Supervisory control of timed discrete event systems under partial observation based on activity models and eligible time bounds

Seong-Jin Park\textsuperscript{a}, Kwang-Hyun Cho\textsuperscript{b,c,*}

\textsuperscript{a}Department of Electrical and Computer Engineering, Ajou University, Suwon 443-749, Republic of Korea
\textsuperscript{b}College of Medicine, Seoul National University, Jongno-gu, Seoul 110-799, Republic of Korea
\textsuperscript{c}Bio-Max Institute, Seoul National University, Gwanak-gu, Seoul 151-818, Republic of Korea

Received 18 June 2004; received in revised form 23 August 2005; accepted 12 September 2005
Available online 8 November 2005

Abstract

To avoid the state–space explosion by including tick events in timed discrete event systems (DESs) under partial observation, a notion of eligible time bounds is introduced and based on the notion, controllability and observability conditions of languages are presented. In particular, this paper shows that these controllability and observability conditions are necessary and sufficient for the existence of a supervisor to achieve the given language specification.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Timed DESs; Partial observation; Supervisory control; Activity models; Eligible time bounds; Controllability; Observability

1. Introduction

Supervisory control of timed discrete event systems (DESs) has been reported in many papers [1–6,8–10]. Brandin and Wonham [2] introduced an event tick to represent a tick of the global clock, and the activity (or logical) models of the systems were extended by including the tick event. In particular, they introduced a forcing mechanism of supervisors for a forcible event which can preempt the tick event and also can be forced to take place within its upper time bound. However, the forcible events make the supervisory control for timed DESs significantly harder than the Ramadge and Wonham’s framework for untimed DESs [11]. We also note that some of the monotonicity of reachable sets underlying the Ramadge and Wonham framework does not hold anymore in the timed DESs framework. The resulting framework of [2] retains most properties of logical DESs, but the addition of tick event can lead to a state–space explosion. The framework was further extended by Lin and Wonham to consider timed DESs under partial observation in [8], but the problem of state–space explosion was still unsolved. In order to avoid the state–space explosion caused by including the tick events, Ho [4], Chen and Hanisch [3], Khatab and Niel [5] considered state-based supervisory control schemes. However, those results were not directly applicable to a supervisory control problem given by language specifications. Khoumsi [6] dealt with the supervisory control of timed DESs modeled by continuous timed automata, but the result did not explicitly take into account the controllability and observability conditions for the existence of a supervisor to achieve a given language specification. On the other hand, Brandin proposed to incorporate the timing information of states in the form of timer variables in [1]; however, the result was only applicable to timed DESs under complete observation.

Park et al. [9,10] presented the notions of eligible time bounds to solve bounded time constraint problems for the timed DESs introduced in [2]. The eligible time bounds can be computed from an activity model and the time bounds of events without the need for complicated timed transition models including the tick, which thereby makes it possible to avoid the state–space explosion caused by the tick.
the previous results were only applicable to timed DESs under complete observation. This paper therefore deals with a supervisory control problem of timed DESs under partial observation to extend the previous results. Based on the detection of observable events, we consider in this paper a supervisor which can enable or force the legal events when its timer value reaches a specified threshold. The partial observation in timed DESs makes it difficult to determine a correct control policy since the timed and logical behaviors could be different even for identical observations but with different event sequences. This paper shows how the notions of controllability and observability can be extended to solve the supervisory control problem of timed DESs under partial observation. Based on the extended notions, this paper presents necessary and sufficient conditions for the existence of a supervisor to achieve the given language specification in particular.

As this paper aims at presenting a method to achieve a given language specification, the proposed approach is not applicable to the cases when we need to enforce timed language specifications for real-time constraints. Note, however, that the language specifications achievable by the proposed approach include logical legal behaviors avoiding the occurrence of any undesirable situation such as a catastrophic breakdown of real-time systems. For instance, we can consider the cases of missing a critical deadline as a computer-controlled robot collides with another objects or a weapon defense system is destructed by offensive missiles. It is possible to avoid such undesirable situations by applying the proposed approach since the proposed supervisory control ensures a proper control action before the timer value of the supervisor reaches a threshold at which such an illegal event can occur. In addition, note that a projection function adopted in this paper for observable events is limited to the observable activity (logical) events although the tick is also regarded as an observable event since only the language specifications are considered as requirements. To deal with real-time specifications, the projection function presented in this paper should therefore be reformulated by including the tick event in the projection, which is, however, not to be considered in this paper.

2. Preliminaries

This paper considers the timed DESs framework presented in [2]. The activity (logical) model is represented by the finite state automaton \( G_{\text{act}} = (A, \Sigma_{\text{act}}, a_0, \delta_{\text{act}}) \), where \( A \) is the finite set of activity states, \( \Sigma_{\text{act}} \) is the set of activity events, \( a_0 \) is the initial activity state, and \( \delta_{\text{act}} : A \times \Sigma_{\text{act}} \mapsto A \) is the activity state transition (partial) function (in this paper blocking issues associated with marked states are not considered). Each event \( \sigma \) in \( \Sigma_{\text{act}} \) is assigned with a lower time bound \( l(\sigma) \in N \) and an upper time bound \( u(\sigma) \in N \cup \{\infty\} \), where \( N \) is the set of natural numbers. From the activity model and time bounds, the timed DES can be modeled by the following finite state automaton:

\[
G = (Q, \Sigma, q_0, \delta),
\]

where \( Q \) is the finite set of states, \( \Sigma \) is the set of events, \( q_0 \) is the initial state, and \( \delta : Q \times \Sigma \mapsto Q \) is the state transition (partial) function (refer to [2] for the detailed definition of its transition structure). The set \( \Sigma \) is decomposed as \( \Sigma = \Sigma_{\text{act}} \cup \{\text{tick}\} \) in which the event \( \text{tick} \) represents the tick of the global clock. A state in \( Q \) consists of an activity state and timer values of activity events, which is the main cause of the state–space explosion in timed DESs.

