Event-Based Model Predictive Control and Verification of Integral Continuous-Time Hybrid Automata

Alberto Bemporad[1], Stefano Di Cairano[1⋆] and Jorge Júlvez[2]

[1] Dip. Ingegneria dell'Informazione, Università di Siena, Italy
bemporad,dicairano@dii.unisi.it
[2] Dep. Informática e Ingeniería de Sistemas, Universidad de Zaragoza,
julvez@unizar.es

Abstract. This paper proposes an event-driven model predictive control scheme with guaranteed closed-loop convergence properties for the class of integral continuous-time hybrid automata (icHA). After converting icHA to a corresponding event-driven representation that allows one to compute the model predictive control action by mixed integer programming, sufficient conditions ensuring event-asymptotic and time-asymptotic convergence are proved. The paper also shows how the same modeling methodology can be employed to efficiently solve problems of verification of safety properties.

1 Introduction

Hybrid systems are complex dynamical systems in which continuous and discrete variables coexist and are mutually dependent. The trajectory of a continuous-time hybrid system can be represented as a sequence of continuous evolutions interleaved by discrete events [1, 2], which cause changes in the equations defining the continuous flow, thus changing the operating mode of the system. The continuous flows and the instants at which the discrete events occur are further influenced by exogenous discrete and continuous input signals.

When optimal control is applied to continuous-time hybrid systems [3–5], the resulting computational problem is usually hard to solve, since it involves the solution of non-convex problems [5]. A numerically efficient approach is based on the application of mixed-integer programming (MIP) to a discrete-time representation of the system, in order to solve finite-horizon optimal control problems [6]. A drawback of this technique is that events (such as mode switches) can only occur at sampling instants, which can induce non-negligible modeling errors [7]. Modeling precision can be clearly improved by reducing the chosen sampling period; however, in a model predictive control (MPC) context [8, 9], the obvious disadvantage is that, for a given time-horizon of prediction, a larger number of

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

control variables is involved in the optimization problem. Better model accuracy is paid by increased computation complexity.

A different approach recently proposed in [7] exploits a continuous-time model of the hybrid system, called integral continuous-time hybrid automaton, and abstracts an event-driven representation of it, in which the time is an additional state variable and events, which can occur at any time instant, cause a change of the speed of the continuous states. Moreover, constraints on state and input variables are enforced along the whole continuous-time trajectory, contrarily to discrete-time approaches that do not ensure constraint satisfaction during the inter-sampling period.

Under the modeling assumption that dynamics are piecewise integral ($\dot{x} = B_i u + f_i$) and input functions u are piecewise constant over time, continuous-time optimal control problems over a finite horizon on integral continuous-time hybrid automata can be solved by MIP, by exploiting an event-driven representation of the system [7].

In this paper, after defining the integral continuous-time hybrid automaton in Section 2, we explain in Section 3 how to represent it as an event-driven model that can be exploited for formulating optimal control problems as mixed-integer programs. In Section 4 we discuss an event-driven model predictive control scheme, providing sufficient convergence conditions and presenting a simple numerical example. Finally, in Section 5 we show how the event-driven model can be exploited for verification of hybrid systems, and test the approach on the well-known train-gate benchmark [10].

2 Integral Continuous Hybrid Automaton

In this paper we consider the class of *integral continuous (-time) Hybrid Automata* (icHA) [7]. Such systems are a continuous-time version of the Discrete Hybrid Automaton (DHA) [11], with integral continuous-state dynamics. The icHA has the same structure of the DHA, consisting of the four components reported in Figure 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 possible continuous-time integral dynamics (i.e., the system modes) 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 $i \in \mathcal{I} = \{1, 2, \dots s\}$ is the system mode. While the main reason for focusing the attention to integral dynamics is computational (see [7] and Equation (6) below), the class of continuous-state dynamics (1) has been widely exploited for modeling and verification of hybrid systems [1, 10], showing to be powerful enough for modeling many practical problems[3].

[3] Given a nonlinear (possibly discontinuous) dynamical model $\dot{x} = f(x, u)$, model (1) can be interpreted as a zero-order approximation of the state-transition function

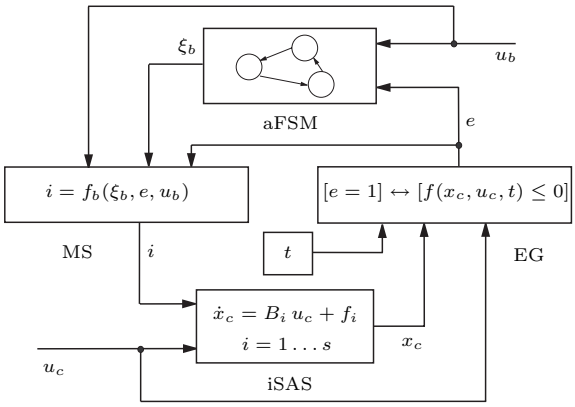


Fig. 1. Integral continuous-time Hybrid Automaton (icHA)

The EG defines the endogenous binary inputs e by linear threshold conditions

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

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

where $n_e^x + n_e^u = n_e$ and $e = [e_1^x \ldots e_{n_e^x}^x \, e_1^u \ldots e_{n_e^u}^x]^T \in \{0,1\}^{n_e}$ is the vector of endogenous binary input variables. The icHA is also excited by exogenous binary input signals $u_b \in \{0,1\}^{m_b}$. We say that an *event* occurs whenever an endogenous input e or an exogenous input (u_c, u_b) changes its value. Accordingly, event instants $t_0 < t_1 < \ldots$ are defined as

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

where we assume that the minimum in (3) exists. As a consequence, the set of admissible input functions is the set $\mathcal{PC}_{(m_c, m_b)}$ of piecewise constant functions $u = \left[\begin{smallmatrix} u_c \\ u_b \end{smallmatrix}\right]$, $u : \mathbb{R} \to \mathbb{R}^{m_c} \times \{0,1\}^{m_b}$ such that $u(t) = u(t_k)$, $\forall t \in [t_k, t_{k+1})$, $\forall k = 0, 1, \ldots$.

