Brief paper

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

S. Di Cairano, A. Bemporad, J. Júlvez

A R T I C L E  I N F O

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

A B S T R A C T

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
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 & Morkoobkee, 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., $v$, $x$) superscripts (e.g., $v^i$) vector components. Subscripts are used for system modes (e.g., $A_i$, $b_i$) and for time instants in a sequence of time instants ($t_i$). Bold symbols denote finite sequences of variables, e.g., $a \triangleq (a(t_i))_{i \in I}^n$. The sets $\mathbb{R}$ and $\mathbb{Z}$ indicate the set of real numbers and integer numbers, respectively, $\mathbb{R}_+$, $\mathbb{R}_0$, $\mathbb{R}_0^+$, $\mathbb{R}_+$, $\mathbb{R}_-$ 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 $\text{Int}(\mathcal{X})$ and its closure by $\text{Cl}(\mathcal{X})$. 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, $\|z\|_Q^2 \triangleq \sum (Qz)^2$, $\|z\|_Q^2 \triangleq \sum (Q^t z)^2$, and $\|z\|_M^2 \triangleq z^t M z$ where $Q \geq 0$. The symbol $I$ indicates the identity matrix of appropriate dimensions, while we use the bold 0 and 1 to indicate matrices and vectors entirely composed of zeros and ones, respectively. The symbol $\rightarrow$ denotes logical implication ($\Rightarrow$) and the symbol $\leftrightarrow$ logical equivalence ($\Leftrightarrow$).

Given a signal $a(\cdot) : \mathbb{R}_+ \rightarrow A$ we indicate its continuous-time trajectory by $a(s), s \in \mathbb{R}_+$, and its event-driven trajectory by $a(j), j \in \mathbb{Z}_0$, where $j$ is the time instant at which the $j$th 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 $Z_{f,1}$, are indicated by $a(t) = a_{0(t)}(t), \ldots, a_{nf(t)}(t)$.

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

\[
\dot{x}_c(t) = B_c u_c(t) + f_c(t),
\]

where $x_c \in \mathbb{R}^{n_c}$ and $u_c \in \mathbb{R}^{n_c}$ are the continuous components of the state and input vectors, respectively, and $i \in I = \{1, 2, \ldots, s\}$ 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 $\dot{x}_i = f(x_i(t), u_i(t))$, model (1) can be interpreted as a piecewise zero-order approximation of the state-transition function with respect to the system state $x_i$ and a first-order approximation with respect to the input vector $u_i$. Similarly to the piecewise affine (PWA) approximation $x_c = A_c x_c + B_c u_c + f_c$, 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 $\delta_c$ according to the linear threshold conditions

\[
[\delta_c^i(t) = 1] \iff \left[ E_i^c \begin{bmatrix} x_c(t) \\ t \\ 1 \end{bmatrix} \leq F_i^c \right], \quad i \in \mathbb{Z}_{[1,n]}.
\]

\[
[\delta_c^i(t) = 0] \iff \left[ E_i^c u_c(t) \leq F_i^c \right], \quad i \in \mathbb{Z}_{[1,n]}.
\]

and we get $\delta_c = \left[ \delta_1^c, \ldots, \delta_{n_c}^c \right] \in \{0, 1\}^{n_c}$, $n_c \triangleq n_c^+ + n_c^-$, as the vector of endogenous binary input variables. The ichA may also be excited by exogenous binary input signals $u_{\Theta} \in \{0, 1\}^{m_{\Theta}}$ entering the mode selector and the asynchronous finite state machine.