The set \( \Sigma_{\text{act}} \) is classified into \( \Sigma_{\text{act}} = \Sigma_{c} \cup \Sigma_{o} \cup \Sigma_{uo} \) in which \( \Sigma_{c} \) is the set of controllable events, \( \Sigma_{o} \) is the set of uncontrollable events, \( \Sigma_{u} \) is the set of observable events, and \( \Sigma_{uo} \) is the set of unobservable events. Further, in timed DESs there is the set \( \Sigma_{lt} = \Sigma_{act} \cup \{\text{tick}\} \) of forcible events which can preempt the tick event by forcing action of supervisors [2]. The forcible events can be either controllable or uncontrollable; either observable or unobservable. The tick is regarded as an observable event since supervisors can detect their timer values, but tick is regarded as an uncontrollable event when there is no feasible forcible event. Let \( \Sigma_{act} \) and \( \Sigma^* \) denote the set of all finite strings of elements in \( \Sigma_{act} \) and \( \Sigma \), respectively, including the empty string \( \varepsilon \). Then, we can define two projections \( P_{o} \) and \( P_{act} \) as follows: a projection \( P_{o} : \Sigma_{act}^* \mapsto \Sigma^*_o \): (i) \( P_{o}(\varepsilon) = \varepsilon \), (ii) for \( s \in \Sigma_{act}^* \), \( \sigma \in \Sigma_{act} \), \( P_{o}(s\sigma) = P_{o}(s)\sigma \), if \( \sigma \in \Sigma_{o} \), and \( P_{o}(s\sigma) = P_{o}(s) \), otherwise, and a projection \( P_{act} : \Sigma^* \mapsto \Sigma^*_{act} \): (i) \( P_{act}(\varepsilon) = \varepsilon \), (ii) for \( s \in \Sigma^* \), \( \sigma \in \Sigma_{act} \), \( P_{act}(s\sigma) = P_{act}(s)\sigma \), if \( \sigma \neq \text{tick} \), and \( P_{act}(s) = P_{act}(s) \), otherwise. Let \( \text{tick}^i \) denote the string of \( \text{tick} \) with length \( i \). For instance, \( \text{tick}^2 = \text{tick}\text{tick} \). In addition, we introduce \( L(s) \) to denote the last event of the string \( s \in \Sigma^* \), e.g., \( L(\sigma\varepsilon\gamma) = \gamma \) for \( \alpha, \beta, \gamma \in \Sigma \). The prefix closure of a language \( L(\subseteq \Sigma^*) \) is \( \text{pr}(L) := \{ t \in \Sigma^* | t u \in L \) for some \( u \in \Sigma^* \). L is said to be prefix-closed (or closed) if \( L = \text{pr}(L) \). For \( s \in \Sigma^* \), let \( \Sigma_L(s) := \{ \sigma \in \Sigma | s\sigma \in \text{pr}(L) \} \). Furthermore, for \( a \in A \) and \( q \in Q \), let \( \Sigma_{G_{act}}(a) := \{ \sigma \in \Sigma_{act} | l(\sigma) = l \} \) and \( \Sigma_{G_{act}}(q) := \{ \sigma \in \Sigma | \delta(q, \sigma) \} \). \( \delta_{act} \) can be extended to \( \Sigma^*_{act} \) by defining \( \delta_{act}(a, \varepsilon) := a \) and \( \delta_{act}(a, \sigma) := \delta_{act}(\delta_{act}(a, \varepsilon), \sigma) \) for all \( s \in \Sigma^*_{act} \) and \( \sigma \in \Sigma_{act} \), which is similarly applicable to \( \delta \). The closed behaviors of \( G_{act} \) and \( G \) are defined as \( L(G_{act}) := \{ s \in \Sigma^*_{act} | \delta_{act}(a_0, s) \} \) and \( L(G) := \{ s \in \Sigma^* | \delta(q_0, s) \} \), respectively.

3. Main results

We first define an eligible lower time bound (el) and an eligible upper time bound (eu) as follows [9,10]:

**Definition 1.** For \( s \in \Sigma_{act}^* \), \( x \in \Sigma_{act} \), \( el(s, x) \) and \( eu(s, x) \) are a minimum value and a maximum value of \( k \) satisfying \( \delta(q, \text{tick}^k x) \), respectively, for any \( v \in P_{act}^{-1}(s) \cap L(G) \) with \( \text{lt}(v) \neq \text{tick} \) and \( q = \delta(q_0, v) \).

After the occurrence of the last activity event of \( s \), the event \( x \) can occur after at least \( el(s, x) \) and at most \( eu(s, x) \) occurrences of \( \text{tick} \), respectively. While the time bounds \( l(x) \) and \( u(x) \) are the given delay and deadline of the event \( x \), respectively, the eligible time bounds can be different with the time bounds according to time bounds of the other events. The algorithm for computing
the eligible time bounds is presented in the following:

(i) For all \( a \in \Sigma_{G_{act}}(a_0) \),
\[
el(\varepsilon, a) := l(x), \quad eu(\varepsilon, a) := \min_{\beta \in \Sigma_{G_{act}}(\varepsilon)} u(\beta).
\]

(ii) Let \( s \in L(G_{act}) \), \( a \in \Sigma_{L(G_{act})}(s) \), and \( s' = \delta_{act}(a_0, s) \), and \( s' \in \Sigma_{act}^* \) satisfy \( s' \triangleright l(s) = s \) and \( a' = \delta_{act}(a_0, s') \). Then,

\[
eu(s, a) := \begin{cases} 0 & \text{if } l(x) < eu(s', x), \\ l(x) - eu(s', x) & \text{otherwise} \end{cases}
\]

(a.1) If \( a \in \Sigma_{G_{act}}(a') \),
\[
el(s, a) := \begin{cases} 0 & \text{if } l(x) < eu(s', x), \\ l(x) - eu(s', x) & \text{otherwise} \end{cases}
\]

(a.2) Else, \( eu(s, a) := l(x) \).

