Non-blocking supervisory control under bounded time constraints based on non-deterministic timed transition models

S.-J. Park, K.-H. Cho and J.-T. Lim

Abstract: A non-blocking supervisory control problem to meet bounded time constraints for non-deterministic real-time discrete event systems (DESSs) is addressed. To this end, notions of trace controllability and time controllability are introduced for the non-deterministic real-time DESSs with respect to a timed language specification representing a bounded time constraint. The existence conditions of a non-blocking supervisor to achieve the timed language specification are presented on the basis of the proposed notions, and these are further illustrated by an assembly work station example.

1 Introduction

Supervisory control of real-time discrete event systems (DESSs) has been studied on the basis of various modelling schemes such as clock automata [1], timed automata [2, 3], timed transition models [4], timed event graphs [5], timed Petri net [6] and timed discrete event models [7]. Particularly, in the work of Brandin and Wonham [4], a tick of the global clock is introduced as an event tick, and the deterministic activity models describing the logical behaviour of the systems are extended accordingly by including the tick event and by augmenting the state space to include a timer for each activity event. The resulting framework retains most properties of logical deterministic DESSs, but the addition of tick event may cause the state space explosion. The works of Chen and Hanisch [6], Ho [7] and Khatib and Niel [8] presented state-feedback control to avoid the state space explosion. Also, the work of Brandin [9] incorporated the timing information in states in the form of timer variables, which allowed the compacter modelling of deterministic real-time DESSs by eliminating tick transitions. However, the bounded time constraint problem of this paper has not been considered in the work of Brandin [9].

The works of Park et al. [10, 11] introduce the notions of eligible time bounds, trace controllability and time controllability in order to solve bounded time constraint problems in deterministic real-time DESSs. They show that the notions can be used to avoid the state space explosion due to the addition of tick. In this paper, we study how these notions can be extended in non-deterministic real-time DESSs with non-blocking requirements. The supervisory control of non-deterministic DESSs in the absence of real-time constraints has been well developed in the literature including the works of Kumar and Shayman [12] and Heymann and Lin [13]. The non-deterministic real-time DESSs considered in this paper are represented as non-deterministic timed transition models. On the basis of the developed notions of eligible time bounds and trace and time controllabilities, this paper presents the necessary and sufficient conditions for the existence of a non-blocking supervisor to achieve a given timed language specification that represents a bounded time constraint.

2 Non-deterministic timed transition models and supervisory control

The non-deterministic timed transition models considered in this paper are the extension of timed transition models developed in Brandin and Wonham [4]. In order to describe the logical (or untimed) behaviour of a non-deterministic real-time DESS, the activity model is represented by the following non-deterministic automaton: $G_{act} = (A, \Sigma_{act}, q_0, \delta_{act}, A_m)$, where $A$ is the finite set of activity states, $\Sigma_{act}$ is the set of activity events, $q_0$ is the initial activity state, $\delta_{act}$: $A \times \Sigma_{act} \rightarrow 2^A$ (power set of $A$) is the activity state transition (partial) function and $A_m \subseteq A$ is the set of activity marked states that represent the completion of some operations or tasks. In $\Sigma_{act}$, each event $\sigma$ is equipped with a lower time bound $l(\sigma) \in N$ and an upper time bound $u(\sigma) \in N$, where $N$ is the set of natural numbers. The event must occur within $l(\sigma)$ and $u(\sigma)$ if there is no other feasible event at the current state except that event. However, if there exists any other feasible event, then the event may occur within $l(\sigma)$ and $u(\sigma)$ because the other feasible event can occur instead of that event. From the activity model and time bounds, the non-deterministic real-time DESS can be modelled as a non-deterministic timed transition model represented by the following non-deterministic automaton: $G = (Q, \Sigma, q_0, \delta, Q_m)$ where $Q$ is the finite set of states, $\Sigma = \Sigma_{act} \cup \{tick\}$ (the additional event tick represents the tick of the global clock), $q_0$ is the initial state, $\delta$: $Q \times \Sigma \rightarrow 2^Q$ is the state transition (partial) function and $Q_m \subseteq Q$ is the set of marked states. The transition structure of $\delta$ is according to the rules presented in Brandin and Wonham [4]. A state in $Q$ consists
of an activity state and timer values of activity events [for 
$q \in Q$, let act(q) be the activity state of $q$]. If the activity 
state in $q \in Q$ is activity marked, then $q \in Q_m$ whatever 
the values of timers may be.

Let $\Sigma_{act}$ and $\Sigma^+$ denote the set of all finite strings 
of elements in $\Sigma_{act}$ and $\Sigma$, respectively, including the empty 
string $\varepsilon$. The transition function $\delta_{act}$ can be extended 
to $\Sigma_{act}$ by defining $\delta_{act}(a, e) := [a]$ and 
$\delta_{act}(a, s, r) := \bigcup_{l \in L_{\delta_{act}}(a, s, r)} \delta_{act}(a^l, \sigma)$ for all $s \in \Sigma^+$ and $\sigma \in \Sigma_{act}$, which 
is similarly applicable to $\delta$. Any subset of $\Sigma^+$ is called a 
language over $\Sigma$. Let $tick'$ denote the string of tick 
with length $i$. For instance, $tick^i = tick$ tick. Moreover, we use 
$le(s)$ to denote the last event of the string $s \in \Sigma^+$, for 
example, $le(\alpha\beta\gamma) = \gamma$ for $\alpha, \beta, \gamma \in \Sigma$. For $s \in \Sigma^+$, $pr(s)$ denotes the set of all strings that are prefixes of $s$, that is, 
$pr(s) := \{t \in \Sigma^* | tu = s \text{ for some } u \in \Sigma^* \}$. The prefix 
closure $pr(L)$ of a language $L \subseteq \Sigma^*$ is the set of prefixes 
of all strings in $L$. $L$ is said to be prefix-closed (or closed) if 
$L = pr(L)$. For $s \in \Sigma^*$, if $s \in pr(L)$, then let 
$\Sigma_1(s) := [\sigma \in \Sigma | \sigma \in pr(L)]$, otherwise 
$\Sigma_1(s) := \emptyset$. Furthermore, for $a \in A$ and $q \in Q$, let $\Sigma_{act}(a) := 
[\sigma \in \Sigma | \sigma \in \delta_{act}(a, s)]$ (is defined) and 
$\Sigma_{act}(q) := [\sigma \in \Sigma | \sigma \in \delta(q, s)]$. Let us define a projection $P_{act} : \Sigma^+ \rightarrow \Sigma_{act}$ 
as follows: (i) $P_{act}(\varepsilon) = [\varepsilon]$, (ii) for $s \in \Sigma^+$, $\sigma \in \Sigma$, 
$P_{act}(\sigma) = P_{act}(s\sigma)$, if $\sigma \neq tick$ and $P_{act}(\sigma) = P_{act}(s)$, 
otherwise. The closed behaviours of $G_{act}$ and $G$ are 
represented by $L(G_{act}) := [s \in \Sigma^* | \delta_{act}(a_0, s)]$ 
and $L(G) := [s \in \Sigma^* | \delta(q_0, s)]$, respectively. Note that 
$L(G_{act})$ and $L(G)$ are prefix-closed languages. In addition, 
the marked behaviours of $G_{act}$ and $G$ are represented by 
$L_m(G_{act}) := [s \in \Sigma^* | \delta_{act}(a_0, s), s \in A_m] \neq \emptyset$ 
and $L_m(G) := [s \in \Sigma^* | \delta(q_0, s) \cap Q_m \neq \emptyset]$, respectively. Note 
that if $s \in L_m(G)$ then $P_{act}(s) \in L_m(G_{act})$.

