COMPUTATION OF CLOSED, CONTROLLABLE, AND WEAKLY OBSERVABLE SUBLANGUAGES FOR TIMED DISCRETE EVENT SYSTEMS

Shigemasa Takai [*] **Toshimitsu Ushio** [**]

[*] *Dept. of Electronics and Information Science,*
Kyoto Institute of Technology,
Matsugasaki, Sakyo-ku, Kyoto 606-8585, Japan
E-mail: takai@dj.kit.ac.jp
[**] *Dept. of Systems Innovation, Osaka University,*
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
E-mail: ushio@sys.es.osaka-u.ac.jp

Abstract: Brandin and Wonham have developed a supervisory control framework for timed discrete event systems (TDESs). Lin and Wonham have extended this framework to the partial observation case. In our previous work, we have defined a new class of supervisors for TDESs under partial observation. We have introduced a notion of weak observability of languages, and proved that $L_m(G)$-closure, controllability, and weak observability are necessary and sufficient conditions for the existence of our nonblocking supervisor. These existence conditions of our supervisor are weaker than those of the Lin and Wonham's supervisor. In this paper, we study computation of a closed, controllable, and weakly observable sublanguage of a given closed regular language. We present an algorithm for computing such a sublanguage larger than the supremal closed, controllable, and normal sublanguage. *Copyright© 2005 IFAC*

Keywords: Timed discrete event system; Supervisory control; Controllability; Weak observability; Forcible event

1. INTRODUCTION

In order to deal with temporal specifications in the supervisory control theory (Ramadge & Wonham, 1987), timing information has to be introduced into discrete event system modeling. Several modeling frameworks for *timed* discrete event systems (TDESs) have been proposed. The TDES model proposed by Brandin and Wonham (1994) is a *discrete-time* model where the time advances discretely when the event *tick* that represents the passage of one time unit occurs. In this TDES model, the behavior of the system is described by a formal language and simply manipulated by operations on automata. This is the reason why we use the TDES model of Brandin and Wonham.

Brandin and Wonham (1994) have extended the framework of Ramadge and Wonham (1987) to TDESs. Lin and Wonham (1995) have studied supervisory control of partially observed TDESs. The Lin and Wonham's supervisor decides to enable or disable the event *tick* as well as controllable events. The supervisor can disable *tick* by forcing an enabled *forcible* event to occur, but does not specify explicitly which enabled forcible event is forced to occur. In our previous work (Takai & Ushio, 2004), we have modified the supervisor such that it selects forcible events to be forced to occur explicitly instead of deciding to enable or disable *tick*. Whether *tick* is enabled or disabled depends on the existence of an enabled forcible event forced by the supervisor. More precisely, if there is no enabled forcible event forced by the supervisor at the current state, then *tick* is enabled; otherwise *tick* is disabled. This implies that, by suitably selecting events to be forced to occur, our modified supervisor can control *tick* more flexibly than the Lin and Wonham's supervisor. In fact, this modification improves the performance of a supervisor under partial observation in the sense that our modified supervisor can achieve a larger class of languages than the Lin and Wonham's supervisor (Takai & Ushio, 2004). In order to characterize the existence of our supervisor, we have introduced a notion of *weak observability* for TDESs (Takai & Ushio, 2004). The weak observability condition together with the $L_m(G)$-closure and controllability conditions is a necessary and sufficient condition for the existence of our nonblocking supervisor.

In this paper, we study computation of a closed, controllable, and weakly observable sublanguage of a given closed regular language. This problem arises when a closed regular language given as a specification does not satisfy the existence conditions of a supervisor. It is desirable that such a sublanguage is as large as possible. However, the supremal closed, controllable, and weakly observable sublanguage does not exist in general. Lin and Wonham (1995) have synthesized a supervisor that achieves a sublanguage larger than the supremal closed, controllable, and normal sublanguage under the assumption that a given language is coherent. We present an algorithm for computing such a closed, controllable, and weakly observable sublanguage without imposing this assumption.

2. PRELIMINARIES

2.1 Timed Discrete Event Systems

The TDES defined by Brandin and Wonham (1994) is a finite automaton[1] $G = (Q, \Sigma, \delta, q_0)$, where Q is the finite set of states, Σ is the finite set of events, a partial function $\delta : \Sigma \times Q \to Q$ is the transition function, and $q_0 \in Q$ is the initial state. The event set is partitioned into two subsets as $\Sigma = \Sigma_{act} \cup \{tick\}$. Σ_{act} corresponds to the usual event set in untimed systems. The additional event *tick* represents the passage of one time unit.