(b.1) If there exists some \( \beta \in \Sigma_{G_{act}}(a) \cap \Sigma_{G_{act}}(a') \),
\[
eu(s, a) := \min_{\beta \in \Sigma_{G_{act}}(a)} u(\beta) - \nu(s', l(s)),
\]

(b.2) Else, \( eu(s, a) := \min_{\beta \in \Sigma_{G_{act}}(a)} u(\beta) \).

Note that for any \( s' \in \Sigma_{G_{act}}(s) \), it holds that \( eu(s, a) = eu(s, a) \) and the computation of el and eu only needs the search of \( L(G_{act}) \) and time bounds of events. It does not require the search of \( L(G) \) which may cause the problem of state-space explosion.

Since a certain string \( s \in \Sigma_{act}^* \) is observed as \( P_o(s) \), the eligible time bounds are not enough to exactly describe the behavior of timed DESs under partial observation. Thus, it is necessary to extend the concept of the eligible time bounds to the case of partial observation as follows:

**Definition 2.** For \( s \in L(G_{act}) \) and \( a \in \Sigma_{act}^* \), let \( s = s' \sigma_0 \lambda_1 \cdots \lambda_n \) and \( \lambda_{n+1} = a \), where \( s' \in \Sigma_{act}^* \), \( \sigma_0 \in \Sigma_{act} \), and \( \lambda_1, \ldots, \lambda_n \in \Sigma_{un} \). Then, an eligible lower time bound (elp) and an eligible upper time bound (eup) under partial observation are defined as

\[
elp(s, a) := \nu(s' \sigma_0, \lambda_1) + \sum_{i=1}^{n} \nu(s' \sigma_0 \lambda_1 \cdots \lambda_i, \lambda_{i+1}),
\]
\[
eup(s, a) := \nu(s' \sigma_0, \lambda_1) + \sum_{i=1}^{n} \nu(s' \sigma_0 \lambda_1 \cdots \lambda_i, \lambda_{i+1}).
\]

For \( s \in \Sigma_{act}^* \) satisfying \( P_o(s) = \varepsilon \), the string \( s' \sigma_0 \) in the above definition is replaced with \( \varepsilon \). The values of elp(s, a) and eup(s, a) imply that after the occurrence of the last observable event \( \sigma_0 \) of \( s \), the event \( x \) happens after at least elp(s, a) and at most eup(s, a) occurrences of tick, respectively.

A supervisor \( S = (V, I_e, I_f, \tau) \) is defined as \( V : \Sigma_{act} \mapsto 2^{\Sigma_{act}} \), \( I_e : \Sigma_o \times \Sigma_{act} \mapsto 2^N \), \( I_f : \Sigma_o \times \Sigma_{for} \mapsto 2^N \), and \( \tau : \Sigma_o \mapsto N \). For \( s_0 \in \Sigma_{act}^* \), \( V(s_0) \) is the set of events to be enabled or forced by the supervisor after the observation of \( s_0 \). Note that \( V(s_0) \supseteq \Sigma_{uc} \) since every uncontrollable event is permanently enabled only during the interval of its eligible time bounds el and eu, while it can be forced in \( I_f \). The supervisor enables an event \( x \in V(s_0) \) if the value of a timer \( \tau(s_0) \) belongs to \( I_e(s_0, x) \), otherwise it is disabled. In addition, the supervisor forces an event \( x \in V(s_0) \) if \( x \in \Sigma_{for} \) and the value \( \tau(s_0) \) belongs to \( I_f(s_0, x) \). For \( s \in \Sigma_{for} \cap V(s_0) \), it holds in general that \( I_f(s_0, x) \subseteq I_e(s_0, x) \), \( I_f(s_0, x) \in I_f(s_0, x) \). If \( \tau(s_0) \in I_e(s_0, x) - I_f(s_0, x) \), the event \( x \) is only enabled while it is forced else when \( \tau(s_0) \in I_f(s_0, x) \). When several legal and forcible events are simultaneously enabled and have an identical forcing time, i.e. \( I_f(s_0, \cdot) \), the supervisor forces all the events simultaneously as its timer value \( \tau \) gets the forcing time. The result of the forcing action is, however, not necessarily deterministic and the selection of an event to occur depends on randomness in this case. For instance, if two legal forcible events \( x \) and \( y \) are simultaneously forced at a forcing time \( I_f(s_0, \cdot) \) then either \( x \) or \( y \) (not both) can randomly occur.