To describe the bounded time behaviours of real-time 
DESs, we define the eligible lower time bound (ELTB) $el$ 
and eligible upper time bound (EUTB) $eu$ as follows.

**Definition 1**: For $s \in \Sigma^+$, $a \in A$, $\alpha \in \Sigma_{act}$, and 
v $P_{act}(s) \cap L(G)$ satisfying $le(v) \neq tick$, let $q \in \delta(q_0, v)$ 
and act(q) = $a$. Then 
$$el(s, a, \alpha) :=$$ a minimum value of $i$ such that 
$$\delta(q, tick^i) \alpha \text{ is defined}$$ 
$$eu(s, a, \alpha) :=$$ a maximum value of $i$ such that 
$$\delta(q, tick^i) \alpha \text{ is defined}$$

Here, we note that (i) for the string $s$ and the activity state $a$, 
the string $v$ and the state $q$ are not unique because the activity 
event $le(v)$ occurs within its time bounds $l(le(v))$ and $u(le(v))$ 
and (ii) the existence of the state $q$ is always guaranteed by 
the existence of the string $v$ because $v \in L(G)$ and 
$q \in \delta(q_0, v)$. 

The $el$ and $eu$ mean that, at the state $q$ (whose activity 
state is $a$) reaching via the string $v$ (whose $P_{act}$ projection 
is $s$), the activity event $\alpha$ can occur after at least $el(s, a, \alpha)$ 
and at most $eu(s, a, \alpha)$ occurrences of tick, respectively. 
Note that the eligible time bounds of certain activity 
events after the occurrence of a string are determined by 
the string in deterministic real-time DESs because 
the activity state reached by a string is unique [10, 11]. 
However, in the non-deterministic DESs, a string 
can lead to different activity states, and accordingly, 
the timed behaviours after the states may be different. Thus, 
the eligible time bounds in non-deterministic real-time 
DESs are determined by a string and the activity state 
reached by the string.

The algorithm for computing eligible time bounds is 
presented as follows

(i) For all $\alpha \in \Sigma_{act}(a_0)$ 
$$el(\varepsilon, a_0, \alpha) = l(\alpha), \quad eu(\varepsilon, a_0, \alpha) = \min_{\beta \in \Sigma_{act}(a_0)} u(\beta)$$

(ii) Let $s \in L(G_{act}), a \in \delta_{act}(a_0, s), \alpha \in \Sigma_{act}(a), a_0 \in \delta_{act}(a_0, le(s))$, where $a' \in \delta_{act}(a_0, s')$ for $s' \in \Sigma_{act}$ satisfying 
$s' le(s) = s$. Then 

(a.1) If $\alpha \in \Sigma_{act}(a')$ and $le(s) \neq \alpha$ 
$$el(s, a, \alpha) = 0, \quad \text{if } l(\alpha) < eu(s', a', \alpha), \quad \text{otherwise}$$

(a.2) Otherwise, $el(s, a, \alpha) = l(\alpha)$ 

(b.1) If there exists some $\beta \in \Sigma_{act}(a) \cap \Sigma_{act}(a') - le(s)$ 
$$eu(s, a, \alpha) = \min_{\beta \in \Sigma_{act}(a) \cap \Sigma_{act}(a')} \min_{\gamma \in \Sigma_{act}(a)} u(\gamma), \quad \text{otherwise}$$

(b.2) Otherwise, $eu(s, a, \alpha) = \min_{\beta \in \Sigma_{act}(a)} u(\beta)$

Note that for any $\alpha, \beta \in \Sigma_{act}(a)$, where $a \in \delta_{act}(a_0, s)$, 
eu(s, a, a) = eu(s, a, a). The intuitive meaning of this 
algorithm is as follows.

For (i): At the initial state $a_0$, the value of $el$ of each event 
is its lower time bound and the value of $eu$ is the minimum 
value of upper time bounds of the events defined at $a_0$. 
Thus, the values of $eu$ for all the events defined at $a_0$ are 
identical.

For (a.1) and (a.2): When the event $\alpha$ is defined at both 
the present state $a$ and the previous state $a'$, the timer 
for this event seamlessly operates during the state transition. 
Hence, the value of $el$ at the present state can be found 
when the event $le(s)$ occurs at the state $a'$ after the 
maximum time elapse, that is, $eu(s', a', \alpha)$.

For (b.1) and (b.2): When there exists some event $\beta$ 
($\neq le(s)$) defined at both $a'$ and $a$, the timer for $\beta$ 
seamlessly operates. Hence, when the event $le(s)$ occurs 
after the minimum time elapse at $a'$, that is, $el(s', a', \alpha)$, 
the event $\beta$ at the present state occurs with a 
maintenance delay. However, the value of $eu$ is finally 
determined by the minimum value of upper time bounds of 
the events defined only at $a$ and the upper time bounds 
subtracted by the minimum time elapse at $a'$ for the events 
defined at both $a$ and $a'$.

**Remark 1**: Although an activity event is assumed to have 
one fixed timing constraint $l(\cdot)$ and $u(\cdot)$, it can actually 
have more than one timing constraint by allowing variable 
lower and upper time bounds according to the reachable 
states. In this case, the presented algorithm is, however, 
not changed except the computation part using $l(\cdot)$ and 
$u(\cdot)$, where these should be replaced with the new time bounds.
Lemma 1: The eligible time bounds computed by the earlier mentioned algorithm satisfy the conditions presented in Definition 1.