Let Σ^* be the set of all finite strings of elements in Σ, including the empty string ε. The function δ can be generalized to $\delta : \Sigma^* \times Q \to Q$ in the natural way. The generated language $L(G)$ of G is defined by $L(G) = \{s \in \Sigma^* | \delta(s, q_0)!\}$. Let $K \subseteq \Sigma^*$ be a language. We denote the set of all prefixes of strings in K by $\overline{K}$. We also denote $\overline{\{s\}}$ by $\overline{s}$ for each $s \in \Sigma^*$. K is said to be (prefix-)closed if $K = \overline{K}$. We define a map $E_K : \Sigma^* \to 2^{\Sigma}$ by $E_K(s) = \{\sigma \in \Sigma | s\sigma \in \overline{K}\}$.

As in the untimed supervisory control framework, the set Σ_{act} is partitioned into two subsets Σ_c and Σ_{uc} of controllable and uncontrollable events, respectively. An event $\sigma \in \Sigma_{act}$ that can preempt the event *tick* is called a *forcible* event (Brandin & Wonham, 1994). The set of forcible events is denoted by Σ_{for}. By forcing an enabled event in Σ_{for} to occur, we can disable the event *tick*. Let $\Sigma_o \subseteq \Sigma$ be the set of *observable* events and $P : \Sigma^* \to \Sigma_o^*$ be the natural projection map (Lin & Wonham, 1995).

2.2 Strict Subautomaton

We consider any automata $G_A = (Q_A, \Sigma, \delta_A, q_{A0})$ and $G_B = (Q_B, \Sigma, \delta_B, q_{B0})$. We say that G_A is a subautomaton of G_B if the following two conditions hold (Cho & Marcus, 1989):

C1) $Q_A \subseteq Q_B$ and $q_{A0} = q_{B0}$.
C2) For any $q \in Q_A$ and any $\sigma \in \Sigma$, if $\delta_A(\sigma, q)!$ then $\delta_A(\sigma, q) = \delta_B(\sigma, q)$.

By the conditions C1) and C2), we have $L(G_A) \subseteq L(G_B)$. An automaton G_A is said to be a *strict subautomaton* of G_B if the following condition C3) holds in addition to C1) and C2).

C3) For any $s \in L(G_B) - L(G_A)$, there exists $s' \in \overline{s}$ such that $\delta_B(s', q_{B0}) \notin Q_A$.

We regard the empty automaton Φ as a strict subautomaton of any automaton G.

2.3 Normal Automaton

For an automaton $G = (Q, \Sigma, \delta, q_0)$, we define a subset $Q(t) \subseteq Q$ for each $t \in P(L(G))$ as $Q(t) = \{q \in Q | \exists s \in P^{-1}(t);\ \delta(s, q_0) = q\}$. An automaton G is said to be normal if, for any $t, t' \in P(L(G))$, $Q(t) \cap Q(t') = \emptyset$ holds whenever $Q(t) \neq Q(t')$ (Cho & Marcus, 1989).

We define an automaton $G^P = (X, \Sigma_o, \xi, x_0)$, called an *observer* of G, where $X = 2^Q - \{\emptyset\}$, $x_0 = Q(\varepsilon)$, and $\xi : \Sigma_o \times X \to X$ is defined by

$$\xi(\sigma_o, x) = \begin{cases} \{q \in Q | \exists s \in P^{-1}(\sigma_o),\ q' \in x; \\ \quad \delta(s, q') = q\},\ \text{if it is nonempty}, \\ \text{undefined, otherwise}. \end{cases}$$

Since $L(G^P) = P(L(G))$ and $\xi(t, x_0) = Q(t)$ for each $t \in L(G^P)$ (Cieslak, Desclaux, Fawaz, &

[1] In this paper, we identify an automaton with its accessible part.

Varaiya, 1988), G is normal if and only if the state space of the observer G^P is a partition of the state space of G.

3. SUPERVISORY CONTROL

In this section, we briefly review the result of Takai and Ushio (2004). A supervisor γ is defined by a map $\gamma : P(L(G)) \rightarrow 2^{\Sigma_{act}} \times 2^{\Sigma_{for}}$ such that, for any $t \in P(L(G))$, $\gamma(t) := (\gamma_1(t), \gamma_2(t))$ satisfies the following two conditions:

(1) $\Sigma_{uc} \subseteq \gamma_1(t)$.
(2) $\gamma_2(t) \subseteq \gamma_1(t) \cap \Sigma_{for}$.