The Boolean state $\xi_b \in \{0,1\}^{n_b}$ is defined as $\xi_b(t) \triangleq x_b(t_k)$ for $t_{k-1} \leq t < t_k$ and

$$x_b(t_{k+1}) = f_{\text{aFSM}}(x_b(t_k), u_b(t_k), e(t_k)), \tag{4}$$

where $f_{\text{aFSM}} : \{0,1\}^{n_b + m_b + n_e} \to \{0,1\}^{n_b}$ is a Boolean function. The Boolean state $\xi_b(t)$ remains constant, $\xi_b(t) \equiv x_b(t_k)$, during the whole interval $t_{k-1} < t < t_k$. At the event instant t_k, the Boolean state switches to the new value

with respect to the state vector x and a first-order approximation with respect to the input vector u. Piecewise affine (PWA) models $\dot{x} = A_i x + B_i u + f_i$ are first-order approximations with respect to both x and u.

$f_{\text{aFSM}}(x_b(t_k), e(t_k), u_b(t_k))$, 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 modeled by introducing additional events and states. Note that transitions of icHA can occur at any time instant, not only at multiples of a given sampling period as in DHA [11].

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

$$i(t) = [1\,2\ldots s] \cdot f_{\text{MS}}(\xi_b(t), u_b(t), e(t)), \tag{5}$$

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 $[1\ldots1]\cdot f_{\text{MS}} = 1,\ \forall(\xi_b(t), u_b(t), e(t)) \in \{0,1\}^{n_b+m_b+n_e}$. Note that if the inputs and the e variables are constant, the Boolean state and the system mode are also constant.

3 Event-Driven Representation of icHA

An icHA (1)-(5) can be converted to an event-driven representation that is suitable for computing solutions to optimal control problems. If the system mode $i(t)$ and the input $u_c(t)$ are constant for $t \in [t_k,\ t_{k+1})$, $k = 1,\ldots,h$, the continuous state at t_h is

$$x_c(t_h) \;=\; x_c(t_0) \;+\; \sum_{k=0}^{h-1} \Big(B_{i(t_k)}(t_{k+1} \;-\; t_k)u_c(t_k) \;+\; f_{i(t_k)}(t_{k+1} \;-\; t_k) \Big). \tag{6}$$

Thus, the system dynamics can be rewritten as the 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 events k and $k+1$, $v_c(k) = q(k)u_c(k)$ is the integral over time period $q(k)$ of the input $u_c(k) = u_c(t_k)$, and time t is an additional state variable. The controlled variables are the input integral $v_c(k)$ and the input duration $q(k)$; the input $u_c(k) = \frac{v_c(k)}{q(k)}$ applied to the continuous-time system is computed from them.

The event generator becomes

$$[e_i^x(k) = 1] \leftrightarrow \left[E_i^x \begin{bmatrix} x_c(k) \\ t(k) \end{bmatrix} \leq F_i^x\right],\ i = 1,\ldots n_e^x \tag{8a}$$

$$[e_i^u(k) = 1] \leftrightarrow \left[E_i^u v_c(k) \leq F_i^u q(k)\right],\ i = 1,\ldots n_e^u \tag{8b}$$

where $e(k) = e(t_k)$, and $e(t) = e(t_k)$, $\forall t \in [t_k, t_{k+1})$ by the definition of t_k in (3). Note that 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. The mode selector equation becomes

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

where $i(t) = i(k),\ \forall t \in [t_k, t_{k+1})$ as a consequence of the event definition, and $\tilde{f}_{\rm MS}(x_b(k), u_b(k), e(k)) = f_{\rm MS}(f_{\rm aFSM}(x_b(k), u_b(k), e(t_k)), u_b(k), e(k))$ because of (5) and the definition of $\xi(t)$. Equation (4), is already defined with respect to the events.

Equations (4), (7), (8), (9) define the behavior of the components of the icHA in an event-driven representation. To take into account (3), however, the following condition must be ensured:

$$[(e(t_k), u_c(t_k), u_b(t_k)) = (\bar{e}, \bar{u}_c, \bar{u}_b)] \rightarrow [(e(t), u_c(t), u_b(t)) = (\bar{e}, \bar{u}_c, \bar{u}_b), \forall t \in [t_k, t_{k+1})]. \tag{10}$$

We consider two different cases: (i) the value u_c or u_b changes, so that an event is externally forced, (ii) an endogenous event occurs (e changes its value). The first case is caused by an arbitrary decision (e.g., by a controller), and no additional constraints are needed. Thus, we only need to ensure that

$$[e(t_k) = \bar{e}] \rightarrow [(e(t) = \bar{e}),\ \forall t \in [t_k, t_{k+1})]. \tag{11}$$

Note that the e variables in (2b) can change only when the input changes, thus they can be dealt with as for externally forced events. Hence, we only need to enforce (11) for (2a).

Let the mapping $\mathrm{cod}() : \{0,1\}^{n_e^x} \rightarrow \mathbb{N}$ associate an integer number j to each allowed value of vector $e^x = [e_1^x \dots e_{n_e^x}^x]^T$ defined in (2a). For example j may be the integer whose binary encoding is e^x. Define the matrix $\bar{E}^x(j)$ and the vector $\bar{F}^x(j)$ by collecting the rows in the inequalities of the EG (8) which are satisfied for e^x such that $\mathrm{cod}(e^x) = j$. In addition, define $\hat{E}^x(j)$, $\hat{F}^x(j)$ by collecting as rows the inequalities of the EG (8), which are not satisfied for e^x such that $\mathrm{cod}(e^x) = j$. In this way, for all the values of state and input such that $\mathrm{cod}(e^x(k)) = j$, $\bar{E}^x(j) \left[\begin{smallmatrix} x \\ t \end{smallmatrix}\right] \leq \bar{F}^x(j)$, $\hat{E}^x(j) \left[\begin{smallmatrix} x \\ t \end{smallmatrix}\right] > \hat{F}^x(j)$. As an example, consider two thresholds $[e_1^x = 1] \leftrightarrow [x \leq 0]$, $[e_2^x = 1] \leftrightarrow [x \leq 1]$. The matrices associated to $e^x = [0\ 1]'$, where $\mathrm{cod}(e^x) = 1$, are $\bar{E}^x(1) = 1$, $\bar{F}^x(1) = 1$, collecting the second threshold condition (satisfied), and $\hat{E}^x(1) = 1$, $\hat{F}^x(1) = 0$.

