$, $\chi(k) = \left[\begin{smallmatrix} x(0) \\ 0 \end{smallmatrix}\right]$. Let $\mathbf{w}^*(\chi(k))$ be the corresponding optimal input sequence, and let $\chi(k+1) = \chi^*_{\chi(k)}(1)$. Consider the input sequence $\tilde{\mathbf{w}}(\chi(k+1)) = \left[w^*_{\chi(k)}(1), \ldots, w^*_{\chi(k)}(N-1), \left[\begin{smallmatrix} 0 \\ \hat{u}_b \\ 0 \end{smallmatrix}\right]\right]$. Then, $\chi_{\chi(k+1)}(i) = \chi_{\chi(k)}(i+1)$, for $i \in \mathbb{Z}_{[0,N-1]}$, and

$$\chi_{\chi(k+1)}(N) = G\left(\chi_{\chi(k+1)}(N-1), \begin{bmatrix} 0 \\ \bar{u}_{b,\infty} \\ 0 \end{bmatrix}\right) = G\left(\chi^*_{\chi(k)}(N), \begin{bmatrix} 0 \\ \bar{u}_{b,\infty} \\ 0 \end{bmatrix}\right) = \chi^*_{\chi(k)}(N).$$

Thus the sequence $\tilde{\mathbf{w}}(\chi(k+1))$ satisfies the dynamical, operating, and terminal constraints in (15), so Problem (15) is solvable at time $k+1$. By induction, solvability at $k = 0$ implies solvability at all event steps $k \in \mathbb{Z}_+$.

We now prove convergence. Because of optimality, $J^*(\chi(k+1)) \leq J(\chi(k+1), \tilde{\mathbf{w}}(\chi(k+1)))$, where

$$J(\chi(k), \tilde{\mathbf{w}}(\chi(k+1))) = J^*(\chi(k)) - L(x(k), t(k), v^*_{\chi(k)}(0), q^*_{\chi(k)}(0)), \tag{21}$$

and hence $J^*(\chi(k+1)) \leq J^*(\chi(k))$. Since $J(\chi(k))$ is lower bounded by 0 and is not increasing with k, there exists $\lim_{k\to\infty} J(\chi(k)) = J_\infty$, so that $\lim_{k\to\infty} J(\chi(k+1)) - J(\chi(k)) = 0$, and hence $\lim_{k\to\infty} L(x(k), t(k), v(k), q(k)) = 0$. The last implies

[4] When using the full state χ we use the notation $\chi(k)$ for $\chi(t(k))$, as stated in Section 1.1.

$\lim_{k\to\infty} \psi(\|\chi(k) - \hat{\chi}\|) = 0$, where $\hat{\chi} = \begin{bmatrix}\hat{x}\\ \hat{t}\end{bmatrix}$. Assume by contradiction $\|\chi(k) - \hat{\chi}\| \not\to 0$. Then there exists $\epsilon \in \mathbb{R}_+$ such that for all $k \in \mathbb{Z}_{0+}$ there exists $k \geq \bar{k}$ such that $\|\chi(k) - \hat{\chi}\| \geq \epsilon$. As ψ is nondecreasing, $\psi(\|\chi(k) - \hat{\chi}\|) \geq \psi(\epsilon)$, which contradicts $\lim_{k\to\infty} \psi(\|\chi(k) - \hat{\chi}\|) = 0$. Therefore, $\lim_{k\to\infty} \left\|\begin{matrix}x(k) - \hat{x}\\ t(k) - \hat{t}\end{matrix}\right\| = 0$ and hence $\lim_{k\to\infty} x(k) = \hat{x}$, $\lim_{k\to\infty} t(k) = \hat{t}$. By linearity of $x(t)$ between two consecutive events it also follows that $\lim_{t\to\hat{t}} x(t) = \hat{x}$. □

Note that function ψ is also called a function of class $\mathcal{M}$ in (Lazar et al., 2007). The condition of ψ being of class $\mathcal{M}$ was sufficient in the proof of Theorem 4.1 to ensure convergence of $\|\chi(k) - \hat{\chi}\|$ to zero when $\psi(\|\chi(k) - \hat{\chi}\|)$ converges to zero, a property which is directly assumed in (Alamir & Bornard, 1994). The assumption about the existence of function ψ is satisfied when L is strictly convex with respect to x and t.