A supervised system \( S/G \) denotes the timed DES \( G \) under the control of a supervisor \( S \). The closed behavior of \( S/G \) is inductively defined as follows: (i) \( e \in L(S/G) \) and \( \tau(e) = 0 \), (ii) for \( s \in \Sigma_{act}^* \), \( \sigma \in \Sigma_{act} \), suppose that \( s \in L(S/G) \) and \( s_\sigma \in L(G) \), and let \( P_o(s) = s_0 \) and \( P_o(s_\sigma) = s_\sigma \), then

\[
(1) \text{ if } s \in \Sigma_{act} \cap V(s_0) \text{ and } \tau(s_0) \in I_e(s_0, s) \cup I_f(s_0, s), \text{ then } s_\sigma \in L(S/G) \text{ and } \tau(s_0) = 0 \text{ if } s \in \Sigma_o, \text{ otherwise } \tau(s_0) \text{ is unchanged},
\]

\[
(2) \text{ if } \tau = \text{tick} \text{ and } i < \nu(s, a) \text{ for all } \beta \in \Sigma_{L(G_{act})}(s_\sigma) \text{ in which } s = s' \sigma \text{ tick} \text{ and } \sigma \in \Sigma_{act}, \text{ then } s_\sigma \in L(S/G) \text{ and } \tau(s_0) \text{ is updated as } \tau(s_0) + 1.
\]

This paper considers the following supervisory control problem: given a closed language specification \( K \subseteq L(G_{act}) \) for a timed DES \( G \), find the existence conditions of a supervisor \( S \) such that \( P_o(L(S/G)) = K \). As mentioned in Introduction, the result presented in this paper is restrictive in the sense that it does not explicitly show a method to achieve timed language specifications reflecting real-time constraints. However, we aim at designing a supervisor that can ensure a proper control action before its timer value reaches a certain threshold at which some illegal event can occur. Hence, the proposed approach can be still useful for applications such as preventing any undesirable logical behavior that can be caused by violating real-time constraints.

To investigate the controllability and observability in timed DESs under partial observation, the timed and logical behaviors for different strings with an identical observation should be analyzed. First, for a closed \( K \subseteq L(G_{act}) \) and \( s \in K \), let

\[
B(K, s) := \begin{cases} \min_{\gamma \in \Sigma_{uc} \cap (\Sigma_{L(G_{act})}(s) - \Sigma_{K}(s))} \nu(s, \gamma), & \text{if } \Sigma_{uc} \cap (\Sigma_{L(G_{act})}(s) - \Sigma_{K}(s)) \neq \emptyset, \\ \infty, & \text{otherwise}. \end{cases}
\]

\( B(K, s) \) represents the minimum value of elp’s for illegal uncontrollable events which do not extend \( s \) inside \( K \). For the language \( K \) to be achieved, there should exist a forcible event
extending $s$ inside $K$ with an elp value smaller than $B(K, s)$. For $s_0 \in P_0(K)$, the binary relation $\subseteq_{K,s_0}$ for $s_1, s_2 \in K$ satisfying $P_0(s_1) = P_0(s_2) = s_0$ is defined as $s_2 \subseteq_{K,s_0} s_1$ iff

(i) $\exists \gamma \in \Sigma_{uc} \cap (\Sigma_{L(G_{act})}(s_2) - \Sigma_k(s_2))$ such that $\text{elp}(s_2, \gamma) < B(K, s_1)$, and

(ii) $\exists \xi \in \Sigma_{L(G_{act})}(s_2)$ and $\text{elp}(s_2, \xi) < B(K, s_1)$ for at least one

\[
\xi \in \Sigma_k(s_1) \cap \Sigma_{for} \text{ satisfying } \text{elp}(s_2, \xi) < B(K, s_1).
\]

For the two different strings $s_1$ and $s_2$ with the identical observation of $s_0$, if the relation $s_2 \subseteq_{K,s_0} s_1$ holds, then after the string $s_1$, there always exist at least one uncontrollable event not belonging to $\Sigma_k(s_1)$ and at least one forcible event belonging to $\Sigma_k(s_1)$. Moreover, after the string $s_2$, there exists a forcible event belonging to $\Sigma_k(s_1)$ with its elp value less than $B(K, s_1)$. Thus, when a supervisor forces the forcible event before $B(K, s_1)$, the event tick is preempted by the event at the moment after $s_1$ and $s_2$.

Then, for the string $s_0 \in P_0(K)$, the activity strings set $|s_0|_K$ is defined as $|s_0|_K := \{s \in \Sigma_{act} | s \in K$ and $P_0(s) = s_0\}$, and it is partitioned into

\[
|s_0|_K = g_1 \hat{\cup} g_2 \hat{\cup} \ldots \hat{\cup} g_n \hat{\cup} \xi,\]

where

(i) $g_i$ is the set of activity strings such that for some $s_i \in g_i$ and any $s' \in g_i$, $s' \subseteq_{K,s_0} s_i$, where $s_i$ is called a master of $g_i$.

(ii) $\xi$ is the set of activity strings such that for any $s, s' \in \xi$ ($s \neq s'$) and a master $g_i$ such that $s_i \subseteq_{K,s_0} s$, $s' \subseteq_{K,s_0} s_i$, and $s' \subseteq_{K,s_0} s_i$.

(iii) If $s_i$ and $s_j$ are the masters of $g_i$ and $g_j$, respectively, then $s_i \subseteq_{K,s_0} s_j$ and also $s_j \subseteq_{K,s_0} s_i$.

After any string in $g_i$, at least one forcible event defined after $s_i$ (the master of $g_i$) is enabled before the minimal value of elp’s for uncontrollable events defined after $s_i$ but not belonging to $\Sigma_k(s_i)$. Further, note that for some $s \in \xi$ and some non-master $s' \in g_i$, it may hold that $s \subseteq_{K,s_0} s'$ or $s' \subseteq_{K,s_0} s$.

Based on the aforementioned notions, we formulate extended conditions for the controllability and observability of a given language specification, and show that they are both necessary and sufficient for the existence of a supervisor which can achieve the specification.