In this definition of a supervisor γ, $\gamma_1(t)$ is the set of events to be permitted to occur. Also, $\gamma_2(t)$ denotes the set of events to be forced to occur. Then, the first condition implies that γ never disables uncontrollable events. The second condition implies that a forcible event is forced to occur only if it is permitted to occur. We inductively define the generated language $L(G, \gamma)$ under the control action of γ as follows:

- $\varepsilon \in L(G, \gamma)$.
- $(\forall s \in L(G, \gamma)$ and $\forall \sigma \in \Sigma)$
  if $\sigma \in \Sigma_{act}$,
  $s\sigma \in L(G, \gamma) \Leftrightarrow \sigma \in E_{L(G)}(s) \cap \gamma_1(P(s))$.
  if $\sigma = tick$,
  $s\sigma \in L(G, \gamma) \Leftrightarrow$
  $[\sigma \in E_{L(G)}(s)] \wedge [E_{L(G)}(s) \cap \gamma_2(P(s)) = \emptyset]$.

Let $t \in P(L(G))$ be any observable event string. Recall that the Lin and Wonham's supervisor (Lin & Wonham, 1995) decides to enable or disable the event *tick*. So, for any $s \in L(G, \gamma) \cap P^{-1}(t)$ with $tick \in E_{L(G)}(s)$, the same control action is assigned to *tick*. On the other hand, our supervisor specifies which forcible events are forced to occur instead of deciding to enable or disable *tick*. Whether *tick* is enabled or disabled depends on the existence of an enabled event forced by our supervisor. For any $s \in L(G, \gamma) \cap P^{-1}(t)$ with $tick \in E_{L(G)}(s)$, if $E_{L(G)}(s) \cap \gamma_2(t) = \emptyset$, that is, no enabled event is forced to occur, then *tick* is enabled; otherwise *tick* is disabled. This implies that, by suitably selecting the set $\gamma_2(t)$, our supervisor can control *tick* more flexibly than the Lin and Wonham's supervisor.

Definition 1. (Brandin & Wonham, 1994) A language $K \subseteq L(G)$ is said to be controllable if the following two conditions hold for any $s \in \overline{K}$:

(1) $E_{L(G)}(s) \cap \Sigma_{uc} \subseteq E_K(s)$.
(2) $[E_K(s) \cap \Sigma_{for} = \emptyset] \wedge [tick \in E_{L(G)}(s)] \Rightarrow tick \in E_K(s)$.

Definition 2. (Takai & Ushio, 2004) A language $K \subseteq L(G)$ is said to be weakly observable if the following two conditions hold:

(1) For any $s, s' \in \overline{K}$ and $\sigma \in \Sigma_c$,
$[P(s) = P(s')] \wedge [\sigma \in E_{L(G)}(s)] \wedge [\sigma \in E_K(s')] \Rightarrow [\sigma \in E_K(s)]$.
(2) Let $t \in P(\overline{K})$ be any observable event string. There exists a subset

$$F(t) \subseteq \left(\bigcup_{s \in \overline{K} \cap P^{-1}(t)} E_K(s) \right) \cap \Sigma_{for}$$

such that any $s \in \overline{K} \cap P^{-1}(t)$ with $tick \in E_{L(G)}(s)$ satisfies

$$tick \notin E_K(s) \Leftrightarrow E_{L(G)}(s) \cap F(t) \neq \emptyset.$$

The first condition of weak observability is equivalent to the observability condition for untimed systems. The second condition implies the existence of a subset of Σ_{for} such that *tick* is correctly controlled by forcing events in the subset to occur.

Theorem 3. (Takai & Ushio, 2004) Consider any nonempty language $K \subseteq L(G)$. There exists a supervisor $\gamma : P(L(G)) \rightarrow 2^{\Sigma_{act}} \times 2^{\Sigma_{for}}$ such that $L(G, \gamma) = K$ if and only if K is closed, controllable, and weakly observable.

Remark 4. The existence conditions of Theorem 3 are weaker than those of the Lin and Wonham's supervisor (Takai & Ushio, 2004).