As detailed in [7], in case of integral dynamics, (11) is guaranteed by the mixed-logical constraint

$$[\mathrm{cod}(e^x(t_k)) = j] \rightarrow \left[\begin{bmatrix} \bar{E}^x(j) \\ -\hat{E}^x(j) \end{bmatrix} \begin{bmatrix} x(t_{k+1}) \\ t(k+1) \end{bmatrix} \leq \begin{bmatrix} \bar{F}^x(j) \\ -\hat{F}^x(j) \end{bmatrix} + \varepsilon \mathbf{1}\right], \tag{12}$$

in which ε is an arbitrary small positive constant that ensures that $e(t) = e(t_k)$, $\forall t \in [t_k, t_{k+1} - \sigma(\varepsilon)]$, and $\sigma(\varepsilon)$ tends to zero as ε tends to zero. Note that $x(t_{k+1})$ is a linear function of $x(k)$, $q(k)$, and $v(k) = \left[\begin{smallmatrix} v_c(k) \\ v_b(k) \end{smallmatrix}\right]$, where $v_b(k) = u_b(k)$, so that (12) is reformulated as mixed-integer inequalities on $x(k)$, $q(k)$, $v(k)$, $e(k)$.

Equations (4), (7), (8), (9), (12) represent a DHA that can be modeled in HYSDEL [11] through which we can obtain an event-driven MLD (eMLD) system

$$x(k+1) = Ax(k) + B_1 w(k) + B_2 e(k) + B_3 z(k) + B_5, \tag{13a}$$

$$t(k+1) = t(k) + q(k), \tag{13b}$$

$$E_2 e(k) + E_3 z(k) \leq E_1 w(k) + E_4 x(k) + E_5. \tag{13c}$$

where $w(k) = \begin{bmatrix} v(k) \\ q(k) \end{bmatrix}$. Differently from the standard discrete-time MLD system [6], in (13) k is an event counter.

Remark 1. Discontinuities of the continuous state trajectory can be introduced by resets. To model resets, additional *reset modes* $i \in \{s+1, \ldots, s_r\}$ are included, (7a) is modified into $x_c(k+1) = (E_i x_c(k) + h_i) + B_{i(k)} v(k) + f_{i(k)} q(k)$, and (7b) into $t(k+1) = t(k) + G_i q(k)$. In modes $i = \{1 \ldots s\}$, $E_i = I$ (where I is the identity matrix), $h_i = 0$ and $G_i = 1$, while in reset modes $i = \{s+1 \ldots s_r\}$ $B_i = 0$, $f_i = 0$ and $G_i = 0$. Note that resets are instantaneous.

The definition of the event-driven dynamics of the icHA by an eMLD system allows the definition of finite horizon optimal control problems that can be solved by mixed-integer programming (MIP) as shown in [6]. With respect to MLD models, the only difference is that the horizon represents the number of events occurred, and the time elapsed along the horizon is a continuous state variable.

4 Event-Driven Model Predictive Control

In [7] event-driven open-loop optimal control strategies are proposed with different cost functions: minimum-time, minimum-effort, and minimum displacement. They are computationally less expensive than their discrete-time counterparts and the system's constraints are satisfied along the whole trajectory instead of only at sampling instants. However, the approach of [7] is an open-loop control strategy. We introduce an event-driven MPC closed-loop strategy here.

Given an icHA, the eMLD model is obtained as explained in Section 3 so that a finite horizon optimal control problem can be formulated as in [6]

$$\min_{q,v} \quad J(x,t,v,q) \tag{14a}$$

$$\text{s.t.} \quad \text{system dynamics (13)} \tag{14b}$$

$$g(x,t,q,v) \leq 0\,, \tag{14c}$$

$$x(0) = x_0, \quad t(0) = t_0, \tag{14d}$$

where $t = \{t(k)\}_{k=0}^{N}$ are the event instants, $x = \{x(k)\}_{k=0}^{N}$ are the corresponding state values, $q = \{q(k)\}_{k=0}^{N-1}$ are the durations of the time intervals between two consecutive events and $v = \{v(k)\}_{k=0}^{N-1}$ are the input integrals during $[t_k, t_{k+1})$. We consider cost functions of the form

$$J(x,t,v,q) = F(x(N)) + \sum_{k=0}^{N-1} L(x(k), t(k), v_c(k), q(k)) \tag{15}$$

where $L(x(k), t(k), v_c(k), q(k)) = \|x(k) - \hat{x}\|_p^{Q_1} + \|t(k) - \hat{t}\|_p^{Q_2} + \|v(k) - \hat{v}\|_p^{R_1} + \|q(k) - \hat{q}\|_p^{R_2}$ is the stage cost, $F(x(N)) = \|x(N) - \hat{x}\|_p^{Q_N}$ is the terminal cost, $p = 1, 2, \infty$, $\|z\|_\infty^Q = \max_i |(Qz)_i|$, $\|z\|_1^Q = \sum_i |(Qz)_i|$ and $\|z\|_2^Q = z^T Q z$, and if not differently stated $\hat{q} = 0$, $\hat{v} = 0$. In (15), N is the number of allowed events, and

as a consequence, the time period considered in the optimization problem will depend on the chosen input profile through the system dynamics: For a fixed N, when the continuous state evolves quickly and switches are frequent, the resulting time horizon will shrink because the system requires a tighter control action; on the contrary, when the dynamics is slow and few mode switches occur, the time-horizon will increase without increasing the complexity of the optimization problem, so that a smaller amount of computation per time unit is required.