**Definition 3.** A closed language $K$ is controllable with respect to a timed DES $G$ if the following conditions hold for all $s_0 \in P_0(K)$:

(i) $\Sigma_k(s) = \{x \in \Sigma_k(s) | \text{elp}(s, x) < B(K, s_i)\}$ for any $s \in g_i$ and $g_i$,

(ii) $\Sigma_k(s) = \{x \in \Sigma_k(s) | \text{elp}(s, x) \leq \text{eup}(s, x)\}$ for any $s \in \xi$, where $s_i$ is a master of $g_i$ and $|s_0|_K = g_1 \hat{\cup} g_2 \hat{\cup} \ldots \hat{\cup} g_n \hat{\cup} \xi$ in (i) and (ii),

(iii) for any $s \in K$ satisfying $\Sigma_{uc} \cap (\Sigma_{L(G_{act})}(s) - \Sigma_k(s)) \neq \emptyset$, there exists $x \in \Sigma_k(s) \cap \Sigma_{for}$ such that $\text{elp}(s, x) < B(K, s).

The controllability condition implies that when there are illegal uncontrollable events which do not extend a string $s$ inside $K$, there must be a forcible event extending $s$ inside $K$ such that the value $\text{elp}(s, x)$ is smaller than $B(K, s)$ (condition (iii)). Moreover, for any string after which the forcible event $x$ is defined, the set of events extending the string inside $K$ with their elp’s less than $B(K, s_i)$ must be equal to the set of events extending the string inside $K$ (condition (i)). For the other strings, the controllability only requires the feasibility, i.e. the value of elp should be equal or less than eup (condition (ii)).

The notion of controllability in [11] states that if some uncontrollable events are defined after any string in $K$ then all of these should extend the string at most inside $K$ because their occurrence cannot be avoided by supervision.

The proposed notion of controllability in this paper intuitively means that although some uncontrollable events may extend a string out of a given language specification $K$, their occurrence in a supervised system can be avoided by the forcing action of eligible and legal forcible events before a timer value of the supervisor reaches the minimal eligible lower time bound of the illegal uncontrollable events represented by $B(K, \cdot)$. The partial observation makes the situation even more complex since it is not obvious when a legal event is to be forced. The forcing action intended to avoid the occurrence of an illegal uncontrollable event after a string $s_1 \in K$ can be actually carried out for forcible events after another string $s_2 \in K$ satisfying $P_0(s_1) = P_0(s_2)$. Hence, in order for $K$ to be achieved, the event sets extending $s_1$ and $s_2$ inside $K$ with eligible lower time bounds less than the minimal eligible lower time bound of the illegal uncontrollable events ($B(K, s_i)$) should be equal to $\Sigma_k(s_1)$ and $\Sigma_k(s_2)$, respectively. In this way, although illegal uncontrollable events are defined, the language $K$ still can be achieved depending on the presented controllability condition.

We next consider how the observability condition can be extended to timed DESs under partial observation. For a closed $K \subseteq L(G_{act}), s \in K$, and $\sigma \in \Sigma_{act}$, let

\[
T(K, s, \sigma) := \begin{cases} & \{l \in N | \text{elp}(s, z) \leq l < B(K, s_i)\} \\
& \text{if } s \in g_i \text{ for some } g_i, \\
& \{l \in N | \text{elp}(s, z) \leq l \leq \text{eup}(s, z)\} \\
& \text{if } s \in \xi,
\end{cases}
\]

where $s_i$ is a master of $g_i$.

**Definition 4.** A closed language $K$ is observable with respect to a timed DES $G$ if the following conditions hold for any $\sigma \in \Sigma_c$ and $s_1, s_2 \in K$ such that $P_0(s_1) = P_0(s_2)$, $s_1 \sigma \in K$ and $s_2 \sigma \in L(G_{act}): s_2 \sigma \in K$ or $T(K, s_1, \sigma) \cap T(K, s_2, \sigma) = \emptyset$ otherwise.

The observability condition states that for the two different strings $s_1$ and $s_2$ with an identical observation, there should be no conflict by the event $\sigma$ in control decision, i.e. $s_1 \sigma, s_2 \sigma \in K$. If there is a conflict by the event, it requires that there should be no time instant at which the conflicted event is enabled.
simultaneously after \( s_1 \) and \( s_2 \), i.e. \( T(K, s_1, \sigma) \cap T(K, s_2, \sigma) = \emptyset \). For the case of \( s_2 \sigma \not\in K \), the conflict can be avoided by disabling the event \( \sigma \) during the time interval \( T(K, s_2, \sigma) \).

The observability condition in [7] only requires that there should be no conflict between the events defined after two different strings with an identical observation. This condition is originally formulated to achieve a given language \( K \) under partial observation. However, in timed models, the control policy of a supervisor is associated with not only the events but also their eligible time bounds. Hence, even if a conflicting decision is required for an event after an identical observation of different strings due to partial observations, the achievement of \( K \) can be still possible by disabling the event during its eligible time bounds permitting illegal extension if the eligible time bounds of the event for the strings do not overlap with each other. Only during the time interval for which the extension of a string by an event is legal, the event should be enabled and for other time intervals, it should be disabled. In other words, a language specification \( K \) cannot be achieved if the events conflict with each other and their eligible time bounds overlap as well under partial observation.

The computational complexity for verifying the presented controllability and observability conditions is as follows. First, we note that for each \( s \in K \) and \( x \) satisfying \( s x \in K \), the values such as \( \text{elp}(s, x) \), \( \text{eup}(s, x) \), \( B(K, s) \), and \( T(K, s, x) \) can be computed separately in prior steps. For real implementations, these can be stored in a certain memory location associated with the pointer \((s, x)\) as a data structure form. Hence, to check the conditions, only the values of the associated data structure for each \( s \in K \) and \( x \) satisfying \( s x \in K \) need to be compared, e.g. whether the value \( \text{elp}(s, x) \) is less than \( B(K, s_1) \) (the condition (i) in Definition 3) or whether \( T(K, s_1, \sigma) \cap T(K, s_2, \sigma) \) is an empty set or not (the condition in Definition 4). Therefore, we know that the computational complexity for verifying the controllability becomes \( O(m ||A|| ||\Sigma_{\text{act}}||) \) and that of the observability condition becomes \( O(m^2 ||A|| ||\Sigma_{\text{act}}||) \), respectively, where \( m \) denotes the number of states in the automaton realization of the language \( K \). On the other hand, the clock tick models of [2,8], in worst cases, require to search \( mp \) states where \( p \) is the maximum upper time bound of events, and as a result the complexity becomes \( O(m ||A|| p^2 ||\Sigma_{\text{act}}||) \) for controllability and \( O(m^2 ||A|| p^3 ||\Sigma_{\text{act}}||) \) for observability, respectively.