Proof: (a.1) According to the transition structure defined in Brandin and Wonham [4], when the event \( a \) (\( \neq \text{let}(s) \)) is defined at the activity states \( a' \) and \( a' \) in \( G_{act} \) that is, \( a \in \Sigma_{s}(a') \cap \Sigma_{s}(a) \), the event can occur at the present state \( a \) after \( l(a) \) occurrences of tick subtracted by the number of tick occurrences in the previous state \( a' \). Hence, a minimum value of \( i \) satisfying \( \delta_{q}, \text{tick'}(a) \rangle \) for any \( v \in P_{act}(s) \cap L(G) \) with \( l(v) \neq \text{tick} \) and \( q \in \delta(qo), v \rangle \) is determined by a maximum number of tick occurrences at \( a' \), that is, \( eu(s', a', a) \). So, when tick \( s' \) has occurred \( eu(s', a', a) \) times at \( a' \) and the value is larger than \( l(a) \), the event \( a' \) can immediately occur when the system reaches the state \( a \). In addition, if the value \( l(a) \) is larger than or equal to \( eu(s', a', a) \), the event can occur after \( l(a) \) occurrences of tick at \( a' \) independently of the number of tick occurrences at \( a' \) by the transition rules described by Brandin and Wonham [4]. Thus, the minimum value of \( i \) satisfying \( \delta_{q}, \text{tick'}(a) \rangle \) for any \( v \in P_{act}(s) \cap L(G) \) with \( l(v) \neq \text{tick} \) and \( q \in \delta(qo), v \rangle \) is \( l(a) \).

The problem is equivalent to the transition rules defined in Brandin and Wonham [4] implies that (i) the EUTB of an event at a state is determined by the smallest upper time bound of events defined at the state; (ii) when there exists an event \( \beta \) (\( \neq \text{let}(s) \)) satisfying \( \beta \in \Sigma_{s}(a') \cap \Sigma_{s}(a) \), the maximum occurrence of tick at the present state \( a \) happens when tick has minimally occurred at the previous state \( a' \). Thus, if the maximum number of tick occurrences at \( a' \), that is, \( el(s', a', l(s)) \) is larger than \( u(\beta) \), then the maximum value of \( i \) satisfying \( \delta_{q}, \text{tick'}(a) \rangle \) for any \( v \in P_{act}(s) \cap L(G) \) with \( l(v) \neq \text{tick} \) and \( q \in \delta(qo), v \rangle \) is 0, and the EUTBs of all events defined at \( a \) are also 0. When \( u(\beta) > el(s', a', l(s)) \) the maximum value of \( i \) satisfying \( \delta_{q}, \text{tick'}(a) \rangle \) is determined by the smallest upper time bound of events defined at \( a \). Thus, it is intuitively clear that the smallest upper bound is the minimum of two values; one is the smallest value of \( u(\beta) \) subtracted by \( el(s', a', l(s)) \) for every event defined at both \( a' \) and \( a \) and the other is the smallest value of \( u(\gamma) \) for any event \( \gamma \) defined only at \( a \).

(b.2) The result is self-evident from the transition structure defined in Brandin and Wonham [4], that the EUTB of an event at a state is determined by the smallest upper time bound of events defined at the state.

The event set \( \Sigma \) is classified into three categories, the controllable events set \( \Sigma_{c} \) the uncontrollable events set \( \Sigma_{uc} \) and the forcible events set \( \Sigma_{for} \). The controllable events can be enabled or disabled by supervisors and the uncontrollable events should be permanently enabled. The forcible events can pre-empt the tick event by forcing action of supervisors. In this sense, the event tick is considered to be controllable [4].

For a timed string \( s = (a_{1}, \ldots, a_{n}, u_{a_{1}}, \ldots, u_{a_{n}}) \subseteq (\Sigma_{act} \times \mathbb{N} \times \mathbb{N})^{*} \), a trace of \( s \) is defined as \( \tau_{t}(s) = a_{1} \cdots a_{n} \), that is, the string of activity events in \( s \). The trace set \( \tau_{t}(L) \) of a timed language \( L \subseteq (\Sigma_{act} \times \mathbb{N} \times \mathbb{N})^{*} \) is the set of traces of all strings in \( L \). For a timed language specification \( K, s \in \tau_{t}(K) \), and \( a \in \delta_{act}(t_{0}, s) \), let

\[
B(K, s, a) = \min_{\gamma \in \Sigma_{act} \cap (\Sigma_{uc}(a) - \Sigma_{for}(a))} e_{a}(s, a, \gamma) \quad \text{if} \quad \Sigma_{uc} \cap (\Sigma_{uc}(a) - \Sigma_{for}(s)) \neq \emptyset, \quad \text{otherwise} \quad \infty.
\]

which is a minimum value of ELTBs of illegal uncontrollable events that are off the trace of \( K \) at the activity state \( a \) reaching via \( s \). This implies that if tick occurs \( B(K, s, a) \) times at the state \( a \), then an illegal uncontrollable event can occur and thereby the specification \( K \) cannot be achieved any longer. Thus, a proper control action should be taken before a timer reaches \( B(K, s, a) \).

For a timed language specification \( K \) and \( s \in \tau_{t}(K) \), the binary relation \( \subseteq_{K,s} \) for \( a_{1}, a_{2} \in \delta_{act}(a_{0}, s) \) is defined as \( a_{2} \subseteq_{K,s} a_{1} \iff \).

(i) \( \delta_{y} \in \Sigma_{uc} \cap (\Sigma_{uc}(a_{2}) - \Sigma_{for}(s)) \) such that \( e_{s}(s, a_{2}, \gamma) < B(K, s, a_{1}) \) and

(ii) \( a \in \Sigma_{uc}(a_{2}) \) and \( e(s, a_{2}, a_{1}) < B(K, s, a_{1}) \) for at least one \( a \in \Sigma_{uc}(s) \cap \Sigma_{uc} \) satisfying \( e(s, a_{1}, a_{1}) < B(K, s, a_{1}) \).

The relation \( a_{2} \subseteq_{K,s} a_{1} \) means that (i) the minimum ELTB of illegal uncontrollable events defined at \( a_{1} \) is less than or equal to that of illegal uncontrollable events defined at \( a_{2} \) (if those exist) and (ii) there exists at least one legal forcible event defined at both \( a_{1} \) and \( a_{2} \) with an ELTB at \( a_{2} \) less than the minimum ELTB of illegal uncontrollable events defined at \( a_{1} \). This notion is motivated from the following fact. When the string is observed, it is uncertain whether the system reaches the state \( a_{1} \) or \( a_{2} \).

With such an uncertainty, to avoid any occurrence of illegal uncontrollable events at every possible state, the occurrence of forcible events should be forced before a timer reaches the minimum ELTB of illegal uncontrollable events defined at \( a_{1} \). As a result, the forcible event is also forced when the state of the system is \( a_{2} \). In addition, we note that if \( a_{2} \subseteq_{K,s} a_{1} \) then, at the activity state \( a_{1} \) there always exist at least one illegal uncontrollable event and one legal forcible event.