Constraint (14c) represents additional constraints in the optimal control problem that have different purposes. Bounds on the continuous-time input value $\underline{u} \leq u_c(t) \leq \overline{u}$ can be cast as the linear constraints

$$\underline{u}q(k) \leq v(k) \leq \overline{u}q(k). \tag{16}$$

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

$$\underline{q} \leq q(k) \leq \overline{q}. \tag{17}$$

A finite value of $\overline{q}$ imposes a maximum time for each control action, in order to prevent the system from running in open-loop with a constant input for too long, because of the receding horizon mechanism. A minimum duration $\underline{q}$ ensures a minimum time interval between two events (thus, between two mode switches), therefore avoiding undesirable effects such as high frequency chattering and Zeno behaviors. Additional constraints in (14c) may concern terminal constraints on the final state and on the final time of the optimization problem. In this case, we consider

$$x(N) \in \mathcal{X}_T, \qquad t(N) \in \mathcal{T}_T, \tag{18}$$

as terminal constraints, where $\mathcal{X}_T$, $\mathcal{T}_T$ can be either polyhedra or isolated points.

The event-driven Model Predictive Control (eMPC) strategy is defined as follows:

1. Let N be the event horizon, and consider the initial instant $\tilde{t}$ and the corresponding state value $x(\tilde{t})$.
2. Solve the optimal control problem (14) with $t_0 = \tilde{t}$ and $x_0 = x(\tilde{t})$ and let $[v^*(0), \ldots, v^*(N-1)]$ be the sequence of optimal input integral values, $[q^*(0), \ldots, q^*(N-1)]$ be the sequence of input action durations, $[x^*(1), \ldots, x^*(N)]$ be the predicted state values at event instants and $[t^*(0), \ldots, t^*(N)]$ be the corresponding time instants at which the events occur.
3. Compute the input value $u_c(\tilde{t}) = \frac{v_c^*(0)}{q^*(0)}$, and apply $u(t) \equiv \begin{bmatrix} u_c(\tilde{t}) \\ v_b^*(0) \end{bmatrix}$ to the icHA during the time interval $[\tilde{t}, \tilde{t} + q(0)]$. [4]

[4] Different strategies may be proposed here, for example apply $u_c(t) \equiv \frac{v_c^*(0)}{q^*(0)}$ for $t \in [\tilde{t}, \tilde{t} + \min\{q(0), T_s\}]$, where T_s is a given maximum time interval the system can be run in open-loop, or apply the optimal input trajectory in open-loop for a fixed time interval (possibly covering more than one optimal event instants) to prevent out-of-time computation problems due to an excessively small duration $q^*(0)$.

4. Set $\tilde{t} \leftarrow \tilde{t} + q(0)$, $x(\tilde{t}) \leftarrow \breve{x} = x(\tilde{t} + q(0))$ and go to 2.

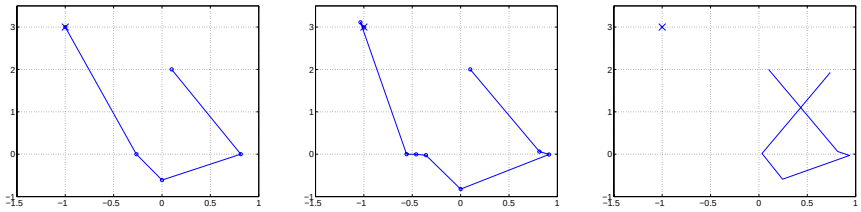
Note that the actual state $\breve{x}$ at the end of each control action can be different from the predicted one $x^*(1)$, because of external disturbances and modeling errors. Clearly, the main advantage of the eMPC strategy with respect to open-loop optimal control [7] is its closed-loop nature, since after each predicted event the real state is read or estimated again and a new optimal input sequence is computed from it. In the current event-driven approach also the prediction of the time instants at which events occur can be updated.

4.1 eMPC Example

In this section we present a simple numerical example showing the behavior of the eMPC strategy and its robustness with respect to disturbances. We consider a system having two continuous states x_1 and x_2, and two state thresholds $[e_1^x = 1] \leftrightarrow [x_1 \leq 0]$, $[e_2^x = 1] \leftrightarrow [x_2 \leq 0]$, 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 Equation (1) for $i = 1, \ldots, 4$ are $f_1 = f_4 = \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right]$, $f_2 = f_3 = \left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right]$, $B_1 = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1.4 \end{smallmatrix}\right]$, $B_2 = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1.5 \end{smallmatrix}\right]$, $B_3 = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1.15 \end{smallmatrix}\right]$, $B_4 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 2.3 \end{smallmatrix}\right]$. Moreover, there are additional constraints on the inputs: when in mode $i = 1$ it must hold that $u_2 \geq -2$, for $i = 2$ $u_2 \leq -0.5$, for $i = 3$ $u_2 \leq 2$, and for $i = 4$ $u_1 \leq 2$ and $-0.5 \leq u_2 \leq 2$.

We want to bring the state of the system from $x_0 = \left[\begin{smallmatrix} 0.1 \\ 2 \end{smallmatrix}\right]$ to $x_f = \left[\begin{smallmatrix} -1 \\ 3 \end{smallmatrix}\right]$ while minimizing function (14a), where $p = \infty$, $Q_1 = \left[\begin{smallmatrix} 10 & 0 \\ 0 & 10 \end{smallmatrix}\right]$, $Q_2 = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$, $R_1 = \left[\begin{smallmatrix} 10^{-3} & 10^{-3} \end{smallmatrix}\right]$, $R_2 = 1$, $\hat{x} = x_f$ 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 system is perturbed by input-additive disturbances, so that the continuous state dynamics is $\dot{x}(t) = B_{i(t_k)}\big(u(t_k) + \xi_k\big) + f_{i(t_k)}$, $\forall t \in [t_k, t_{k+1})$, where ξ_k is a sequence of time-uncorrelated stochastic vectors in which each component is independent from the other and uniformly distributed in $[-0.1, 0.1]$.