**Definition 2.1.** $P \mathcal{E}_{(m_{\Theta}, m_c)}$ is the set of piecewise constant functions $u(t) \triangleq \left[ u_{\Theta}(t) \right] : \mathbb{R} \rightarrow \mathbb{R}^{m_{\Theta}} \times \{0, 1\}^{m_c}$ 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.
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 \( u \in R \to R^m \times [0, 1]^{m_0} \in \mathcal{P}E_{(m, m_0)} \), i.e., the input trajectory is piecewise constant along time. □

**Definition 2.2.** An event occurs whenever an endogenous input \( \delta_e \) or an exogenous input \( \delta_i \) changes value. Given the initial time \( t_0 \in R \), the event instants are defined recursively as

\[
\tau_k \triangleq \min \{ t : (u_t(t), u_{t+1}(t), \delta_e(t)) \neq (u_{t-1}(t), u_{t-1}(t), \delta_e(t-1)) \},
\]

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

The Boolean state \( \xi_{t+1} \in [0, 1]^{m_0} \) is defined as

\[
\xi_{t+1} = x_{t+1} = x_{t+1} \circ \nu_{t+1} \circ \delta_{e}(t),
\]

and \( f_{ASM} : [0, 1]^{m_0+m+n_r} \to [0, 1]^{m_0} \) is the Boolean function defining the transitions of the asynchronous finite state machine. The Boolean state \( \xi_{t} \) remains constant, \( \xi_{t} = x_{t} \), during the whole interval \( t_{k-1} \leq t < t_k \). At the instant time \( t_k \), the Boolean state switches to the new value \( f_{ASM}(x_{t_k}, \delta_{e}(t_k), x_{t_k}) \), 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

\[
i(t) = \{1, 2, \ldots, s\} \cdot f_{MS}(\xi_t(t), u_t(t), \delta_e(t)),
\]

where \( f_{MS} : [0, 1]^{m_0+m+n_r} \to [0, 1]^{s} \) is a Boolean function satisfying the mutual exclusivity relation \( 1 \cdot f_{MS}(\xi_t(t), u_t(t), \delta_e(t)) = 1 \)

for all \( (\xi_t(t), u_t(t), \delta_e(t)) \in [0, 1]^{m_0+m+n_r} \). Note that when \( u_t \) and \( \delta_e \) 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 \( t = \{t_j\}^{j=0}_{j} \) be a finite sequence of ordered time instants \( t_0 < t_1 < \cdots < t_{k-1} \). Let \( u_t : R \to R^m \) be a piecewise constant function with breakpoints in the instants in \( u_t \) and \( u = \{u_t(t)\}_{t=0}^{\infty} \) the corresponding sequence of levels. For the affine dynamics \( \dot{x}_t(t) = Ax_t(t) + Bu(t) + f \), the state value at \( t_j \in Z^{[0, h-1]} \), is a nonlinear function of \( u \) and \( u_t \). However, in the case of integral dynamics \( (A = 0) \), \( x_t(t) = x_t(0) + \sum_{k=0}^{j} (t_k + 1) f(t_k) \). Consider now a system with switched integral dynamics (1), and let the set of mode-switch instances \( \{ t : \xi_t(t) \neq \xi_t(t+1), \delta_e(t) = 0 \} \subset R \). 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

\[
x_{t+1} = x_t + B_{u(t)} u_t + f_{u(t)} \delta_e(t)
\]

(7a)

\[
t_k + 1 = t_k + \delta_e(t_k)
\]

(7b)

where \( k \) is the event counter, \( x_{t_k} = x_{t_k} \), \( t_k = t_k \), \( i_k(t_k) = i_k(t_k) \), \( q \) is the time interval between event instances \( k \) and \( k+1 \), \( u_{t+1}(k) = q u_{t}(k) \) is the integral over time period \( q \) of the input \( u_t = u_t(k) \). Note that in (7b) time \( t \) is treated as an additional state variable. The controlled variables are the input integral \( v_{t}(k) \) and the input level duration \( q(k) \), from which the actual input \( u_t(k) = v_{t}(k) / q(k) \) to be applied to the continuous-time system is immediately recovered.