Example 1: Let \( \delta_{act}(a_{0}, s) = \{a_{1}, a_{2}, a_{3}, a_{4}\}, \Sigma_{tr}(K) = \{a, \beta\}, \Sigma_{for} = \{a\}, \Sigma_{uc} \subseteq \Sigma_{uc}(a_{1}) = \Sigma_{uc}(a_{2}) = \Sigma_{uc}(a_{4}) = \{a, \gamma\} \) and \( \Sigma_{uc}(a_{3}) = \{\beta\} \). Also, as shown in Fig. 1, let \( e(s, a_{1}, a_{1}) = 1, e(s, a_{4}, a_{3}) = 3, e(s, a_{2}, a_{1}) = 2, e(s, a_{2}, a_{5}) = 4, e(s, a_{3}, a_{3}) = 3, e(s, a_{3}, a_{5}) = 5; e(s, a_{3}, \beta) = 1, e(s, a_{4}, \beta) = 3 \). Then, according to the definition, it holds that \( a_{2} \subseteq_{K,s} a_{1}, a_{3} \subseteq_{K,s} a_{2}, a_{1} \subseteq_{K,s} a_{3}, a_{5}, a_{3} \subseteq_{K,s} a_{1} \) and \( a_{4} \) has no relation with other states.

For a string \( s \in \tau_{t}(K), let \) we partition the activity states set \( \delta_{act}(a_{0}, s) \) into

\[
\delta_{act}(a_{0}, s) = g_{1} \cup g_{2} \cup \cdots \cup g_{4} \cup \emptyset \times \emptyset.
\]

where \( \emptyset \) is the disjoint union of the sets and

(i) \( g_{1} \) is the set of activity states such that for some \( a_{i} \in g_{1} \) and any \( a \in g_{1}, a \subseteq_{K,s} a_{i} \), where \( a_{i} \) is called a master of \( g_{1} \).

Fig. 1 ELTBs of all the events at states \( a_{1} - a_{4} \)
(ii) $fl$ is the set of activity states such that for any $a, a' \in fl$ ($a \neq a'$) and a master $a_t$ of $g_i, a' \nleq_{K, a} a, a \nleq_{K, a} a_t$ and $a, a' \nleq_{K, a}$.

(iii) if $a_t$ and $a_j$ are the masters of $g_i$ and $g_j$, respectively, then $a_t \nleq_{K, a} a_j$ and $a_j \nleq_{K, a} a_t$.

This partition is required to group some reachable states by a string, which is determined by forcible events to avoid illegal uncontrollable events with a minimum ELTB. At any state in $g_i$, at least one legal forcible event defined at $a_t$ (a master of $g_i$) can be forced before the minimum ELTB of illegal uncontrollable events defined at $a_t$. In other words, the forcing action determined in $a_t$ affects the occurrence of legal forcible events at the states in $g_i$. However, the forcing action determined in $g_i$ has no effect on the legal forcible events defined at the states of the other $g_j$'s and $fl$. Moreover, note that for some $a \in fl$ and some non-master $a' \in g_i$, it may hold that $a \nleq_{K, a} a'$ or $a' \nleq_{K, a} a$. In Example 1, we can partition $\delta_{act}(a, s)$ as follows: $\delta_{act}(a_0, s) = g_i \cup g_j \cup fl$, where $g_i = \{a_1, a_2\}, g_j = \{a_3\}$ and $fl = \{a_4\}$.

Next, for a legal event $a$ after a trace $s \in tr(K)$, that is, $\alpha \in tr(K)$, we define a minimum $el$ and a maximum $eu$ as follows:

$$\min_g eu(s, \alpha) := \min_{a \in g} (\min_{a \in fl} el(s, a, \alpha))$$

$$\min_f el(s, \alpha) := \min_{a \in fl} el(s, a, \alpha)$$

$$\min el(s, \alpha) := \min (\min_g eu(s, \alpha), \min_f el(s, \alpha))$$

If the event $\alpha$ is defined at one state in some $g_i$, then the event can occur after $Min(\alpha, el(s, \alpha))$ occurrences of tick. Otherwise, it can occur after $Min(\alpha, el(s, \alpha))$ occurrences of tick. Hence, after a trace $s \in tr(K)$, the activity event $\alpha$ on the trace of $K$ can occur after at least $Min(\alpha, el(s, \alpha))$ ticks. In Example 1, $Min(\epsilon, \alpha) = 1$, $Min(\alpha, el(s, \alpha))$ (undefined), $Min(\alpha, el(s, \beta))$ and $Min(\alpha, el(s, \beta)) = 1$. Hence, $Min(\alpha, el(s, \alpha)) = 1$ and $Min(\alpha, el(s, \beta)) = 1$. Also

$$Max_g eu(s, \alpha) := \max(B(K, s, a, -1))$$

$$Max_f eu(s, \alpha) := \max_{a \in fl} eu(s, a, \alpha)$$

$$Max eu(s, \alpha) := \max(Max_g eu(s, \alpha), Max_f eu(s, \alpha))$$

If the event $\alpha$ is defined at one state in some $g_i$, then the event must occur within $Max(\alpha, eu(s, \alpha))$ occurrences of tick to avoid illegal uncontrollable events. Otherwise, the event is permitted to occur within its possible maximum EUTB $Max(\alpha, eu(s, \alpha))$. Thus, after a trace $s \in tr(K)$, the activity event $\alpha$ on the trace of $K$ can occur after at most $Max(\alpha, eu(s, \alpha))$ ticks in the system $G$. In Example 1, $Max(\epsilon, eu(s, \alpha)) = 4$, $Max(\alpha, eu(s, \alpha)) = Min(\epsilon, eu(s, \alpha)) = 3$. Hence, $Max(\epsilon, eu(s, \alpha)) = 4$ and $Max(\alpha, eu(s, \alpha)) = 3$. Also, let $Max(\alpha, el(s, \alpha)) := max_{s \in L(G, s)} Min(\alpha, el(s, \alpha))$. For some finite interval $M \leq 2^{2N}$, let $Low(M)$ and $Upp(M)$ be the lower bound and the upper bound of $M$, respectively. For instance, if $M = \{2, 3, 4\}$, then $Low(M) = 2$ and $Upp(M) = 4$. We define a supervisor $S = (V, I, \tau)$ as follows: $V: L(G, s) \rightarrow 2^{2N}$ (events set to be enabled or forced), $I: L(G, s) \times \Sigma_{act} \rightarrow 2^N$ (time interval) and $\tau: L(G, s) \times \Sigma_{act} \rightarrow N$ (timer) such that

(i) $V(s) \supseteq \Sigma_{L(G, s)} \cup \Sigma_{fl}$ for any $s \in L(G, s)$, where any forcible event $\alpha \in (V(s) \cap \Sigma_{for})$ is forced only at the time $Upp(s, \alpha)$ and other events in $V(s)$ are enabled.