Figure 2 reports the continuous-time trajectories generated by the eMPC controller with a prediction horizon of 4 events applied for 8 steps. In the undisturbed case (Figure 2(a)) 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 (Figure 2(b)) the eMPC is able to counteract them, and to still bring the system close to x_f, 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 reported in Figure 2(c), showing that the effects of the interaction of the disturbance with the switching nature of the system are not negligible.



(a) eMPC, undisturbed. (b) eMPC, disturbed. (c) Open-loop, disturbed.

Fig. 2. Example, controlled system trajectory

4.2 Conditions for Convergence of eMPC

We consider the case in which the terminal sets $\mathcal{X}_T$, $\mathcal{T}_T$ are isolated points or polytopes separately.

Definition 1. *A state value* $\bar{x} = \left[\begin{smallmatrix} \bar{x}_c \\ \bar{x}_b \end{smallmatrix}\right]$ *is an equilibrium point 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* $\bar{e}_\infty$ *such that:*

1. $\bar{e}_\infty = f_{EG}(\bar{x}_c, \bar{u}_{c,\infty}, t)$, $\forall t \geq 0$, *where* f_{EG} *is the event generator (2);*
2. $\bar{x}_b = f_{aFSM}(\bar{x}_b, \bar{u}_{b,\infty}, \bar{e}_\infty)$*;*
3. $\bar{\imath} = [0, 1, \dots s] \cdot f_{MS}(\bar{x}_b, \bar{u}_{b,\infty}, \bar{e}_\infty)$*;*
4. $B_{\bar{\imath}}\, \bar{u}_{c,\infty} + f_{\bar{\imath}} = 0$.

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

Terminal equality constraint. If the terminal set reduces to a point, Constraint (18) can be written as

$$x(N) = \hat{x}, \qquad t(N) = \hat{t}\,, \tag{19}$$

where $\hat{x}$ and $\hat{t}$ are referred to as target state and target time, respectively. As a consequence, the terminal cost can be removed from (15). In the following we denote by $\chi(k) = \left[\begin{smallmatrix} x(k) \\ t(k) \end{smallmatrix}\right]$ the state of the eMLD system.

Consider an initial state $\chi_0 = \chi(k)$ and solve Problem (14), obtaining the optimal cost $J^*(\chi_0)$, the optimal state trajectory $X^*(\chi_0) = [\chi_1^*(\chi_0) \dots \chi_N^*(\chi_0)]$ and the optimal input $\mathbf{w}^*(\chi_0) = [w_0^*(\chi_0), \dots w_{N-1}^*(\chi_0)]$, where $w_i = \left[\begin{smallmatrix} v_i \\ q_i \end{smallmatrix}\right]$. Let the eMPC control action at step k be $w_{MPC}(k) = w_0^*(\chi_0)$ and let the initial state for the next optimization problem be $\chi_1 = \chi(k+1) = G(\chi_0, w_{MPC}(k))$, where $G(\chi(k), w(k))$ is the state update function (13).

Theorem 1. *Let $\underline{q} = 0$ in (17), $\hat{x}$ be an equilibrium point with steady-state input $\bar{u}_\infty$, $\hat{q} = 0$, $\hat{v} = \left[\begin{smallmatrix} 0 \\ \bar{u}_{b,\infty} \end{smallmatrix}\right]$, and Q_1, Q_2, R_1, R_2 full rank. If Problem (14) is feasible for $x_0 = x(k)$, $t_0 = t(k)$, then it is feasible for $x_0 = x_1^*(x(k))$, $t_0 = t_1^*(x(k))$ and the state and time converge to the target state $\hat{x}$ and target time $\hat{t}$, respectively, as the number of events tends to infinity.*

Proof. Let $\chi_1 = \chi_1^*(\chi(k))$. The input sequence $\tilde{\mathbf{w}}(\chi_1) = \left[w_1^*(\chi_0), \dots w_{N-1}^*(\chi_0), \left[\begin{smallmatrix} 0 \\ \bar{u}_{b,\infty} \\ 0 \end{smallmatrix}\right]\right]$ obtained by shifting $\mathbf{w}^*(\chi_0)$ to the left is feasible for Problem (14), when χ_0 is replaced by χ_1. Then $\chi_i(\chi_1) = \chi_{i+1}(\chi_0)$ for $i = 0, \dots, N-1$ and $\chi_N(\chi_1) = G\left(\chi_{N-1}^*(\chi_1), \left[\begin{smallmatrix} 0 \\ \bar{u}_{b,\infty} \\ 0 \end{smallmatrix}\right]\right)$ is equal to $\chi_{N-1}(\chi_1) = \chi_N(\chi_0)$. Thus, the dynamics and the operating constraints are satisfied at $\chi_N(\chi_1)$ and the sequence $\tilde{\mathbf{w}}(\chi_1)$ satisfies the constraints in (14).

Next, we show that the sequence of cost values is decreasing by applying the same approach of [6]. Because of optimality, $J^*(\chi_1) \leq J(\chi_1, \tilde{\mathbf{w}}(\chi_1))$, where

$$J(\chi_1, \tilde{\mathbf{w}}(\chi_1)) = J^*(\chi_0) - L(x(0), t(0), v(0), q(0)), \tag{20}$$

and hence $J^*(\chi_1) \leq J^*(\chi_0)$. Since $J(\chi(k))$ is lower bounded by 0 and the sequence is not increasing, $\lim_{k\to\infty} J(\chi(k)) = J_\infty$, so that $\lim_{k\to\infty} J(\chi(k+1)) - J(\chi(k)) = 0$, implying that $\lim_{k\to\infty} x(k) = \hat{x}$, $\lim_{k\to\infty} v(k) = \hat{v}$, $\lim_{k\to\infty} q(k) = 0$, $\lim_{k\to\infty} t(k) = \hat{t}$. □

Note that convergence is asymptotic with respect to the number of events, but nonetheless the state converges to the target state $\hat{x}$ in the finite time $\hat{t}$. In the more common case of $t(N)$ unconstrained and $Q_2 = 0$, $\lim_{k\to\infty} x(k) = \hat{x}$ but it is possible that $\lim_{k\to\infty} t(k) = \infty$, thus having time-asymptotic convergence; the proof follows directly from the previous one.

