Sampled-Data Event Control of Hybrid Systems for Control Specifications Given by Predicates

Yoshiyuki Tsuchie and Toshimitsu Ushio
Graduate school of Engineering Science,
Osaka University, Toyonaka,
Osaka 560-8531, Japan,
yoshiyuki@hopf.sys.es.osaka-u.ac.jp, ushio@sys.es.osaka-u.ac.jp.

Abstract. We consider a hybrid system controlled by a sampled-data controller whose action is periodically time-driven, that is, the control inputs can change only at the particular time instants. We introduce a transition system as semantics of the controlled hybrid system and consider a control specification given by a predicate. First, we derive a necessary and sufficient condition for the predicate to be control-invariant and show that there always exists the supremal control-invariant sub-predicate for any predicate. Finally, we propose a procedure to compute it.

1 Introduction

In a direct method for a design of a sampled-data controller, a sampled-data controlled system is described as a model with continuous-time variables (a plant) and discrete-time variables (a digital controller). So a hybrid system is a suitable continuous-time model for the direct method [1].

Silva and Krogh proposed an extension of a hybrid automaton with time-driven events to model explicitly discrete transitions that are based on time-driven sampling of the continuous state and define a transition system as semantics to verify its dynamics[2, 3]. Tsuchie and Ushio discussed the state feedback control of a hybrid automaton with time-driven events. However, the controller is designed in continuous-time setting. In this paper, we discuss a sampled-data event controller and consider a control specification given by a predicate on the state set of the controlled hybrid system, where the sampled-data event controller assigns a set of control-enabled events, called a control pattern, based on the state of the hybrid system and updates it at each sampling time so that all reachable states of the closed-loop system satisfy the predicate.

We use a labeled transition system $T = (Q, Act, \mathcal{T}, Q_0)$ in order to define semantics of controlled hybrid systems, where $Q$ is a states set, $Act$ is a label set, $\mathcal{T} \subseteq Q \times Act \times Q$ is a state transition relation, and $Q_0 \subseteq Q$ is the initial state set. Let $\mathcal{P}(Q)$ be the set of all predicates on $Q$. A partial order “$\leq$” for $\mathcal{P}(Q)$ is defined as follows: for $P_1, P_2 \in \mathcal{P}(Q)$, $P_1 \leq P_2 \iff P_1(q) \leq P_2(q) \forall q \in Q$. For
each $a \in Act$, we define two predicates as follows[4]:

\[
D_a(\mathcal{T})(q) = \begin{cases} 
1 & \text{if } a \in \{\bar{a} \in Act| \exists q' \in Q \text{ s.t. } (q, \bar{a}, q') \in \mathcal{T}\}, \\
0 & \text{otherwise},
\end{cases}
\]

\[
wpl_a(P, \mathcal{T})(q) = \begin{cases} 
1 & \text{if } P(q') = \{\forall q' \in \{q, a, \bar{q}' \in \mathcal{T}\}, \\
0 & \text{otherwise}.
\end{cases}
\]

For a subset $A \subseteq Act$, we define $wp_A(P, \mathcal{T}) = \bigvee_{a \in A} wp_a(P, \mathcal{T})$.

2 Controlled Hybrid Automaton

We consider a plant modeled by a hybrid automaton $H=(V, E, \Sigma, inv, init, flow, jump)$, where $V$ and $\Sigma$ are the set of nodes and events, $E \subseteq V \times \Sigma \times V$ is the set of edges, that is, $(e(v, \sigma, v')) \in E$ is an edge $e$ from $v$ to $v'$ labeled by $\sigma$ and corresponds to a discrete transition by the occurrence of $\sigma$, $inv(v) \subseteq \mathbb{R}^n$ is the set of values which the continuous state can take in $v$, $flow(v) \subseteq \mathbb{R}^n \times \mathbb{R}^n$ is a set of values which $(x, \dot{x})$ can take in $v$, $init(v)$ is the set of all possible initial continuous states in $v$, for each $e(v, \sigma, v') \in E$, $jump(e) \subseteq 2^{\mathbb{R}^n \times \mathbb{R}^n}$ is the jump relation, that is, $(x, x') \in jump(e)$ means that the continuous state $x \in inv(v)$ jumps to $x' \in inv(v')$ when $\sigma \in \Sigma$ occurs[1].

We assume that $H$ has forcible events which can be forced to occur by external control actions and are controllable. Let $\Sigma_f$ be the set of forcible events. Then, note that $\Sigma_c \cap \Sigma_u = \emptyset$, $\Sigma = \Sigma_c \cup \Sigma_u$, and $\Sigma_f \subseteq \Sigma_c$. The state set $Q_H$ of $H$ is given by $Q_H = \{(v, x)|v \in V, x \in inv(v)\}$. Let $guard(e)$ be an occurrence condition of the discrete transition by $e(v, \sigma, v') \in E$, that is,

\[
guard(e) = \{x \in inv(v)|\exists x' \in inv(v') \text{ s.t. } (x, x') \in jump(e)\}.
\]

We assume that $guard(e)$ is a closed set for any $e(v, \sigma, v') \in E$ and $\sigma \in \Sigma_f$.