4. CLOSED, CONTROLLABLE, AND WEAKLY OBSERVABLE SUBLANGUAGES

Throughout this section, we assume that the control specification is given by a closed regular language $K \subseteq L(G)$. We consider a problem to synthesize a supervisor $\gamma : P(L(G)) \rightarrow 2^{\Sigma_{act}} \times 2^{\Sigma_{for}}$ such that $L(G, \gamma) \subseteq K$. This problem arises when K does not satisfy the existence conditions of a supervisor. In order to solve the problem, we need to compute a closed, controllable, and weakly observable sublanguage of K. It is desirable that such a sublanguage is as large as possible. Lin and Wonham (1995) have synthesized a supervisor that achieves a sublanguage larger than the supremal closed, controllable, and normal sublanguage of K, denoted by $\sup CN(K)$, under the assumption that K is coherent, that is, for any $s \in K$,

$$\begin{aligned}
&tick \in E_{L(G)}(s) \\
&\Rightarrow \forall s', s'' \in P^{-1}P(s) \cap K; \\
&\quad [E_K(s') \cap \Sigma_{for} = E_K(s'') \cap \Sigma_{for}] \\
&\quad \wedge [E_{L(G)}(s') \cap \Sigma_{for} = E_{L(G)}(s'') \cap \Sigma_{for}].
\end{aligned}$$

In this section, we compute a closed, controllable, and weakly observable sublanguage larger than $\sup CN(K)$ without imposing this assumption.

For any automaton G, a normal automaton G_N such that $L(G) = L(G_N)$ can be effectively constructed (Cho & Marcus, 1989). Therefore, without loss of generality, we assume that the TDES G to be controlled is a normal automaton in this section. For the normal automaton G, we define an equivalence relation $\equiv_Q$ on Q as follows:

- $q \equiv_Q q'$ if $q, q' \in Q(t)$ for some $t \in P(L(G))$.

Let $\mathcal{C}$ be the partition of Q induced by $\equiv_Q$.

We present an algorithm for computing a closed, controllable, and weakly observable sublanguage larger than $\sup CN(K)$.

Algorithm 1. Let $K \subseteq L(G)$ be a closed regular language.

(1) Construct a strict subautomaton $G_S = (Q_S, \Sigma, \delta_S, q_0)$ of G such that $L(G_S)$ satisfies the controllability condition, the first condition of weak observability, and $\sup CN(K) \subseteq L(G_S) \subseteq K$. (A procedure for constructing G_S is described in Algorithm 2.)

(2) For each $C \in \mathcal{C}$ with $C \cap Q_S \neq \emptyset$, find a subset

$$\begin{aligned}\tilde{F}(C) \subseteq \{\sigma \in \Sigma_{for} | \exists q \in C \cap Q_S;\\ [\delta(tick, q)!] \wedge [\neg\delta_S(tick, q)!] \wedge\\ [\delta_S(\sigma, q)!]\}\end{aligned}$$

satisfying that, for any $q \in C \cap Q_S$, if $\delta(tick, q)!$ and $\neg\delta_S(tick, q)!$, then there exists $\sigma \in \tilde{F}(C)$ such that $\delta_S(\sigma, q)!$.

(3) For each $q \in Q_S$ with $\delta_S(tick, q)!$, remove the $tick$ transition originating from q if there exists $\sigma \in \tilde{F}(C)$ such that $q \in C$ and $\delta_S(\sigma, q)!$. Denote the resulting subautomaton of G_S by $G'_S = (Q'_S, \Sigma, \delta'_S, q_0)$.

First, we show how to construct a strict subautomaton G_S in Step 1. For the supremal closed and normal sublanguage $\sup N(K)$ of K, there exists a strict subautomaton G'_K of the normal automaton G such that $L(G'_K) = \sup N(K)$ (Cho & Marcus, 1989). Such a strict subautomaton G'_K is effectively constructed (Cho & Marcus, 1989). Note that $\sup CN(K) \subseteq L(G'_K) \subseteq K$. We consider the following iterative scheme to construct G_S from G'_K.

Algorithm 2. Let G'_K be a strict subautomaton of the normal automaton G such that $L(G'_K) = \sup N(K)$.

(1) Let $i := 0$ and $G^0 := G'_K$.

(2) Construct a strict subautomaton $G^{i,C}$ of G^i such that $L(G^{i,C})$ is the supremal closed and controllable sublanguage of $L(G^i)$.

(3) By using Algorithm 1 of Takai and Ushio (2003), construct a strict subautomaton G^{i+1} of $G^{i,C}$ such that $L(G^{i+1})$ is a sublanguage of $L(G^{i,C})$ that satisfies the first condition of weak observability and $\sup N(L(G^{i,C})) \subseteq L(G^{i+1})$.

(4) If $G^{i+1} = G^i$, then stop and output G^i as G_S. Otherwise, set $i := i + 1$ and go to Step 2.

Remark 5. Algorithm 2 is of exponential complexity since the complexity of constructing G'_K is exponential (Cho & Marcus, 1989).