Remark 2. When q_0^* is very small the time required for solving the next optimization problem may be insufficient An approach to avoid $q(k) \to 0$ is to set $\hat{q} = q_\infty > 0$, $Q_2 = 0$, $\hat{v} = 0$ and $R_1 = 0$. In this way, if a steady state input $\bar{u}_\infty$ exists, eventually unknown, then $w_{MPC}(k) = \left[\begin{smallmatrix} \bar{u}_{c,\infty} q_\infty \\ \bar{u}_{b,\infty} \\ q_\infty \end{smallmatrix}\right]$ when $x(k) = \hat{x}$, which has zero cost. It must be noted that solutions in which $q(k) = 0$ are still feasible, but not optimal.

Next, we consider the case $\underline{q} > 0$ in (17), that ensures a minimum dwell time.

Theorem 2. *Let $\bar{u}_\infty$ be the steady-state input corresponding to the equilibrium point $\hat{x}$, let $\hat{v} = \left[\begin{smallmatrix} \bar{u}_{c,\infty} \hat{q} \\ \bar{u}_{b,\infty} \end{smallmatrix}\right]$ and $\underline{q} \leq \hat{q} \leq \overline{q}$. Let $Q_2 = 0$ and $t(N)$ be unconstrained. If Problem (14) is feasible for $\chi_0 = \chi(k)$, it is also feasible for $\chi(0) = \chi_1 = G(\chi_0, w_{MPC}(k))$ and the state converges to $\hat{x}$.*

Proof. Let $\mathbf{w}^*(\chi_0)$ be the optimal input sequence of the problem with initial state χ_0. Then $\tilde{\mathbf{w}}(\chi_1) = [w_1^*(\chi_0), \dots w_{N-1}^*(\chi_0), \left[\begin{smallmatrix} \hat{v} \\ \hat{q} \end{smallmatrix}\right]]$ is feasible since $x(N+1) = x(N) = \hat{x}$, while fulfilling also all the other constraints. Furthermore, (20) holds and convergence is ensured. □

Note that the eMPC controller in the example of Section 4.1 was designed basing on the hypotheses of Theorem 2.

Remark 3. When the constraint $q \geq \underline{q} > 0$ is added, the optimal control problem might become unfeasible. A sufficient condition for feasibility is that $\forall i$, $\exists \bar{u}_i$ that satisfies the constraints of mode i and verifies $B_i\,\bar{u}_i + f_i = 0$. Such condition ensures the existence of an input that blocks the system state in each mode.

Terminal cost and terminal set. The terminal constraints are defined by

$$Sx(N) \leq M, \quad S_T t(N) \leq M_T, \tag{21}$$

where S is a suitable matrix and M, S_T, M_T are suitable vectors. For the sake of simplicity, we discuss the case in which the target time is not constrained nor weighted (thus S_T, M_T are empty and $Q_2 = 0$), $\hat{x} = 0$ and $\hat{v} = 0$, and the icHA system is time invariant (i.e. conditions in (2) do not depend on t), so that we can disregard the eMLD additional state t (the extensions are straightforward). We assume that there are no Boolean inputs, and that in a neighborhood of the origin the mode i is such that $f_i = 0$ in (1) and the discrete state is constantly $x_b = [0 \dots 0]^T$. The last two conditions ensure that the translation of the eMLD yields an equivalent piecewise affine (PWA) model [12] that is linear in a neighborhood of the origin. We use here the results of [13] for convergence of MPC in discrete-time.

Let $\mathcal{X}_T$ be the polytope $\{x : Sx \leq M\}$, $W(x) = \{V(x) \times Q\}$ be the set of feasible solutions $w_0^*(x) = \begin{bmatrix} v_0^*(x) \\ q_0^*(x) \end{bmatrix}$ to Problem (14) when $x_0 = x$, and consider an auxiliary state-feedback controller

$$\tilde{w}(k) = h\big(x(k)\big). \tag{22}$$

The results on [13] ensure that if (i) $h\big(x(k)\big) \in W(x)$ $\forall x \in \mathcal{X}_T$, (ii) $\mathcal{X}_T$ is a positively invariant set for system (13) in closed loop with (22), and (iii) the inequality

$$F(G(x(k), h(x(k))) - F(x(k)) + L(x(k), h(x(k))) \leq 0, \tag{23}$$

is satisfied, then if Problem (14) is feasible at step k, it is feasible at step $k+1$ and the state converges asymptotically to the target state.

The problem reduces to computing the auxiliary controller, that for the event-driven approach of this paper has the structure

$$w(x) = \begin{bmatrix} v(x) \\ q(x) \end{bmatrix} = \begin{bmatrix} f_1(x(k)) \\ f_2(x(k)) \end{bmatrix}. \tag{24}$$

Consider the discrete-time system Σ_d with sampling time T_s described by equations (13a), (13c) in which $q(k) = T_s$ and the index k represents the sampling step counter. Let $J_d(x, v) = F_d(x(N)) + \sum_{k=0}^{N-1} L_d(x(k), v(k))$ be the cost function, where $L_d(x(k), v_c(k)) = \|x_c(k) - \hat{x}\|_p^{Q_1} + \|v_c(k) - \hat{v}\|_p^{R_1}$, $F_d(x(N)) =$

$F(x(N))$, and $h_d(x(k))$ be an auxiliary piecewise linear (PWL) state-feedback controller. The decreasing cost condition of [13] for asymptotic stability is

$$F_d(G_d(x(k), h_d(x(k))) - F_d(x(k)) + L_d(x(k), h_d(x(k))) \leq 0. \tag{25}$$

The following proposition shows that the auxiliary controller for Σ_d allows proving convergence of the event-driven system.