The event generator becomes

\[
\{ \delta_e(t_k) = 1 \Rightarrow [E_i^{\mathbf{x}_t(t)} \leq F_v(t_k) ] \}, \quad i \in \{1, \ldots, s\}
\]

(8a)

\[
\{ \delta_e(t_k) = 1 \Rightarrow [E_i^{q(t)_k} \leq E_i^{q(t)_k} ] \}, \quad i \in \{1, \ldots, s\}
\]

(8b)

where \( \delta_e(t) = \delta_e(t_k) \) and \( c(t) = c(t_k) \), 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 (8b) by multiplying by \( q(t) \) both sides of the right clamp. The mode selector equation becomes

\[
i(t) = \{1, 2, \ldots, s\} \cdot f_{MS}(x_{t_k}(k), u_t(k), \delta_e(k)),
\]

(9)

where \( i(t) = i(k) \), for all \( t \in [t_k, t_{k+1}) \), as a consequence of the event definition, and \( f_{MS}(x_{t_k}(k), u_t(k), \delta_e(k)) = f_{MS}(f_{ASM}(x_{t_k}(k), u_t(k), \delta_e(k)), u_t(k), \delta_e(k), \delta_e(k)) \) because of (6) and of the definition of \( \xi_{t+1} \). Finally, Eq. (5) is already defined with respect to the event instants.

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

\[
\{ (\delta_e(t), u_t(k), u_t(k)) = (\delta_e, \bar{u}_t, \bar{u}_t) \}

(10)

We consider two different cases: (i) the value of \( u_t \) or \( u_{t+1} \) changes, so that an event is externally forced, for instance by a controller, (ii) the value of \( \delta_e \) 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 \( u_t(k), u_{t+1}(k) \) are constant between event instants and simply restrict condition (10) to

\[
\{ \delta_e(t+1) = \bar{\delta}_e \}, \quad \forall t \in [t_k, t_{k+1}) \}

(11)

i.e., the current mode is kept until the next endogenous event. Moreover, as \( \delta_e(k) \) 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 \( \text{cod}(\cdot) : [0, 1]^{m} \to Z_{0+} \) associate a nonnegative integer number \( d \) to each allowed value of vector \( \delta_e \in [\delta_e^{0} \ldots \delta_e^{n_r}] \) defined in (2a). For example \( d \) may be the integer whose binary encoding is \( \delta_e \). Define the matrix \( E_i(d) \) and the vector \( F_i(d) \) by collecting the rows in the inequalities of the EG (8) which are satisfied for \( e_i \) such that \( \text{cod}(\delta_e(k)) = d \). Define \( \tilde{E}_i(d), \tilde{F}_i(d) \) by collecting as rows the inequalities of the EG (8), which are not satisfied for \( \delta_e \) such that \( \text{cod}(\delta_e) = d \). Hence (11) is equivalently replaced by

\[
\{ \text{cod}(\delta_e(k)) = d \}

(12)
As an example, consider two thresholds $\delta_0^1 = 1 \iff [x \leq 0]$, $\delta_0^2 = 1 \iff [x \leq 1]$. The matrices associated to $\delta_0^i = [0 \ 1 \ 1]^T$, where cod($\delta_0^i$) = 1, are $\hat{E}^i(1) = 1$, $\hat{F}^i(1) = 1$, collecting the second threshold condition (satisfied), and $\hat{F}^i(1) = 1$, $\hat{F}^i(1) = 0$.

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

$$\left[\begin{array}{c} \text{cod}(\delta_r^i(k)) = d \\ \hat{E}^d(d) \\ \hat{F}^d(d) + \epsilon T \end{array}\right] \implies \left[\begin{array}{c} \hat{E}^i(t-k+1) \\ \hat{F}^i(t-k+1) \end{array}\right],$$

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

**Proof.** Because of the integral dynamics, the state trajectory for $t \in [t_k, t_{k+1})$ is the line $x(t) = x(t-k) + \gamma(t-k)$, where $\gamma = \{(k+1)-(k-1)\}$. Condition (12) and condition (13) define two polyhedra, $\mathcal{P}$ and $\mathcal{P}'$, respectively. Since $\epsilon > 0$, $\mathcal{P}' \supset \mathcal{P}$ and if $\epsilon \to 0$, $\mathcal{P}' \to \mathcal{P}$. Condition (13) ensures that $x(t-k+1) \in \mathcal{P}'$, and (12) ensures that $x(t) \in \mathcal{P}$, hence for all $t \in [t_k, t_{k+1})$, $x(t) \in \mathcal{P}$. Thus, by linearity of the trajectory there exists $\gamma(t)$ such that $x(t) \in \mathcal{P}$. For all $t \in [t_k, t_{k+1})$, $\gamma(t)$ exists. Since for $\epsilon \to 0$, $\mathcal{P}' \to \mathcal{P}$, then for $\epsilon \to 0$ also $\gamma(t)$ exists. \hfill $\Box$