The trajectory obtained by the approach of Theorem 4.1 is defined for $t \in [0, \hat{t}]$. For $t \geq \hat{t}$ the target state can be maintained by applying a steady-state input, if this exists according to Definition 4.1. Despite the convergence of x is ensured in finite time $\hat{t}$, it may be asymptotic with respect to the number k of events.

Next Theorem 4.2 considers the case in which the terminal time $t(N)$ is unconstrained (i.e., $\mathcal{T}_N = \mathbb{R}_{0+}$), L does not depend on t (for instance, $Q_2 = 0$ in the definition of $L(x, t, v, q)$), and in which $\underline{q} > 0$. The lower bound $\underline{q} > 0$ ensures a minimum dwell time of the system between two consecutive mode switches, which may prevent for instance that $q(k)$ gets too small for solving the next optimization problem.

Theorem 4.2. *Let $\mathcal{X}_N = \{\hat{x}\}$, $\mathcal{T}_N = \mathbb{R}_{0+}$ and let $\underline{q} > 0$. Let $\hat{x}$ be an equilibrium with corresponding steady-state input $\hat{u}$, and assume there exists $\hat{q}$, $\underline{q} \leq \hat{q} \leq \bar{q}$, such that $(\hat{x}, \hat{t}, \hat{q}, \hat{v})$ satisfies constraints (15c) and (15d) for $\hat{v} = \begin{bmatrix}\hat{q}\\ \hat{u}_b\end{bmatrix}$ and for all $\hat{t} \geq 0$. Let L be independent of t, be such that $L(\hat{x}, t, \hat{q}, \hat{v}) = 0$, for all $t \in \mathbb{R}$, and $L(x, t, q, v) \geq \psi(\|x - \hat{x}\|)$, where $\psi : \mathbb{R}_{0+} \to \mathbb{R}_{0+}$ is a nondecreasing function, $\psi(0) = 0$, $\psi(\alpha) \in \mathbb{R}_+$, for all $\alpha \in \mathbb{R}_+$. If Problem (15) is feasible for the initial state for $x_0 = x(0)$ and $t_0 = 0$, then it is recursively feasible, i.e., it is feasible for all $\begin{bmatrix}x(k+1)\\ t(k+1)\end{bmatrix} = G(\chi(k), w^*_{\chi(k)}(0))$, $k \geq 0$, and $\lim_{t\to\infty} x(t) = \hat{x}$.*

The proof is similar to the proof of Theorem 4.1 and it can be found in (Di Cairano, 2008, Sec.5.4.2).

In order to have that $\lim_{k\to\infty} v(k) = \hat{v}$, $\lim_{k\to\infty} q(k) = \hat{q}$, it is enough to assume that $L(x, t, q, v) \geq \psi\left(\left\|\begin{matrix}x - \hat{x}\\ v - \hat{v}\\ q - \hat{q}\end{matrix}\right\|\right)$.

Note that the eMPC controller designed in Section 4.1 satisfies the hypotheses of Theorem 4.2.

5. Conclusions

We have introduced integral continuous-time hybrid automata (icHA), a special class of continuous-time hybrid dynamical models that can be suitably controlled through numerically viable optimization tools. The key idea for reformulating the infinite-dimensional optimization problem into a tractable finite-dimensional one is to use an event-driven control strategy. Differently from standard optimization-based discrete-time techniques, the event-driven approach guarantees that the constraints are enforced continuously in time, while keeping the computation load comparable. This paper only scratched the surface of the modelling power of the approach, by emphasizing for instance the relations that exist between icHA and linear hybrid automata. Finally, we have shown how closed-loop control strategies for icHA can be designed through receding horizon ideas, for which we have provided sufficient conditions for finite time and asymptotic convergence.

References

Alamir, M., & Bornard, G. (1994). On the stability of receding horizon control of nonlinear discrete-time systems. *Systems and Control Letters*, *23*(4), 291–296.

Bemporad, A. (2004). Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE Transactions of Automatic Control*, *49*(5), 832–838.

Bemporad, A., Di Cairano, S., & Júlvez, J. (2006). Event-based model predictive control and verification of integral continuous-time hybrid automata. In *Lect. notes in computer science*: *Vol. 3927*. *Hybrid systems: Computation and control* (pp. 93–107). Springer-Verlag.

Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, *35*(3), 407–427.

Borrelli, F., Falcone, P., & del Vecchio, C. (2007). Event-based receding horizon control for two-stage multi-product production plants. *Control Engineering Practice*, *15*(12), 1556–1568.