Proposition 1. *Let T_s be such that $\underline{q} \leq T_s \leq \overline{q}$, $\hat{q} = T_s$, $Q_2 = 0$ and $h_d\big(x(k)\big)$ be a discrete-time PWL controller with sampling time T_s, such that $h_d(x) \in V(x)$ $\forall x \in \mathcal{X}_T$. Let $\mathcal{X}_T$ be a positively invariant set for system Σ_d in closed-loop with $h_d(x)$, and (25) be satisfied. Then $x(k) \to 0$ for $k \to \infty$.*

Proof. System (13a), (13c), when $q(k) = T_s$ is a discrete-time MLD system of the form $x((k+1)T_s) = G_d(x(kT_s), v(kT_s))$ for which an equivalent PWA system can be computed [12]. Since we have supposed that in a neighborhood of the origin the continuous-time system has no affine terms and that the Boolean state is $[0, \ldots, 0]$, also the discrete-time PWA system is linear in such a region and the results of [13] hold. The discrete-time controller is equivalent to an event-driven controller that raises an event every T_s time units. Then the controller $w(k) = h(x(k)) = \begin{bmatrix} h_d(x(k)) \\ T_s \end{bmatrix}$ is an auxiliary event-driven controller for system (13) that respects condition (23), since (23) is equal to (25) because of the chosen cost function ($\hat{q} = T_s$, $Q_2 = 0$).

Thus $h(x(k))$ is a state-feedback controller that respect hypotheses of [13] for system (13) interpreted as discrete-time systems, proving convergence of $x(k) \to 0$ as $k \to \infty$. □

Proposition 1 ensures that if a discrete-time PWL controller respecting the hypotheses of [13] exists, for instance computed as in [14], then the eMPC controller is converging. The sampling time of the auxiliary controller is used to compute a valid $\mathcal{X}_T$ and can be changed in the design phase, without changing anything in the event-driven system but the parameter $\hat{q}$. In order to relax the assumption $x_b = [0, \ldots, 0]^T$, one may require convergence only for the continuous state as in [15], thus without weighting x_b in the cost function.

5 Event-Based Verification of icHA

In Section 4 we have exploited the icHA and its discrete-event reformulation for MPC design. However, this model can be conveniently exploited also for verification of safety and liveness properties. The main advantage of the event-based approach is that verification queries, whenever their negation can be formulated as a combination of linear and logical constraints, can be posed as feasibility

problems of mixed-integer programs

$$\begin{aligned} \min_{q,v} \quad & 0 \\ \text{s.t.} \quad & \text{system dynamics (13)} && \text{(26a)} \\ & g(x,t,q,v) \leq 0\,, && \text{(26b)} \\ & x(0) \in \mathcal{X}_0, \quad t(0) \in \mathcal{T}_0. && \text{(26c)} \\ & H(x(N),t(N)) \leq 0 && \text{(26d)} \end{aligned}$$

where (26a) and (26b) are the same as in (14), (26c) defines the set of possible initial states and (26d) is the region in which the query to be verified is false and it is enforced on the system's final state. Note that since we are considering mixed integer programming, $H()$ can be any combination of linear and logical constraints and $\mathcal{X}_0$, $\mathcal{T}_0$ can be any union of polyhedra. The event-horizon on which the property is verified is defined by the constant N. If Problem (26) admits a feasible solution, then there exists a trajectory departing from a valid initial state that violates the query to be verified, thus, the query is false. Note that if $\underline{q} = 0$, (26d) ensures safety $\forall k = 1, \ldots, N$, even if it is formulated only with respect to the N^{th} step. In fact, if a feasible solution to Problem (26) for a horizon $k < N$ exists, then a feasible solution of (26) also exists, by extending the solution on k steps by "fictitious" events separated by $q = 0$ time units. Thus, the infeasibility implies that at any event instant constraint (26d) is unsatisfied. This implies also the safety of the whole trajectory, since trajectories are piecewise-linear because of the integral dynamics [7]. An intuitive explanation of this property is the following. In order to reach an interior point of the unsafe region, one of the thresholds delimiting such a region must be crossed. However, every time a threshold is crossed an event occurs and the state at such instant is inside the new region. If no state values at the event instants reside in the unsafe region, then the thresholds delimiting such a region cannot have been crossed. Note that if this approach is applied to standard discrete-time models, the infeasibility of the mixed-integer program would only ensure safety at sampling instants.

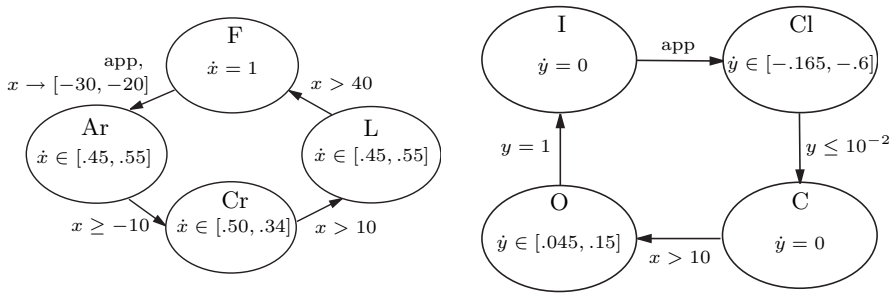
It is easy to recognize similarities between the icHA and the Linear Hybrid Automaton (LHA) [1, 10], a model which has been widely exploited for verification of hybrid systems [10]. The LHA considers discrete and continuous states, the continuous dynamics are defined by discrete state dependent differential inclusions in the form $\sum_i a_i^j \dot{x}_i \in [b^j, c^j]$, where j is the discrete state index, i is the continuous state index, a_i^j, b^j, c^j are constants and x_i are the continuous state variables. The discrete states have associated invariant sets, defined by linear constraints over continuous state variables, the discrete state transitions are enabled by linear conditions over continuous state variables and after each of them the continuous state can be reset. The discrete state dynamics are defined by an aFSM with resets in both models, and the equations of the continuous dynamics switch according to the discrete state. For any given discrete state, all admissible continuous state trajectories of an LHA can be produced by an icHA by a proper selection of the input functions $u \in \mathcal{PC}$ and, viceversa, all icHA trajectories can

be generated by an LHA by appropriately choosing the ranges of the differential inclusion. For instance, the dynamics $a \leq \dot{x}(t) + \dot{y}(t) \leq b$ can be modeled as $\dot{x} = u_1(t)$, $\dot{y} = u_2(t)$ along with $a \leq u_1(t) + u_2(t) \leq b$. The discrete state transitions of icHA are deterministic, those of LHA are not. However, in an icHA the non-determinism can be modeled by adding external signals $\eta(k)$[5] in (13). For instance, a transition of an LHA that can be fired whenever $a \leq x(t) \leq b$ can be modeled by adding the input $-\frac{b-a}{2} \leq \eta(k) \leq \frac{b-a}{2}$ and by setting the transition to occur when $x(k) + \eta(k) = \frac{b+a}{2}$.