The following theorem presents the main result of this paper.

**Theorem 1.** For a closed language \( K(\neq \emptyset) \subseteq L(G_{\text{act}}) \), there exists a supervisor \( S \) for a timed DES \( G \) such that \( P_{\text{act}}(L(S/G)) = K \) if and only if \( K \) is controllable and observable with respect to \( G \).

**Proof.** (If) Consider a supervisor \( S = (V, I_c, I_f, \tau) \) defined as follows:

\[
V(s_0) = \{ x \in \Sigma_c | s_0 x \in |s_0|_K \text{ and } s_0 x \in K \} \cup \Sigma_{\text{uc}},
\]

\[
I_c(s_0, x) = \bigcup_{s} T(K, s, x) \text{ for } x \in V(s_0) \text{ and } s_0 \in |s_0|_K,
\]

\[
I_f(s_0, x) = B(K, s_0) - 1 \text{ for } x \in \Sigma_K(s_0) \text{ and a master } s_0 \text{ of some } g_i,
\]

\[
\tau(x) = 0.
\]

The proof can be done by induction on the length of the strings in the two languages \( P_{\text{act}}(L(S/G)) \) and \( K \). The base case is for strings of length 0. By definition, \( e \in P_{\text{act}}(L(S/G)) \) and \( e \in K \). Thus the base case simply holds. The induction hypothesis is that for all strings \( s_0 \) such that \( |s_0| \leq n, s_0 \in P_{\text{act}}(L(S/G)) \) if and only if \( s_0 \in K \), where \( |s_0| \) denotes the length of \( s_0 \). Let us prove the same for strings of the form \( s a \sigma \) where \( |s_0| = n \) and \( \sigma \in \Sigma_{\text{act}} \). Let \( s \in L(S/G) \), \( P_{\text{act}}(s) = s_0 \), and \( P_0(s) = s_0 \).

First, let \( s_0 \sigma \in P_{\text{act}}(L(S/G)) \) and assume that \( s_0 \sigma \not\in K \). Then we obtain \( \sigma \in V(s_0) \) by the definition of \( S \). Let us consider the following two cases: (1) \( \sigma \in \Sigma_{\text{uc}} \), (2) \( \sigma \not\in \Sigma_{\text{uc}} \).

Case 1: \( \sigma \in \Sigma_{\text{uc}} \): The relations \( s_0 \sigma \in P_{\text{act}}(L(S/G)) \), \( \sigma \in V(s_0) \), and \( s_0 \sigma \not\in K \) imply that there exists \( s' \in |s_0|_K \) such that \( s' \sigma \in K \) and \( T(K, s_0, \sigma) \cap T(K, s', \sigma) = \emptyset \). It contradicts the observability assumption of \( K \).

Case 2: \( \sigma \not\in \Sigma_{\text{uc}} \): Since \( K \) is controllable, there exists \( x \in \Sigma_K(s_0) \cap \Sigma_{\text{for}} \) satisfying \( \text{elp}(s_0, x) < B(K, s_0) \). By the above definition of \( I_f \), it follows that \( B(K, s_0) - 1 \notin I_f(s_0, x) \). Then, the supervisor \( S \) forces the event \( x \) when \( \tau(s_0) = B(K, s_0) - 1 \), which results in tick \( \sigma \not\in L(S/G) \) for any \( i \). Thus, it holds that \( s_0 \sigma \not\in P_{\text{act}}(L(S/G)) \) which contradicts the assumption. This completes the proof for \( s_0 \sigma \in K \).

To prove the other direction of the induction step, let \( s_0 \sigma \in K \). Then, the controllability of \( K \) implies \( (1) \) \( \text{elp}(s_0, \sigma) < B(K, s_1) \) for some \( g_i \) with \( s_0 \in |s_0|_K \), where \( s_1 \) is a master of \( g_i \), or \( (2) \) \( \text{elp}(s_0, \sigma) < \text{eup}(s_0, \sigma) \) for some \( fl \) with \( s_0 \in fl \). For both cases, it holds that \( T(K, s_0, \sigma) < I_c(s_0, x) < B(K, s_1) \), by the above definition of \( I_c \). Thus, when the timer \( \tau(s_0) \) reaches a value in \( I_c(s_0, \sigma) \), the event \( \sigma \) is enabled by the supervisor \( S \) and as a result \( s_0 \sigma \in L(S/G) \). It then follows that \( s_0 \sigma \in P_{\text{act}}(L(S/G)) \). This completes the proof of the whole induction steps.

(Only if) Assume that a supervisor \( S \) satisfies \( P_{\text{act}}(L(S/G)) = K \). First, let us prove that \( K \) is controllable with respect to \( G \) using a contradiction method. According to the definition of controllability, the following three cases can be considered.