(ii) $I(s, \alpha)$ is a finite interval satisfying:

(a) in the case of $V(s) \cap \Sigma_{for} \neq \emptyset$

$Upp(I(s, \alpha)) \leq \max eu(s, \alpha) \land \min eu(s, \alpha)$

Low(I(s, \alpha)) \geq \min eu(s, \alpha), \quad \text{if } \alpha \in \Sigma_c$

Upp(I(s, \alpha)) \leq \max eu(s, \alpha) \land \min eu(s, \alpha), \quad \text{otherwise}$

(b) in the case of $V(s) \cap \Sigma_{for} = \emptyset$

$Upp(I(s, \alpha)) = \max eu(s, \alpha) \land \min eu(s, \alpha)$

Low(I(s, \alpha)) \geq \min eu(s, \alpha), \quad \text{if } \alpha \in \Sigma_c$

Upp(I(s, \alpha)) = \max eu(s, \alpha) \land \min eu(s, \alpha), \quad \text{otherwise}$

(iii) $\tau(\epsilon, \alpha) := 0$ for all $\alpha \in \Sigma_{L(G, s)}$.

The statement (i) means that all uncontrollable events belong to $V(s)$ because they are permanently enabled. The interval $I(s, \alpha)$ indicates that the event $\alpha$ is permitted to occur after Low(I(s, \alpha)) ticks and should occur before Upp(I(s, \alpha)) ticks in the controlled system. Note that Low(I(s, \alpha)) of a controllable event $\alpha$ may be larger than Min(\alpha, eu(s, \alpha)) because the timing of its occurrence can be controlled by disablement of the supervisor. In addition, when there exist forcible events in $V(s)$, Upp(I(s, \alpha)) may be smaller than Max(\alpha, eu(s, \alpha)) because its occurrence timing can be controlled by the forcing action. The timer $\tau(\epsilon, \alpha)$ records the number of occurrences of tick for each event $\alpha$ after the occurrence of $le(\alpha)$. When the timer has a value within $I(s, \alpha)$, the event $\alpha$ becomes enabled by the supervisor. Further, if the event is forcible and the timer value reaches $Upp(I(s, \alpha))$, then the supervisor forces the event.

A supervised system denoted by $S/G$ means the nondeterministic real-time DES $G$ under the control of a supervisor $S$. In other words, $S/G$ is a subpart of $G$, which is formed by preventing the occurrence of some events in $G$ through the supervisor’s action. As a consequence, the behaviour of $S/G$ becomes a subset of that of $G$ satisfying a given desired specification. The behaviour of $S/G$ denoted by $L(S/G)$ is inductively defined as follows: (i) $\epsilon \in L(S/G)$, (ii) for $s \in \Sigma^*$ and $\alpha \in \Sigma$, suppose that $s \in L(S/G)$ and $s\alpha \in L(G)$, and let $P_{act}(s) = s\alpha$, then

(i) if $\alpha \in \Sigma_{act}, \quad \alpha \in L(G, s) \text{ and } \tau(s\alpha, \alpha) \in L(G, s)$, then $\alpha \in L(S/G)$ and $\tau(s\alpha, \alpha, \beta) = 0$ for all $\beta \in \Sigma_{L(G, s)}$.

(ii) if $\alpha \in L(S/G)$ and $\tau(s\alpha, \beta) < Upp(I(s\alpha, \beta))$ for some $\beta \in \Sigma_{L(G, s)}$, then $s\alpha \in L(S/G)$, $\tau(s\alpha, \beta) = \tau(s\alpha, \beta) + 1$ and $\tau(s\alpha, \gamma) = 0$ for all $\gamma \in \Sigma_{L(G, s)}$ s.t. $\tau(s\alpha, \gamma) = Upp(I(s\alpha, \gamma))$.

The statement (i) means that the event $\alpha$ in $V(s\alpha)$ can occur in $S/G$ when its timer value lies within $I(s\alpha, \alpha)$. In addition, when the event has occurred, the timer values of all the events become reset to 0. The statement (ii) means that tick can occur in $S/G$ if there exists an event $\beta$ of which the timer value does not reach the upper bound of $I(s\alpha, \beta)$. Then the timer value of $\beta$ increases by 1 as one unit of time has elapsed. For the event $\gamma$ of which the timer value reaches its upper bound of $I(s\alpha, \gamma)$, its timer becomes reset to 0 because $\gamma$ is not defined at the state where tick has occurred. In particular, note that if there exist $\gamma \in \Sigma \cap \Sigma_{L(G, s)}$ and $\alpha \in \Sigma_{for} \cap \Sigma_{L(G, s)}$ such that $\gamma, \alpha \in L(G, s), \alpha \in \delta_{act}(a_0, s_0)$ and $Upp(I(s\alpha, \alpha)) < \epsilon(s\alpha, \alpha, \gamma)$, then the supervisor assigns the event $\alpha$ when the timer value $\tau(s\alpha, \alpha)$ becomes equal to $Upp(I(s\alpha, \alpha))$. 

422

3 Existence conditions of supervisors

To formulate a supervisory control problem for bounded time constraints, let us define a language-to-timed language transformation $T : 2^{L(L)} \rightarrow 2^{\Sigma_{act} \times N \times N}$ inductively according to: (i) $e \in T(L)$, (ii) for $s_i \in (\Sigma_{act} \times N \times N)^*$ and $(a, t_i, u_i) \in T(L)$ and only if (a) $s_i \in T(L)$ (b) for all $s \in P_{act}^1(tr(s_i)) \cap L$ satisfying le($s$) $\neq$ tick, $t_i$ and $u_i$ are the minimum value and the maximum value of $i$, respectively, such that $s$ tick$^i \alpha \in L$.

In (b), ‘le($s$) $\neq$ tick’ is required because the minimum and maximum values denote the numbers of tick occurrences after the last event activity as its last event. In other words, for a real-time DES $G$, $s_i(a, t_i, u_i) \in T(L(G))$ means that after a string with an activity event as its last event and the trace $tr(s_i)$ as its $P_{act}$ projection, the event $\alpha$ occurs after at least $t_i$ ticks and at most $u_i$ ticks in $G$. Through this transformation, the behaviour of a real-time DES modelled by a timed transition model can be described as a timed language.