The effect of adding $\epsilon$ in Eq. (13) is to expand the polyhedron $\mathcal{P}$. This is necessary for allowing $\delta_r^i(k+1) \neq \delta_r^i(k)$, otherwise the system would be constrained to remain always in the same mode. The value of $\gamma(t)$ is used for the trajectory spends in $\mathcal{P}' \setminus \mathcal{P}$. When $\mathcal{P}' \to \mathcal{P}$, $\gamma(t)$ is used.

Note that, within a given mode $i(k), x(k+1)$ is an affine function of $x(k), q(k)$, and $v(k) \triangleq \left[\begin{array}{c} v_i(k) \\ v_0(k) \end{array}\right]$, where $v_i(k) = u_i(k)$, so that (13) can be reformulated as a set of mixed-integer inequalities on $x(k), q(k), v(k), \delta_r(k)$ (Williams, 1999). Indeed, Eqs. (5), (7), (8), (9), (13) represent a DHA that can be converted into the event-driven MILP (eMILP) system

$$x(k+1) = Ax(k) + B_1w(k) + B_2d(k) + B_2z(k) + B_2,$$  

$$t(k+1) = t(k) + q(k),$$  

$$E_2d(k) + E_2z(k) \leq E_1w(k) + E_1x(k) + E_2 + E_0q(k),$$

where $w(k) \triangleq \left[\begin{array}{c} w_i(k) \\ q(k) \end{array}\right]$, for instance using the tool HYSDS (Torrisi & Bemporad, 2004). Differently from the standard discrete-time MILP (MILP) model (Bemporad & Morari, 1999), in the eMILP (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 $x_i(t_{k+1}) = \Phi_i x_i(t_k) + \mu_i$ can be included as follows. To model resets one must add reset modes $i \in \{1 + 1, \ldots, s\}$, modify (7a) into $x_i(k+1) = (\Phi_i x_i(k) + \mu_i) + B_i \nu_i(k) + f_iq(k)$, and (7b) into $t(k+1) = t(k) + \Gamma_i q(k)$, where in modes $i \in \{1, \ldots, s\}$, $\Phi_i = I, \mu_i = 0$, and $\Gamma_i = 1$, while in reset modes $i \in \{1 + 1, \ldots, s\}, \Phi_i = 0, \mu_i = 0$, and $\Gamma_i = 0$.

When a reset occurs Eq. (14) does not apply, as $t_{k+1} = t_k$. In this case, assuming that two consecutive resets cannot occur, the continuous-time trajectory $x_i(t)$ may be discontinuous in $t_k$, and is defined as follows. Let $j$ be the mode immediately before the reset mode $i$. Then, $x_j(t) = x_j(t_k-1) + B_j \nu_j(k-1) + f_jq(k-1), x_i(t_k) = \Phi_i x_i(k-1) + B_i \nu_i(k-1) + f_iq(k-1) + \mu_i$, and $\lim_{t \to t_k^-} x_i(t) = x_i(t_k)$. In other words, we define the state trajectory $x_i(t)$ as right continuous.

In accordance with the above definition of resets, in eMILP models these are instantaneous, contrary to resets in discrete-time MILP models that are constrained to last one sampling interval (Torrisi & Bemporad, 2004). An equivalent event-driven PWA system (ePWA) of the eMILP (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 iCHAI has some limitations, the iCHAI covers 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 iCHAI, as intuitively shown by the example in Section 3.3. A formal proof of the capabilities of iCHAI to model linear hybrid automata can be obtained by resorting to an equivalent piecewise affine representation of the iCHAI 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, & Jüle 2005) an example of formal verification of a LHA performed through its equivalent iCHAI formulation was given.