Let $\mathcal{F}(P, \sigma, v, x, x')$ be a set of functions $F : [0, \delta] \rightarrow \mathbb{R}^n$ with $F(0) = x$, $F(\delta) = x'$, $F(e) \in inv(v)$ and $(F(e), \dot{F}(e)) \in flow(v)$ for any $e \in (0, \delta)$, and $P(v, F(\epsilon)) = F(v, F(\epsilon))$ for any $\epsilon_i, \epsilon_j \in (0, \delta)$. Moreover, $\mathcal{F}(P, \sigma, v, x, x') = \bigcup_{x' \in inv(v')} \mathcal{F}(P, \sigma, v, x, x')$.

Let $f : Q_H \rightarrow 2^2 \times 2^\Sigma_f$ be an event controller denoted by $f = (f_1, f_2)$, where $f_1$ and $f_2$ give a set of control-enabled events and forced events by the controller, respectively. Note that, for any $q \in Q_H$, $f_2(q) \subseteq f_1(q) \cap \Sigma_f$ and $\Sigma_u \subseteq f_1(q)$. Let $T$ be a sampling period. Then, the control input signal is denoted by, for each time $t$, $f((v(nT), x(nT)))$, where $v(t), x(t)$ is a state trajectory at the time $t$ and $n = \lfloor t/T \rfloor$. Denoted by $H^f$ is $H$ controlled by the sampled-data event controller $f$.

We define transition systems to be used as semantics for the hybrid automaton $H$.

(I) A sampled-data time-abstract transition system is defined by $S^a(P) = (Q_s, Act_{sa}, \mathcal{F}_{sa}(P), Q_{s0})$, where $Q_s = Q_H \times I_1 \times I_2 \times [0, T]$ is the set of states, $Q_{s0} = \{(q_0, \gamma_1, \gamma_2, 0) \in Q_s|q_0 \in Q_{H0}\}$ is the initial state set, $Act_{sa} = \Sigma \cup \{\tau_u, \tau_c\}$ is the set of events, and $\mathcal{F}_{sa}(P)$ is the set of transition relations. Intuitive meaning
of each element of a state \((q, \gamma_1, \gamma_2, \omega) \in Q_s\) is as follows: \(q\) indicates a state of \(H\), \(\omega\) is an elapsed time from the latest sampling time, \(\gamma_1 \in \Gamma_1\) and \(\gamma_2 \in \Gamma_2\) are control patterns assigned at the latest sampling time, and \(\mathcal{F}_a(P)\) is defined as follows: Consider \(q_s = ((v, x), \gamma_1, \gamma_2, \omega)\) and \(q'_s = ((v', x'), \gamma'_1, \gamma'_2, \omega') \in Q_s\).

Then, (A) for \(\sigma \in \Sigma, (q_s, \sigma, q'_s) \in \mathcal{F}_a(P)\) iff \(\sigma \in (\gamma_1 = \gamma'_1, \gamma_2 = \gamma'_2, \omega = \omega')\), and \(\exists e(v, \sigma, v') \in E\) s.t. \((x, x') \in \text{jump}(e)\), (B) \((q_s, \tau_a, q'_s) \in \mathcal{F}_a(P)\) iff \(q = q'_s\), \(\omega = T\), and \(\omega' = 0\), and (C) \((q_s, \tau_n, q'_s) \in \mathcal{F}_a(P)\) iff \(v = v', \omega < \omega'\), and \(\exists F \in \mathcal{F}(P, \omega' - \omega, v, x, *)\) such that \(F(\omega' - \omega) = x', \gamma_1 = \gamma'_1, \gamma_2 = \gamma'_2, \) and \(\delta \notin \gamma_2 \forall t \in [0, \omega' - \omega)\) and \(e(v, \delta, \dot{e}) \in E\) with \(F(t) \in \text{guard}(e)\).

The transition relation labeled by \(\tau_n\) means the uncontrollable time elapses which cannot be interrupted by any controller.

Next, we define a transition system as semantics of the controlled hybrid automaton \(H^f\).

(II) A sampled-data time-abstract transition system controlled by \(f\) is defined by \(S^f(H^f, P) = (Q_s, \text{Act}_{sa}, \tau_a, \mathcal{F}_f^P(P), \mathcal{F}_a^P)\), where the state set \(Q_s\) is the same state set as that of \(S^a(P)\) and \(Q^f_0 = \{(q_0, \gamma_1, \gamma_2, 0) \in Q_s | q_0 \in Q_{H^0}, f(q_0) = (\gamma_1, \gamma_2)\}\). \(\text{Act}_{sa} = \Sigma \cup \{\tau_n, \tau_c\}\). \(\mathcal{F}_f^P(P) \subseteq Q_s \times \text{Act}_{sa} \times Q_s\) is defined as follows:

- (A) for each \(a \in \text{Act}_{sa} \setminus \{\tau_c\}\), \((q_s, a, q'_s) \in \mathcal{F}_a^P(P)\) iff \((q_s, a, q'_s) \in \mathcal{F}_a(P)\) and (B) \((q_s, \tau_c, q'_s) \in \mathcal{F}_f^P(P)\) iff \((q_s, \tau_c, q'_s) \in \mathcal{F}_a(P), f(q) = (\gamma'_1, \gamma'_2), \omega = T\), and \(\omega' = 0\).