The practical consequence of the similarities between LHA and icHA is that many systems that are modeled as LHA can be modeled also as icHA and verified by solving problem (26) by mixed integer programming for which efficient algorithms and tools exist. A formal proof of equivalence between subclasses of LHA and of icHA is beyond the scope of this paper.

5.1 Verification Example

Consider the "train-gate" system [10], with small modifications. The system consists of a train that must safely cross a gate, meaning that when the train is crossing the gate, this must be closed. The gate can be *idle (I)*, *closing (Cl)*, *closed (C)* or *opening (O)*. A train can be *arriving (Ar)*, *crossing (Cr)*, *leaving (L)* or *far (F)*, depending on its position with respect to the gate. The corresponding automata with continuous-time differential inclusions are reported in Figure 3, where x is the train position and y is the gate position. Note that the signal *app* forces a transition in which x is reset. We performed the tests on a Pentium IV-M 2 GHz, equipped with 1 GB Ram, running MATLAB 6.5 and CPLEX 9.0.



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

Fig. 3. Train-Gate system

The system is modeled as an icHA and converted to eMLD form. Let the initial state be (x_0, y_0), where $x_0 \in [-25, -20]$ and $y_0 = 1$, and (Ar,Cl) as discrete

[5] $\eta(k)$ is added in (13) as an additional component of $v(k)$.

state, and the query be: *"Does the system always stay out of the unsafe state* (Cr, Cl)*?"*. Problem (26) is solved for $N = 6$ proving its infeasibility, meaning that an unsafe trajectory does not exist. The computation required 0.984 seconds. If the differential inclusion in state Cl is changed to $\dot{y} = [-0.145, -0.4]$ a solution is found, meaning that the gate is closing too slowly.

Another query that can be verified is the following: *"Does the train always reach the state* F *in less than* 100 *time units, when departing from* x_0*?"*. The answer is no, since there exists a feasible solution to problem (26) in which (26d) is $x \leq 40$ and $-t \leq -100$. This query was tested in 0.312 seconds. Differently from the previous one, this query involves the capability of the system to reach its objective, thus it is related to the system liveness.

6 Conclusions

In this paper we have shown how to obtain an event-driven representation of an integral continuous-time hybrid automaton and we have analyzed model predictive control and verification schemes for such systems. The main advantage is that a continuous-time hybrid system can be analyzed as a discretely evolving one, so that MIP techniques can be exploited for computing the eMPC control action and for verification of safety properties. In addition, a lighter computational burden may be result with respect to the discrete-time approach.

References

1. Henzinger, T.A.: The theory of hybrid automata. In: Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey (1996) 278–292
2. Lygeros, J., Johansson, K.H., Simic, S.N., Zhang, J., Sastry, S.: Dynamical properties of hybrid automata. IEEE Tr. Automatic Control **48** (2003) 2–17
3. Gokbayrak, K., Cassandras, C.: Hybrid controllers for hierarchically decomposed systems. In Krogh, B., Lynch, N., eds.: Hybrid Systems: Computation and Control. Springer-Verlag (2000) 117–129
4. Shaikh, M.S., Caines, P.E.: On the optimal control of hybrid systems: Optimization of trajectories, switching times, and location schedules. In: Hybrid Systems: Computation and Control, Springer-Verlag (2003) 466–481
5. Xu, X., Antsaklis, P.J.: Results and perspectives on computational methods for optimal control of switched systems. In: Hybrid Systems: Computation and Control, Springer-Verlag (2003) 540–555
6. Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints. Automatica **35** (1999) 407–427
7. Bemporad, A., Di Cairano, S., Júlvez, J.: Event-driven optimal control of integral continuous-time hybrid automata. In: Proc. 44th IEEE Conf. on Decision and Control, Seville, Spain (2005) To Appear.
8. Maciejowski, J.: Predictive control with constraints. Englewood Cliffs, NJ: Prentice Hall. (2002)
9. Qin, S., Badgwell, T.: A survey of industrial model predictive control technology. Control Engineering Practice **11** (2003) 733–764

10. Henzinger, T.A., Ho, P.H., Wong-Toi, H.: HyTech: A model checker for hybrid systems. Int. J. on Software Tools for Technology Transfer **1** (1997) 110–122
11. Torrisi, F.D., Bemporad, A.: HYSDEL — A tool for generating computational hybrid models. IEEE Tr. Contr. Systems Technology **12** (2004) 235–249
12. Bemporad, A.: Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. IEEE Tr. Automatic Control **49** (2004) 832–838
13. Lazar, M., Heemels, W., Weiland, S., Bemporad, A.: Stability of hybrid model predictive control. In: Proc. of Int. Workshop on Assessment and Future Directions of NMPC, Germany (2005)
14. Lazar, M., Heemels, W., Weiland, S., Bemporad, A.: Stability of hybrid model predictive control. In: Proc. 43th IEEE Conf. on Decision and Control, Paradise Island, Bahamas (2004) 4595–4560
15. Ferrari-Trecate, G., Cuzzola, F.A., Morari, M.: Lagrange stability and performance analysis of discrete-time piecewise affine systems with logic states. Int. J. Control **76** (2003) 1585–1598