Another class of systems that can be modelled as iCHAI 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 iCHAI. The application to continuous Petri net control of the techniques proposed here has been presented in (Jüle 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 iCHAI (1)–(6)

$$\min_{x(t), q(t), v(t), \nu(t), \rho(t)} J(x, t, q, v),$$

s.t. 

ichai dynamics (1)–(6)  

$$g(x(t), q(t), v(t), \nu(t)) \leq 0, \quad k \in \mathbb{Z}_{[0,N-1]}$$

$$g_\nu(x(N), v(N)) \leq 0$$

$$x(0) = x_0, \quad t(0) = t_0,$$

where $\nu(\cdot)$ is a convex function of $x(t)$, $t(t), q(t), v(t), \nu(t) = \{q(k)\}_{k=0}^N$ are the time instants at which events occur, $x = \{x(k)\}_{k=0}^N$ are the corresponding state values, $q = \{q(k)\}_{k=0}^N$ are the durations of the time intervals between two consecutive events, $q(k) = t(k+1) - t(k)$, $v = \{v(k)\}_{k=0}^N$ are the input integrals computed on the corresponding intervals $\{t(k), t(k+1)\}_{k=0}^N$, $x_0$ 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 iCHAI dynamics (15b) as the equivalent eMILP 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

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

For instance, one can set \( L(x, t, v, q) = \|x - \hat{x}\|_p^2 + \|t - \hat{t}\|_p^2 + \|v - \hat{v}\|_p^2 + \|q - \hat{q}\|_p^2 \), \( F(x, t) = \|x - \hat{x}\|_p^2 + \|t - \hat{t}\|_p^2 \), \( p \in \{1, 2, \infty\} \), where \( \|\cdot\|_p \) 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, \( x(0) = \hat{x} \), 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 \( t \) or set \( \mathcal{T}_N \) can be formulated similarly.

The minimum-time criterion looks for the sequence \( (x, t, q, v) \) that minimizes the time needed to bring the system from \( x_0 \) to the final state \( \hat{x} \). This can be obtained by setting \( L(x(k), t(k), q(k), v(k)) = \|q(k) - \hat{q}(k)\|_p F(x(N), t(N)) = 0 \) 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 \( J(x, t, q, v) = \int_0^T \|u(t)\|_1 dt = \sum_{k=0}^{N-1} \int_{t(k)}^{t(k+1)} \|u(t)\|_1 dt \). 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 \( \hat{x}(t) \), that we assume piecewise linear and continuous (a special case is \( \hat{x}(t) = \hat{x}_c(t) \)).

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

Proposition 3.1. Let \( x_c(t) \) for all \( t \in [t_0, t_N] \), be the trajectory of continuous states of an iCA 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 \( [t_k, t_{k+1}] \), \( t \in \mathbb{Z}_{0,N-1} \) and continuous over \( [t_0, t_N] \). Then \( \text{max}_{t \in [t_0, t_N]} \|x_c(t) - \hat{x}_c(t)\|_\infty \) is finite.

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

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 \( x_c(t) \) similar to the ones used in the proof of Proposition 3.1, state constraints \( h \leq H(x(t)) \leq \bar{h}, t \in \{t, t+1\} \) can simply be enforced through constraints on the state at event instants only

\[ h \leq H(x(t)) \leq \bar{h}, \quad k \in \mathbb{Z}_{0,N}. \] (18)

Similarly, input bounds \( q \leq q(k) \leq \bar{q}, t \in \{t, t+1\} \) can be rewritten immediately as the linear constraints

\[ q(k) \leq v(k) \leq \bar{q}(k), \quad k \in \mathbb{Z}_{0,N-1}. \] (19)

Different input bounds for different modes can be enforced by the logical constraint \( [i(k) = \overline{i}] \rightarrow [\bar{q}(k) \leq v(k) \leq q(k)] \), where \( \bar{q} \) and \( \bar{i} \) are upper and lower bounds of the input while in mode \( \overline{i} \). 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

\[ q \leq q(k) \leq \bar{q}, \quad k \in \mathbb{Z}_{1,N-1}. \] (20)

A finite \( \overline{t} \) 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 \( q \) 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 \( q \) 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 \( Nq \). Finally, the lower bound \( q \) 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 \( t_c \leq t(k) \leq \bar{t}_c, k \in \mathbb{Z}_{1,N} \) 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 (C), 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
closed while \( x^2 = 1 \) means completely open. Instead of the differential inclusions (Henzinger et al., 1997) used for verification, we define the dynamics by \( \dot{x} = u(t) + f_i \), where \( f_i(t) = \begin{bmatrix} f_i(t) \\ 0 \end{bmatrix} \).