In this paper, we consider the following control problem. Given a closed timed language specification $K$ for a non-deterministic real-time DES $G$, the existence conditions of a supervisor $S$ such that $T(L(S/G)) = K$. Now, let us introduce the notions of trace controllability and time controllability of a given timed language specification $K$ and show that they are both necessary and sufficient for the existence of a supervisor achieving the specification. First, recall that for a string $s \in tr(K)$, the activity states set $\delta_{act}(a, s)$ is partitioned into $\delta_{act}(a, s) = g_1 \cup g_2 \cup \cdots \cup g_n \cup fl$. Then, in order to introduce the trace controllability, let us define the following two activity event sets

$$G(K, s) := \bigcup_{g_i \in g} \left( \{a \in \Sigma_{act(K)}(s) | el(s, a, a) < B(K, s, a)\} \right)$$

where $a_i$ is a master of $g_i$, and

$$F(K, s) := \bigcup_{a \in fl} \left( \{a \in \Sigma_{act(K)}(s) | el(s, a, a) < eu(s, a, a)\} \right)$$

$G(K, s)$ is the set of legal events after $s$, the occurrence of which can be guaranteed by proper forcing actions before the occurrence of any illegal uncontrollable event. $F(K, s)$ is the set of legal events after $s$, the occurrence of which can be guaranteed without consideration of any forcing action to avoid illegal uncontrollable events.

Definition 2: A timed language $K$ is trace controllable with respect to a non-deterministic real-time DES $G$ if, for all $s \in tr(K)$

(i) $\Sigma_{act(K)}(s) = G(K, s) \cup F(K, s)$

(ii) for any $a \in \delta_{act}(a, s)$ s.t. $(\Sigma_{act} - \Sigma_{act(K)}(s)) \cap \Sigma_{uc} \neq \emptyset$, there exists $a \in \Sigma_{act(K)}(s) \cap \Sigma_{for}$ s.t. $el(s, a, a) < B(K, s, a)$.

The statement (i) means that any legal event after the trace $s$ should have its ELTB less than a minimum ELTB of illegal uncontrollable events that can occur after $s$. The statement (ii) means that for any state reachable by a trace, there should exist at least one legal forcible event with its ELTB less than a minimum ELTB of illegal uncontrollable events defined at the state. In other words, this trace controllability of a timed language specification makes it possible to design a supervisor that can exactly achieve the trace set of the specification.

For $s_i \in (\Sigma_{act} \times N \times N)^*$, the set of prefixes of $s_i$, defined as $pr(s_i) := \{v_i \in (\Sigma_{act} \times N \times N)^* | v_i^1 = s_i, \text{ for some } u_i \in (\Sigma_{act} \times N \times N)^*\}$, and the prefix-closure $pl(L)$ of a timed language $L$, is the set of prefixes of all strings in $L$. $L_i$ is said to be prefix-closed (or closed) if $pr(L_i) = L_i$. Moreover, let $\Sigma_{act}(s_i) := \{a \in \Sigma_{act} \times N \times N | a_i \in pr(L_i)\}$. Now, we define the time-controllability of a timed language as follows:

Definition 3: A timed language $K$ is time-controllable with respect to a non-deterministic real-time DES $G$ if, for all $s \in pr(K)$ and $(a, t_i, u_i) \in \Sigma_{act(K)}(s_i)$

(i) $t_i \geq \min el(s, a, a) \quad \text{if} \quad a \in \Sigma_{c}$

$t_i = \min el(s, a, a) \quad \text{otherwise}$

(ii) $u_i = \max el(s, a, a) \quad \text{if} \quad \Sigma_{act(K)}(s_i) \cap \Sigma_{for} = \emptyset$

$\max el(tr(K), s) \leq u_i \\ \leq \max eu(s, a, a) \quad \text{otherwise}$

where $s = tr(s_i)$.

Time controllability of $K$ implies that a lower bound $t_i$ of a controllable event $a$ on the trace of $K$ can be exactly achieved by a proper disablement action, and if there are forcible events on the trace of $K$ and an uncontrollable event off the trace of $K$ after $s$, the proper forcing action of supervisors can achieve the upper bound $u_i$ between $\max el(tr(K), s)$ and $\max eu(s, a)$. In this manner, the time controllability makes it possible to exactly achieve time constraints imposed on a specification.

The following theorem shows that the trace controllability and time controllability are the necessary and sufficient conditions for the existence of a supervisor that can satisfy a given bounded time constraint represented by the timed language $K$.

Theorem 1: For a closed timed language specification $K$, there exists a supervisor $S$ for a non-deterministic real-time DES $G$ such that $T(L(S/G)) = K$ if and only if $K$ is trace controllable and time controllable with respect to $G$.

Proof: (If) Consider a supervisor $S = (V, I, \tau)$ defined as follows: for all $s_i \in pr(K)$ and $(a, t_i, u_i) \in \Sigma_{act(K)}(s_i)$

$$I(tr(s_i)) := \Sigma_{act(K)}(tr(s_i)) \cap (\Sigma_{uc} \cup \Sigma_{for})$$

$$\tau(\epsilon, s) := \{m \in N | \min el(s, a, a) \leq m \leq u_i\}$$

In addition, for any $p \in \Sigma_{uc}(s_i) - \Sigma_{act(K)}(s_i)$, let $I(s, \rho) := \{m \in N | \min el(s, a, a) \leq m \leq \max eu(s, a)\}$. To prove $T(L(S/G)) = K$, assume for all $s_i \in (\Sigma_{act} \times N \times N)^*$ that $s_j \in (\Sigma_{uc} \cup \Sigma_{for})$, then we must show that for all $(a, t_i, u_i) \in (\Sigma_{act} \times N \times N), s(a, t_i, u_i) \in T(L(S/G)) \Rightarrow s(a, t_i, u_i) \in T(L(S/G))$.

(i) $(\Rightarrow)$ Let $s(a, t_i, u_i) \in T(L(S/G))$ and $tr(s) = s$. Then, as $a \in P_{act}(L(S/G), a) \in V(s)$. Suppose that $a \in \Sigma_{uc}(s)$. Then, (1), it holds that $\beta \in \Sigma_{act(K)}(s) \cap \Sigma_{uc}$. By trace controllability of $K$, for any $a \in \delta_{act}(a, s)$ satisfying $a \in \Sigma_{uc}(s, a)$, there exists $\beta \in \Sigma_{act(K)}(s) \cap \Sigma_{for}$. s.t. $el(s, a, a) < B(K, s, a)$.

Also, by time controllability of $K$...
K, it holds that $Upp(l(s, \beta)) < el(s, a, a)$. Then, as the supervisor $S$ forces the event $\beta$ when $\tau(s, \beta) < el(s, a, a)$, we see $P_{act}(L(S)/G)$. This is a contradiction. Thus, $\alpha \in \Sigma_{tr}(\tau_{st}(s))$. In addition, according to the definition of $L(S)/G$, $t_{le} = Low(l(s, \alpha))$ and $t_{ue} = Upp(l(s, \alpha))$. Then, $(\alpha, t_{le}, t_{ue}) \in \Sigma_{pr}(l(s))$. But $\tau(s, t_{le}, t_{ue}) \in K$. (ii) ($\Leftarrow$) Let $s'(\alpha, t_{le}, t_{ue}) \in K$ and $tr(s) = s$. Then, $\alpha \in F(s)$ and by trace controllability of $K$, $\alpha \in G(K, s)$. Therefore from (i) and (ii), we conclude that $T(L(S)/G) = K$.