Case 1: For some \( s_0 \in P_0(K) \), \( s \in |s_0|_K \) and \( g_i \) with \( s \in g_i \), assume that \( \Sigma_K(s) \neq \{ x | x \in \Sigma_K(s) \} \text{ \text{elp}(s, x) < B(K, s_1) \} \), where \( s_1 \) is a master of \( g_i \). Then, there exists \( x \in \Sigma_K(s) \) satisfying \( \text{elp}(s, x) < B(K, s_1) \). This means that there exists \( \gamma \in \Sigma_{\text{uc}} \cap \Sigma_{\text{act}} \) such that \( \text{elp}(s, \gamma) \leq \text{elp}(s, x) \) by the definition of \( B(K, s_1) \). Further, it follows from \( s_0 \sigma \in K \) and \( P_{\text{act}}(L(S/G)) = K \) that \( s_0 \sigma \in P_{\text{act}}(L(S/G)) \) which implies \( \sigma \in V(s_0) \) and \( \text{elp}(s, x) \in I_c(s_0, x) \). Since \( \gamma \in \Sigma_{\text{uc}} \), it is true that \( \gamma \in V(s_0) \) and hence, in order to avoid the occurrence of \( \gamma \) after \( s \), the supervisor \( S \) should force a forcible event before its timer value \( \tau(s_0) \) becomes \( \text{elp}(s, \gamma) \). However, since \( \text{elp}(s, \gamma) < \text{elp}(s, x) \), the forcing action leads to the non-occurrence of the event \( \gamma \) after \( s \), i.e. \( s \sigma \not\in P_{\text{act}}(L(S/G)) \). This is a contradiction to the assumption \( P_{\text{act}}(L(S/G)) = K \).

Case 2: For some \( s_0 \in P_0(K) \) and \( s \in |s_0|_K \), let \( s \in fl \). Since \( s \in fl \), there is no forcible event after \( s \) to be forced in order to avoid the occurrence of illegal uncontrollable events. Hence, for the string \( s \), we only need to show that the events extending
We consider a simple example illustrated in Fig. 1 to explain satisfying \( P_s \) satisfying \( \Sigma_s \) sets interval bounds as is the case for the clock tick models of [2,8]. However, it never occurs in the timed DES \( G \), i.e. \( \Sigma_x \notin P_{act}(L(G)) \) which also implies \( \Sigma_x \notin P_{act}(L(S/G)) \). Since \( \Sigma_x \notin K \), this contradicts \( P_{act}(L(S/G)) = K \).

Case 3: For some \( s \in K \) satisfying \( \Sigma_{uc} \cap (\Sigma_{L(G_{act})} - \Sigma_K(s)) \neq \emptyset \), assume that there does not exist any \( \alpha \in \Sigma_{uc} \cap \Sigma_{L_{for}} \) such that \( \epsilon p(s, \alpha) < B(K, s) \). Then, according to the definition of \( B(K, s) \), there exists \( \gamma \in \Sigma_{uc} \cap (\Sigma_{L(G_{act})} - \Sigma_K(s)) \) satisfying \( \epsilon p(s, \gamma) = B(K, s) \). In order to avoid the occurrence of \( \gamma \) after \( s \), the supervisor should force a forcible event before its timer \( \tau \) reaches the value \( B(K, s) \). However, there is no forcible forcible event after \( s \), and hence the occurrence of \( \gamma \) is not avoidable, i.e. \( s \gamma \notin K \), it is also a contradiction.

From the above cases, we conclude that \( K \) is controllable with respect to \( G \).

To prove that the observability of \( K \) is also satisfied, assume that for some \( \sigma \in \Sigma \) and \( s_1, s_2 \in K \) such that \( P_o(s_1) = P_o(s_2) \), \( s_1 \sigma \in K \) and \( s_2 \sigma \notin K \), it holds that \( s_2 \sigma \notin K \) and \( T(K, s_1, \sigma) \cap T(K, s_2, \sigma) \neq \emptyset \). Then, it follows from \( P_{act}(L(S/G)) = K \) that \( s_2 \sigma \in P_{act}(L(S/G)) \) which implies the supervisor \( S \) enables the event \( \sigma \) when its timer reaches a value in \( T(K, s_1, \sigma) \). Since \( T(K, s_1, \sigma) \cap T(K, s_2, \sigma) \neq \emptyset \), the supervisor also enables the event \( \sigma \) after the string \( s_2 \) when its timer reaches a value in \( T(K, s_1, \sigma) \cap T(K, s_2, \sigma) \) that is a time instant at which the event \( \sigma \) is simultaneously enabled after \( s_1 \) and \( s_2 \). Thus, it holds that \( s_2 \sigma \in P_{act}(L(S/G)) \). However, since \( s_2 \sigma \notin K \), it is a contradiction to \( P_{act}(L(S/G)) = K \). Therefore, we conclude that \( K \) is observable with respect to \( G \).