The actual state \( x(t) \) is \( x(t) = x(0) \dot{x} + \int_0^t u(s) ds \).

Consider the following problem: from the initial state \( \begin{bmatrix} x_{0c} \\ x_{0b} \end{bmatrix} \)

with \( x_{0c} = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \)

and \( x_{0b} = (Ar, Cl) \), the train must safely cross the gate in minimum time. For control purposes we have modelled the system as an iCHAs (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 \( \dot{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \) which was applied with \( N = 5 \), under the safety constraint imposing that the discrete state \((Cr, O)\) is never reached. The target state \( x_\delta \) 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, \( T_{\text{eps}} \) is the sampling period, \( T_{\text{cpu}} \) is the computation time. In Table 1, \( T_{\text{eps}} \) indicates the time period during which the current discrete-time solution violates the safety constraint. The violations occur in the intersampling period, \( T_{\text{eps}} = 2 T_i \) is the upper bound on \( T_{\text{eps}} \), 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 \( T_i \), also \( T_{\text{eps}} \) 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.

---

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 iCHAs and its eMLoC 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 \( \mathbf{v}^*(x(t)) \triangleq [v^{*-}_{x(0)}, \ldots, v^{*-}_{x(N)}] \) be the sequence of optimal input integral values, \( \mathbf{q}^*(x(t)) \triangleq [q^{*-}_{x(0)}, \ldots, q^{*-}_{x(N)}] \) be the sequence of input action durations, \( \mathbf{x}^*(x(t)) \triangleq [x^{*-}_{x(0)}, \ldots, x^{*-}_{x(N)}] \) be the predicted state values at event instants, and \( \delta^*(x(t)) \triangleq [\delta^{*-}_{x(0)}, \ldots, \delta^{*-}_{x(N)}] \) the corresponding time instants at which the events occur, computed from initial state \( x(t) \) and initial time \( t \).
3. Compute the input value \( u(t) = \begin{bmatrix} u^*_c(t) \\ v_{x(0)}^* \end{bmatrix} \) during the time interval \([t, t + \delta^{*-}_{x(0)}]\) to the iCHAs. \( ^3 \)
4. Set \( t_0 = t + \delta^{*-}_{x(0)}, x_0 = x(t + \delta^{*-}_{x(0)}) \) in (15) and go to Step 2.

The actual state \( x(t + \delta^{*-}_{x(0)}) \) at the end of each control action may be different from the predicted one \( x^{*-}_{x(0)} \) 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 \( t + \delta^{*-}_{x(0)} \). 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 \( [a^2_1 = 1 \leftrightarrow x^1 \leq 0], [a^2_2 = 1 \leftrightarrow x^2 \leq 0] \), 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

\[ f_1 = f_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, f_2 = f_3 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 1.4 \end{bmatrix}, B_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, B_4 = \begin{bmatrix} 1 \\ 2.3 \end{bmatrix}. \]

Moreover, there are additional constraints on the inputs: when in mode \( i = 1 \) it must hold that

\[ v^1_{x(0)} \geq 10 \]
Theorem 4.1. Let \( X_N = \{ \hat{x} \} \), \( T_N = \{ \hat{t} \} \), and let \( q = 0 \). Let \( \hat{x} \) be an equilibrium with corresponding steady-state input \( \hat{u} \), let \( \hat{q} = 0 \), \( \hat{v} = 0 \), 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 \( L(x, t, q, v) \geq \psi \left( \| x - \hat{x} \|^2 \right) \), 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 \( p_0 = 0 \), then it is recursively feasible, i.e., it is feasible for all \( \{ \hat{x}(k) \} \), \( \{ \hat{t}(k) \} \), \( \{ \hat{q}(k) \} \), \( \{ \hat{v}(k) \} \), \( \{ \hat{w}^*(k) \} \) for \( k \in \mathbb{N} \), and moreover \( \lim_{k \to \infty} x(k) = \hat{x} \).