(Only if) Note that $V(s) = \Sigma_{tr}(s)$ for any $s \in tr(K)$ because $T(L(S)/G) = K$. First we show that $K$ is trace controllable with respect to $G$. Generally, it holds that $\Sigma_{tr}(s) \supseteq G(K, s) \cup F(K, s)$. From this, suppose that $K$ is not trace controllable, that is, there exists $\alpha \in \Sigma_{act}$ s.t. $\alpha \in \Sigma_{tr}(s) - (G(K, s) \cup F(K, s))$. Then, for any $\alpha \in \delta_{act}(a_0, s)$,

(a) $el(s, a, a) \geq B(K, s, a)$ or (b) $el(s, a, a) > eu(s, a, a)$

In case (a), as the supervisor $S$ forces the forcible events before the time $B(K, s, a)$, the event $\alpha$ never occurs. In case (b), the event $\alpha$ also does not occur in the uncontrolled system after the string $s$. Thus, $s \not\in tr(L(S)/G)$, which contradicts $T(L(S)/G) = K$. Next, suppose that $\Sigma_{ue} = \Sigma_{tr}(s) - \Sigma_{act}(s)$, it is true that $\Sigma_{ue} \cap (\Sigma_{G}(a) - \Sigma_{tr}(s)) = \emptyset$. Then, $s(a, t_{le}, t_{ue}) \in G(K, s)$ with time controllability of $K$. Also, the time controllability of $K$ guarantees that $t_{le}$ and $t_{ue}$ are the minimum and maximum values of $i$, respectively, such that $i$'s tick $\alpha \in L(S)/G$, where $P_{act}(s'') = s$ and $el(s'') \neq tick$. Therefore from (i) and (ii), we conclude that $T(L(S)/G) = K$.

To investigate a non-blocking supervisory control problem, we consider the following notions of non-blockingness.

**Definition 4:** For a closed timed language $K$, the closed language $tr(K)$ is non-blocking with respect to $G_{act}$ if, for all $s \in tr(K)$ and $a \in \delta_{act}(a_0, s)$, there exists $t \in \Sigma_{act}$ such that $s_{t} \in tr(K)$ and $\delta_{act}(a, t) \cap A_{m} \neq \emptyset$.

**Definition 5:** A supervisor $S$ is non-blocking for a non-deterministic real-time DES $G$ if, for all $s \in L(S)/G$ and $q \in \delta(q_0, s)$, there exists $t \in \Sigma_{act}$ such that $s_{t} \in L(S)/G$ and $\delta(q, t) \cap Q_{m} \neq \emptyset$.

We note that the non-blocking properties of the trace set of $K$ and the supervisor $S$ are given in terms of the activity model $G_{act}$ and the timed transition model $G$, respectively. Then, for a supervisor satisfying a given timed language specification, the following theorem shows that the non-blockingness of the trace set of the specification is the necessary and sufficient condition for the supervisor to be non-blocking for $G$.

**Theorem 2:** For a closed timed language specification $K$ for a non-deterministic real-time DES $G$ and a supervisor $S$ with $T(L(S)/G) = K$, $S$ is non-blocking for $G$ if and only if $tr(K)$ is non-blocking with respect to $G_{act}$.

Proof: (If) Consider a supervisor $S = (V, I, \tau)$ defined at the 'If' part in the proof of Theorem 1. Let $s \in tr(T(L(S)/G))$ and $a \in \delta_{act}(a_0, s)$. From $T(L(S)/G) = K$, $s \in tr(K)$, and there exists $s' \in L(S)/G$ s.t. $P_{act}(s'') = s$ and $q = (a, \ldots) \in \delta(q_0, s)$. Then

$$\exists v \in \Sigma_{act} \text{ s.t. } sv \in tr(K) \text{ and } \delta_{act}(a, v) \in A_{m}$$

by non-blockingness of $K$

$$\Rightarrow sv \in tr(T(L(S)/G))$$

by $T(L(S)/G) = K$

$$\Rightarrow sv' \in \Sigma_{act} \text{ s.t. } sv' \in L(S)/G$$

and $\delta(q, v') \cap Q_{m} \neq \emptyset$,

where $P_{act}(s'') = sv$ and $q = (a, \ldots)$. Therefore we conclude that $S$ is non-blocking for $G$.

(Only if) For $s \in \Sigma_{act}$, $s \in tr(K)$ implies that

$$s \in tr(T(L(S)/G)) \text{ by } T(L(S)/G) = K$$

$$\Rightarrow \exists s' \in L(S)/G \text{ s.t. } P_{act}(s') = s$$

$$\Rightarrow \text{ for any } q = (a, \ldots) \in \delta(q_0, s')$$

as $S$ is non-blocking for $G$.

$$\exists v' \in \Sigma_{act} \text{ s.t. } sv' \in L(S)/G$$

and $\delta(q, v') \cap Q_{m} \neq \emptyset$.

Therefore we conclude that $tr(K)$ is non-blocking with respect to $G_{act}$. □

4 Example

Consider an assembly work station performing peg-in-hole linking operations and welding operations for the pegs and the holes shown in Fig. 2a. The work station has three input pegs: peg 1, peg 2 and peg 3. After a peg and a hole are linked, they are fixed through a welding operation. Let us assume that there is no sensor that can distinguish...
between the incoming pegs. Then, the arrival of a peg in the work station is labelled as the identical event $a$. When a peg arrives, hole $i$ in a hole storage can be selected for a linking operation (event $b_i$), where $i = 1, 2, 3$. If peg 1, peg 2 and peg 3 are linked with hole 1, hole 2 and hole 3, respectively, then the welding operations for the linked parts are assumed to be successfully completed (event $c$). In addition, it is assumed that (i) peg 1 is not linked with hole 2 and hole 3, (ii) peg 2 is not linked with hole 3, (iii) if hole 1 is linked with peg 2 or peg 3 and hole 2 is linked with peg 3, then loose linking of parts causes a failed welding operation (event $d$) because of abnormal welding situations such as the spattering fault when a tip of a torch nozzle is choked with the molten material called weld spatter. Then, the process must be stopped to clear the weld spatter in the tip. On the basis of the foregoing descriptions of the system’s behaviour, the overall logical behaviour of the system can be modelled by $G_{act}$ as shown in Fig. 2b, where a self-loop by $b_2$ and $b_3$ at $a_1$ represents the fact that the input peg 1 is not linked with hole 2 and hole 3. Moreover, a self-loop by $b_3$ at $a_2$ represents the fact that the input peg 2 is not linked with hole 3.