Cassandras, C. G., & Mookherjee, R. (2003). Receding horizon control for a class of hybrid systems with event uncertainties. In *Proc. of American Control Conf*.

De Schutter, B., & van den Boom, T. T. J. (2001). Model predictive control for max-plus-linear discrete event systems. *Automatica*, *37*(7), 1049–1056.

Di Cairano, S. (2008). Model predictive control of hybrid dynamical systems: Stabilization, event-driven, and stochastic control. Ph.D. Thesis, Dipartimento Ingegneria dell'Informazione, Universita' di Siena. http://phd.dii.unisi.it/people/tesi/172_stefano_dicairano.pdf.

Di Cairano, S., & Bemporad, A. (2006). An equivalence result between linear hybrid automata and piecewise affine systems. In *Proc. of 45th IEEE conf. on decision and control*. (pp. 2631–2636).

Henzinger, T. A. (1996). The theory of hybrid automata. In *Proc. of the 11th ann. IEEE symp. on logic in computer science*. (pp. 278–292).

Henzinger, T. A., Ho, P. H., & Wong-Toi, H. (1997). HyTech: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer*, *1*(1–2), 110–122.

Julvez, J., Bemporad, A., Recalde, L., & Silva, M. (2004). Event-driven optimal control of continuous petri nets. In *Proc. of 43th IEEE Conf. on decision and control*. (pp. 69–74).

Lazar, M., Heemels, W. P. M. H., Bemporad, A., & Weiland, S. (2007). Discrete-time non-smooth nonlinear MPC: Stability and robustness. In *Lect. notes control inf. sciences*: *vol. 358*. *Assessment and future directions of nonlinear model predictive control* (pp. 93–103). Springer-Verlag.

Lygeros, J., Johansson, K. H., Simic, S. N., Zhang, J., & Sastry, S. S (2003). Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, *48*, 2–17.

Maciejowski, J. M. (2002). *Predictive control with constraints*. Prentice Hall, NJ: Englewood Cliffs.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*(6), 789–814.

Miao, L., & Cassandras, C. G. (2007). Receding horizon control for a class of discrete-event systems with real-time constraints. *IEEE Transactions on Automatic Control*, 825–839.

Silva, M., & Recalde, L. (2002). Petri nets and integrality relaxations: A view of continuous Petri net models. *IEEE Transactions on Systems, Man, and Cybern.*, *32*(4), 314–327.

Torrisi, F. D., & Bemporad, A. (2004). HYSDEL — A tool for generating computational hybrid models. *IEEE Transactions on Controls Systems Technology*, *12*(2), 235–249.

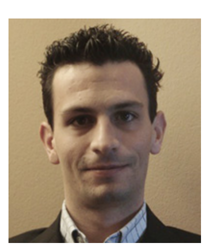
Van Beek, D. A., Pogromsky, A., Nijmeijer, H., & Rhooda, J. E. (2004). Convex equations and differential inclusions in hybrid systems. In *Proc. of 43th IEEE conf. on decision and control*. (pp. 1424–1429).

Williams, H. P. (1993). *Model building in mathematical programming*. 3rd ed.: John Wiley & Sons.

Wong-Toi, H. (1997). The synthesis of controllers for linear hybrid automata. In *Proc. of 36th IEEE conf. on decision and control*. (pp. 4607–4612).

Xu, X., & Antsaklis, P. J. (2003). Results and perspectives on computational methods for optimal control of switched systems. In *Hybrid systems: Computation and control* (pp. 540–555). Springer-Verlag.

Xu, X., & Antsaklis, P. J. (2002). A linear programming approach to time optimal control of integrator switched systems with state constraints. In *Proc. of 41th IEEE conf. on decision and control*. (pp. 2168–2173).



S. Di Cairano received the Master degree (Laurea) cum laude in Computer Engineering in 2004, and the PhD degree in Information Engineering in 2008, both from the University of Siena, Italy. During his doctoral studies, his main research areas were hybrid dynamical systems and hybrid model predictive control, and their application to networked control systems. He was a visiting student at the Information and Mathematical Modeling Department, the Technical University of Denmark, in 2002–2003. He was visiting graduate student at the Control and Dynamical Systems Department, the California Institute of Technology, in 2006–2007. In April 2008, he joined the Powertrain Control R&A Dept, the Ford Motor Company, Dearborn, MI, where he is investigating and developing advanced control strategies for powertrain, vehicle dynamics, and production planning. His primary research interests are in model predictive control, hybrid control strategies, and their applications in the automotive industry.