We extend a predicate \(P_H : Q_H \rightarrow \{0, 1\}\) on \(Q_H\) to a predicate \(P_s \in \mathcal{P}(Q_s)\) on \(Q_s\) as follows: \(P_s(q, \gamma_1, \gamma_2, \omega) = P_H(q)\) for any \((q, \gamma_1, \gamma_2, \omega) \in Q_s\).

## 3 Control-Invariance

We extend a concept of the control-invariance to a hybrid system with sampled-data state feedback control.

**Definition 1.** Let \(H\) and \(P \in \mathcal{P}(Q_s)\) be a hybrid automaton and a predicate. A predicate \(P\) is said to be control-invariant if there exists a controller \(f\) such that satisfies \(P \leq \text{wp}_a(P, \mathcal{F}_a^P(P))\forall a \in \Sigma \cup \{\tau_n, \tau_c\}\). We call the controller \(f\) a permissive controller.

We introduce a necessary and sufficient condition for the control-invariance.

**Theorem 1.** \(P \in \mathcal{P}(Q_s)\) is control-invariant iff the following conditions hold:

\[
P(q, \gamma_1, \gamma_2, T) \leq \bigvee_{(\gamma_1, \gamma_2) \in \Gamma_1 \times \Gamma_2} P(q, \gamma_1, \gamma_2, 0) \quad \text{for any } q \in Q_H. \quad (4)
\]

Theorem 1 shows that we can restrict our interest in behavior on the time interval \([0, T]\) in order to verify the control-invariance of the hybrid automaton with an event controller. Then, if \(P \in \mathcal{P}(Q_s)\) is control-invariant, one of permissive controllers \(f\) is defined as follows: \(f(q) = (\gamma_1, \gamma_2)\) with \(P(q, \gamma_1, \gamma_2, T) \leq P(q, f(q), 0)\).

In general, a given predicate \(P \in \mathcal{P}(Q_s)\) is not necessarily control-invariant. We propose a procedure for the computation of the supremal control-invariant...
subpredicates of $P$. Let $C(P) \in 2^{\mathcal{P}(Q_s)}$ and $0 \in \mathcal{P}(Q_s)$ be the set of all control-invariant subpredicates of $P \in \mathcal{P}(Q_s)$ and the predicate with $0(q_s) = 0$ for each $q_s \in Q_s$. Note that $C(P) \neq \emptyset$ since $0 \in C(P)$. We call $P^\uparrow \in C(P)$ a supremal control-invariant subpredicate of $P$ if $P' \preceq P^\uparrow$ for each $P' \in C(I(Q_s))$.

The following theorem shows that there exists $P^\uparrow$ for any predicate $P \in \mathcal{P}(Q_s)$.

**Theorem 2.** Let $I$ be any index set. If $P_i \in \mathcal{P}(Q_s)$ is control-invariant for each $i \in I$, then $P_I = \bigvee_{i \in I} P_i$ is control-invariant.

We consider the following iterative scheme: $P_0 = P$ and, for $k = 0, 1, 2, \ldots$,

$$P_{k+1} = P_k \land \left( \bigwedge_{a \in \Sigma \cup \{\tau\}} \text{wlp}_a(P_k, \mathcal{F}_{sa}(P_k)) \right) \land \text{mlp}_{\tau_c}(P, \mathcal{F}_{sa}(P_k)),$$

where

$$mlp_{\tau_c}(P, \mathcal{F}_{sa}(P))(q_s) = \begin{cases} 1 & \text{if } \neg D_{\tau_c}(\mathcal{F}_{sa}(P)) \text{ or, } \\
 & \exists q'_s \in \text{Post}(q_s, \tau_c, \mathcal{F}_{sa}(P)) \text{ s.t. } P(q'_s) = 1, \\
0 & \text{otherwise.} \end{cases}$$

**Theorem 3.** If $P_k = P_{k+1}$ for some $k \geq 0$, then $P^\uparrow = P_k$.

Practically, the iterative computation in Theorem 3 is implemented by using a bisimulation relation, and its termination is closely related to the existence of a finite bisimulation.

### 4 Conclusion

This paper considered the sampled-data event control of a hybrid automaton with forcible events as a model of computer-controlled systems where control specifications are given by predicates. We introduced transition systems as semantics and showed a necessary and sufficient condition for the control specification to be control-invariant. Finally, we proved that there always exists the supremal control-invariant subpredicate for any predicate and proposed an iterative scheme to compute it.

The procedure for computation of the supremal control-invariant subpredicate is not decidable in general. So it is future work to investigate a condition under which the procedure is decidable.

### References