The event set $\Sigma$ is classified into $\Sigma = \{b_1, b_2, b_3, \text{tick}\}$, $\Sigma_{for} = \{b_1, b_2, b_3\}$ and $\Sigma_{uc} = \Sigma - \Sigma_{for}$. The lower time bound and the upper time bound of each event are given as follows: $(a, 1, 2)$, $(b_1, 1, \infty)$, $(b_2, 1, \infty)$, $(b_3, 1, \infty)$, $(c, 2, 5)$ and $(d, 2, \infty)$, where $(a) = 1$ and $(u(a)) = 2$.

Consider a specification $K_1 = pr(a, 1, 2 (b_1 1, 1, 2 (c, 2, 4))$. Then, $tr(K_1) = pr(ab_1c)$. For $s = ab_1$, $\Sigma_{pr(K_1)}(s) = \{c\}$ and $\Sigma_{pr(K_1)}(s) \cap \Sigma_{for} = \emptyset$. Also, $\delta_{act}(a, s) = \{a_1, a_0, a_1\} = \emptyset$ and hence $G(K_1, s) = \emptyset$, $F(K_1, s) = \{c\}$. Thus, $\Sigma_{uc(K_1)}(s) = G(K_1, s) \cup F(K_1, s)$, however, as there does not exist $\sigma \in \Sigma_{pr(K_1)}(s) \cap \Sigma_{uc}$ such that $e(s, \sigma, c) < B(K_1, s, a_0) = e(s, a_0, d) = 2$, $K_1$ is not trace controllable. In other words, at the state $a_0$ reachable by the string $ab_1$ in $tr(K_1)$, there are no legal forcible events with their ELTBs less than the minimum ELTB of the illegal uncontrollable event $d$, $B(K_1, s, a_0) = 2$. In addition, $K_1$ is not time controllable by the following reason. For $s = (a, 1, 2 (b_1 1, 2) \in pr(K_1))$, $(c, 2, 4) \in \Sigma_{pr(K_1)}(s)$. Then, $\min e(s, c) = 2$ but $\max e(s, c) = 5$, where $s = tr(s) = ab_1$. In other words, after the string $ab_1$ there is no legal forcible event that can guarantee the occurrence of the event $c$ within four ticks, and thereby the occurrence of $c$ after five ticks cannot be avoided. Therefore there does not exist a supervisor $S$ such that $T(S(G)) = K_1$. If $K_1 = pr(a, 1, 2 (b_1 1, 2 (c, 2, 5))$, then it is time controllable because the occurrence of $c$ after $ab_1$ can be guaranteed within five ticks. However, it is not trace controllable.

Consider another specification

$$K_2 = pr(a, 1, 2 (b_3 1, 2 (c, 2, 1, 2) + (b_2, 1, 2 (b_1 1, 2) (c, 2, 5)))$$

We note that $K_2$ is trace controllable because no illegal string in $tr(K_2)$ leads to the state $a_0$ or $a_7$ at which the illegal uncontrollable event $d$ can occur. $K_2$ is also time controllable because the time bounds of $K_2$ can be met, for example, the occurrence of $b_i$ $(i = 1, 2, 3)$ can be guaranteed within two ticks by forcing those when $tick$ occurs twice. Note that if the time bound of $c$ in $K_2$ were not 5 but 4, then $K_2$ would not be time controllable any more (although it is still trace controllable). Furthermore, $tr(K_2)$ is non-blocking with respect to $G_{act}$ for any string in $tr(K_2)$ because there exist other strings leading to the marked states after the string in $G_{act}$. Therefore according to Theorem 2, there exists a non-blocking supervisor $S$ such that $T(L(S(G))) = K_2$ in this case.

On the basis of $K_2$, the supervisor $S$ can be designed as follows: for all $s_i \in pr(K_2)$ and $(\sigma, t_{\sigma}, \tau_{\sigma}) \in \Sigma_{pr(K_2)}(s_i)$, $T(tr(s_i)) = \Sigma_{pr(K_2)}(s_i) \cup (\Sigma_{uc(K_2)}(s_i) \cap \Sigma_{uc})$, $l(tr(s_i), \sigma) = \{m \in N \mid t_{\sigma} \leq m \leq t_{uc}\}$ and $\tau(\sigma, \rho) = 0$ for all $\rho \in \Sigma_{uc(K_2)}(s_i)$. In addition, for any $\rho \in \Sigma_{uc(K_2)}(s_i)$, $l(tr(s_i), \sigma) = \{m \in N \mid \min e(s, \rho) \leq m \leq \max e(s, \rho)\}$. For example, for $s_i = (a, 1, 2 (b_3, 1, 2, 1), V(tr(s_i))) = V(ab_3) = \{c, b_2, r, a_3, c = 2, 5\}, l(ab_3, b_2) = \{1, 2\}$. After the occurrence of a string $s \in L(G)$ such that $P_{uc(s)} = ab_3$ and let $s \neq tick$, the supervisory system generates the following strings: $t, b_3, t', r, b_2, t, r, t', r, t', r, l, r, l$, where $t$ denotes the event $tick$. When the successive two ticks occur, the supervisor chooses a forcing action via the forcible event $b_2$. As a result, if the activity state is $a_0$ after $ab_3$, then the event $c$ occurs within its time constraints 2 and 5 independent of the forcing action. If the activity state is $a_1$, then the forcing action makes the system move the activity state $a_0$, and if the activity state is $a_1$, then the supervisory system remains in the same activity state $a_1$. The supervisor $S$ implements the following logical control strategy: when a peg arrives, the supervisor requests that hole 3 should be selected for linking with the peg. If the peg is peg 3, the assembly operation is successfully completed. However, if the peg is not peg 3, hole 3 is not linked with the peg. Next, the supervisor requests that hole 2 should be selected. If the peg is peg 2, the assembly operation is successfully completed. However, if the peg is not peg 2, hole 2 is not linked with the peg. Finally, the supervisor requests that hole 1 should be selected. As the peg must be peg 1, the assembly operation is successfully completed.

5 Conclusions

In this paper, we have investigated the supervisory control of non-deterministic real-time DESs represented by non-deterministic timed transition models. On the basis of the activity models and the time bounds, we have presented the existence conditions of a non-blocking supervisor to meet a given bounded time constraint represented by a timed language specification.

6 Acknowledgments

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 21C Frontier Microbial Genomics and Application Center Program, Ministry of Science and Technology (Grant MG05-0204-3-0), Republic of Korea.
References