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

\[
\chi_{x(k+1)+1}(N) = G \left( \chi(k+1) \left[ \begin{array}{c} 0 \\ 0 \\ \tilde{u}_b \end{array} \right]; \right)
\]

Thus the sequence \( \chi(k+1) \) 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, \( J^*(\chi(k+1)) \) is lower bounded by 0 and is not increasing with \( k \), there exists \( \lim_{k \to \infty} J^*(\chi(k)) = J^*(\hat{x}) \) such that \( \lim_{k \to \infty} J^*(\chi(k+1)) = J^*(\hat{x}) \) 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 \( \lim_{k \to \infty} J^*(\chi(k)) = J^*(\hat{x}) \) so that \( \lim_{k \to \infty} J^*(\chi(k+1)) = J^*(\hat{x}) \) and hence \( \lim_{k \to \infty} J^*(\chi(k)) = J^*(\hat{x}) \). The last implies

4 When using the full state \( \chi \) we use the notation \( \chi(k) \) for \( \chi(t(k)) \), as stated in Section 1.1.
Theorem 4.2. Let $X_N \equiv \{k\}$, $T_N = T_{\geq 0}$, and let $q > 0$. Let $\hat{x}$ be an equilibrium with corresponding steady-state input $\hat{u}$, and assume there exists $\hat{q}$ such that $\hat{x}, \hat{t}, \hat{q}, \hat{v}$ satisfies constraints (15c) and (15d) for $\hat{v} = \hat{q}$ and for all $\hat{t} \geq 0$. Let $b$ be independent of $t$, be such that $L(\hat{x}, \hat{t}, \hat{q}, \hat{v}) = 0$, for all $\hat{t} \in T$, and $L(x, t, q, v) \geq \psi (|x - \hat{x}|)$, where $\psi : R_{\geq 0} \rightarrow R_{\geq 0}$ is a nondecreasing function, $\psi (0) = 0$, $\psi (\alpha) \in R_{\geq 0}$, and 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 $x(t+k) = G(x(k), u^w_x(k+1)(0))$, $k \geq 0$, and $\lim_{k \to \infty} x(t) = \hat{x}$.

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 $L(x, t, q, v) \geq \psi (|x - \hat{x}|)$.

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 (iCH, 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


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.
His research interests also include networked control, optimization algorithms, stochastic systems, and distributed algorithms.

A. Bemporad received the master degree in Electrical Engineering in 1993 and the Ph.D. in Control Engineering in 1997 from the University of Florence, Italy. He spent the academic year 1996/97 at the Center for Robotics and Automation, Dept. Systems Science & Mathematics, Washington University, St. Louis, as a visiting researcher. In 1997–1999, he held a postdoctoral position at the Automatic Control Lab, ETH, Zurich, Switzerland, where he collaborated as a senior researcher in 2000–2002. Since 1999 he is with the Faculty of Engineering of the University of Siena, Italy, where he is currently an associate professor. He has published several papers in the areas of hybrid systems, model predictive control, automotive control, multiparametric optimization, computational geometry, and robotics. He is coauthor of the Model Predictive Control Toolbox (The Mathworks, Inc.) and author of the Hybrid Toolbox for Matlab. He was an Associate Editor of the IEEE Transactions on Automatic Control during 2001–2004. He is Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society since 2002.

J. Júlvez received the M.S. and Ph.D. degrees in Computer Science Engineering from the University of Zaragoza in 1998 and 2005. His Ph.D. was related to the study of qualitative and quantitative properties of continuous Petri nets. In 2005 he joined, as a PostDoc researcher, the Department of Software in the Technical University of Catalonia where he spent three years. In 2008 he joined the Department of Computer Science and Systems Engineering in the University of Zaragoza as an assistant professor. His current research is mainly related to the control of hybrid systems, and the performance evaluation and optimization of large discrete event systems.