We now consider the computational complexity for \( I_c \) and \( I_i \) introduced in the proof of Theorem 1. For \( s \in K \) and \( \sigma \in \Sigma_{act} \) satisfying \( s \sigma \in K \), it is required to search another string \( s' \) satisfying \( P_o(s) = P_o(s') = s_0 \). Initially, \( I_c(s_0, s) \) is set to an empty set. If a string \( s' \) is found, then a union operation of the sets \( T(K, s', \sigma) \) and \( I_c(s_0, s) \) is conducted. It is not required to carry out other operations for each discrete time value in the time interval \( T(K, s', \sigma) \). Hence, we do not need to enumerate all Cartesian products of the discrete state set and the time interval bounds as is the case for the clock tick models of [2,8]. We consider a simple example illustrated in Fig. 1 to explain the computational procedure of \( I_c \) where \( P_o(s_1) = P_o(s_2) = P_o(s_3) = s_0 \). We conclude that the computational complexity for \( I_c \) is \( O(m^2 \| \Sigma_{act} \|) \), where \( m \) denotes the number of states in the automaton realization of the language \( K \). On the other hand, the computational complexity for \( I_i \) becomes \( O(m^2 \| \Sigma_{for} \|) \) from the following reason. For any \( s_0 \in P_o(K) \), it is required to search a string \( s \) satisfying \( P_o(s) = s_0 \). If one \( s \) is found, then we need to check whether the string is a master of some \( g_i \) or not. This can be easily verified since it is the value computed and assigned for each string in the prior step. If \( s \) is a master of some \( g_i \), i.e. \( s_i \), then it is needed to search a forcible event \( \alpha \) defined after the string and then the value \( B(K, s_i) = 1 \) is inserted into \( I_i(s_0, \alpha) \). The clock tick model of [8], in worst cases, requires to search the state–space with \( mp \) states where \( p \) is the maximum upper time bound of events. Hence, the design of a supervisor with partial observations based on the clock tick model requires the computational complexity of \( O(m^2 \| \Sigma_{act} \|) \). We therefore conclude that the proposed approach is more efficient than the existing methods based on clock tick models with regard to the aspect of computational complexity. We note that this has been possible by pursuing several divided computations instead of a direct massive computation. However, we also note that the computational complexity of the proposed approach can be still exponential since the cardinality \( m \) of the state–space grows exponentially if different modules are composed.

4. An illustrative example

Consider a timed DES \( G \) with the activity model \( G_{act} \) shown in Fig. 2 and the following time bounds of the events: \((d_1, 1, 1), (d_2, 1, 1), (d_3, 1, 1), (d_4, 1, 1), (a_1, \infty), (b, 2, \infty), \) and \((c, 2, 3)\) where \((d_1, 1, 1)\) means \( f(d_1) = 1 \) and \( u(d_1) = 1 \). It is assumed that \( (a, b) \), \( \Sigma_o = \{a, b, c\} \) and \( \Sigma_{for} = \{a\} \). For several \( s \in L(G_{act}) \) satisfying \( P_o(s) = \epsilon \), the eligible time bounds of events are as follows: \( e(d_1, a) = 1, e(d_1, c) = 2, e(d_1, a) + e(d_1, c) = 3, e(d_2, a) = 1, e(d_2, b) = 2, e(d_2, a) = e(d_2, b) = \infty, e(d_3, a) = 1, e(d_3, c) = 2, \) and \( e(d_3, b) = \infty \). Finally, the values of \( e(d_1, a) \) and \( e(d_1, c) \) are as follows: \( e(d_1, a) + e(d_1, c) = 2, e(d_1, c) = e(d_1, c) + e(d_1, c) = 3, e(d_1, a) = e(d_1, c) = e(d_1, c) = e(d_1, c) = 4, e(d_2, a) = 2, e(d_2, b) = 3, e(d_2, a) = e(d_2, b) = \infty, e(d_3, a) = 3, e(d_3, c) = 4, \) and \( e(d_3, b) = e(d_3, c) = 5 \). We first consider a language \( K_1 = \{a, b, c\} \) and \( |s_0|_K = \{a, b, c\} \). Let \( s_0 = a \). Then, \( B(K_1, d_1) = e(d_1, c) = 3 \) and \( |s_0|_K = |s_1, d_1, d_3, d_3| = g_1 \), where \( g_1 = |d_1, d_2| \).
We can now show that \( K_1 \), we conclude that there exists a supervisor \( a \) is observable with respect to \( \{ T( K_1) \} \) to the definition presented in the “if” part of the proof of Theorem 3. However, for \( d_3 \) to \( G \) and according to Theorem 1, we conclude that there exists a supervisor \( a \) satisfying \( P_{act} = \{ d_3 \}, a \notin K_2 \) but \( d_2 \notin K_2 \). However, it follows from \( d_2 \) to \( g_1 \) and \( d_4 \) to \( T( K_2) \) and \( T( K_2, d_4) \), hence \( T( K_2, d_2, b) \in \Theta \) and \( T( K_2, d_4, b) \in \emptyset \). For the other strings, the observability condition is also satisfied. Therefore, \( K_2 \) is observable with respect to \( G \), and according to Theorem 1, we conclude that there exists a supervisor \( S \) satisfying \( P_{act} = \{ L(3) \} = K_2 \). The supervisor can be designed according to the definition presented in the “if” part of the proof of Theorem 1. For example, when there is no observed event from the initial state, \( V = \{ d_1, d_2, d_3, d_4, a, b, c \} \), \( I_{act}(a, a) = \{ 2 \}, I_{act}(c, b) = \{ 3, 4, 5 \}, I_{act}(d_3, c) = \{ 4, 5 \}, \) and \( I_{act}(a, c) = \{ 2 \} \). When the timer value \( \tau(a) \) reaches 2, the supervisor forces the event \( a \) to prevent the occurrence of the event \( c \) when \( d_1 \) has occurred. If the string \( d_3 \) has occurred, the forcing of the event \( a \) does not influence the occurrence of the event \( c \) because the forcible event \( a \) is not defined after \( d_3 \).

5. Conclusions

In this paper, the controllability and observability of languages have been generalized to formulate the existence conditions of a supervisor which can achieve the given language specifications for timed DESs under partial observation. In particular, this paper has shown that the existence conditions can be verified based on the eligible time bounds without considering timed transition models including the tick events which can cause the problem of state–space explosion.

Acknowledgements

This work was supported by an Ajou University grant, by a grant from the Korea Ministry of Science and Technology (Korean Systems Biology Research Grant, M10503010001-05N030100111), and by the 2IC Frontier Microbial Genomics and Application Center Program, Ministry of Science and Technology (Grant MG05-0204-3-0), Republic of Korea.

References