In Step 2 of Algorithm 2, $G^{i,C}$ can be effectively constructed from G^i by slightly modifying the algorithm of Wonham and Ramadge (1987) that computes the supremal controllable sublanguage in the untimed setting. Note that the first condition of weak observability is equivalent to the observability condition in the untimed setting. So we can use Algorithm 1 of Takai and Ushio (2003) in Step 3. G^{i+1} is obtained by simply removing some states and transitions from G^i. Since G'_K has a finite number of states and transitions, Algorithm 2 terminates in a finite number of steps. Moreover, the generated language $L(\Phi) = \emptyset$ of the empty automaton Φ satisfies the controllability condition and the first condition of weak observability. Therefore, we can effectively construct G_S.

In Step 2 of Algorithm 1, we have to find a subset $\tilde{F}(C) \subseteq \Sigma_{for}$ for each $C \in \mathcal{C}$ with $C \cap Q_S \neq \emptyset$. Intuitively, $\tilde{F}(C)$ is the set of events forced to occur in order to prevent illegal $tick$ transitions from occurring. Controllability of $L(G_S)$ implies that, for any $q \in Q_S$, if $\delta(tick, q)!$ and $\neg\delta_S(tick, q)!$, then there exists $\sigma \in \Sigma_{for}$ such that $\delta_S(\sigma, q)!$. Thus, there always exists a subset $\tilde{F}(C)$ satisfying the requirement of Step 2. In order to reduce the number of legal $tick$ transitions that are unnecessarily removed in Step 3, it is desirable that $\tilde{F}(C)$ is as small as possible.

Lemma 6. Consider the notations of Algorithm 1 for a closed regular language $K \subseteq L(G)$. The language $L(G'_S)$ is controllable and weakly observable.

PROOF. The automaton G'_S is constructed from G_S by only removing some of the $tick$ transitions. A removed $tick$ transition is originating from a state $q \in Q_S$ such that there exists $\sigma \in \tilde{F}(C) \subseteq \Sigma_{for}$ with $q \in C$ and $\delta_S(\sigma, q)!$. Thus, $L(G'_S)$ satisfies the controllability condition and the first condition of weak observability.

We show that $L(G'_S)$ satisfies the second condition of weak observability. We consider any $t \in P(L(G'_S))$. Then, there exists $s' \in L(G'_S) \subseteq L(G_S)$ such that $P(s') = t$. Since $\delta_S(s', q_0) = \delta(s', q_0) \in Q_S$, we have $C \cap Q_S \neq \emptyset$, where $C := Q(t)$. Let

$$F(t) := \left(\bigcup_{s \in L(G'_S) \cap P^{-1}(t)} E_{L(G'_S)}(s) \right) \cap \tilde{F}(C),$$

where $\tilde{F}(C)$ satisfies the requirement of Step 2 of Algorithm 1. Since $\tilde{F}(C) \subseteq \Sigma_{for}$, we have

$$F(t) \subseteq \left(\bigcup_{s \in L(G'_S) \cap P^{-1}(t)} E_{L(G'_S)}(s) \right) \cap \Sigma_{for}.$$

We consider any $s \in L(G'_S) \cap P^{-1}(t)$ such that $tick \in E_{L(G)}(s)$. Let $q := \delta'_S(s, q_0) = \delta_S(s, q_0) = \delta(s, q_0) \in C \cap Q_S$. We need to show that

$$tick \notin E_{L(G'_S)}(s) \Leftrightarrow E_{L(G)}(s) \cap F(t) \neq \emptyset.$$

First, we suppose that $tick \notin E_{L(G'_S)}(s)$. If $tick \notin E_{L(G_S)}(s)$, then there exists $\sigma \in \tilde{F}(C)$ with $\delta_S(\sigma, q)!$ by the definition of $\tilde{F}(C)$. Also, since $\delta(\sigma, q)!$ and $\delta'_S(\sigma, q)!$, we have $\sigma \in E_{L(G)}(s) \cap F(t) \neq \emptyset$. If $tick \in E_{L(G_S)}(s)$, the $tick$ transition originating from q has been removed in Step 3 of Algorithm 1. Then, there exists $\sigma' \in \tilde{F}(C)$ such that $\delta_S(\sigma', q)!$. Since $\delta(\sigma', q)!$ and $\delta'_S(\sigma', q)!$, we have $\sigma' \in E_{L(G)}(s) \cap F(t) \neq \emptyset$. Next, we suppose that $E_{L(G)}(s) \cap F(t) \neq \emptyset$. Clearly, if $tick \notin E_{L(G_S)}(s)$ then $tick \notin E_{L(G'_S)}(s)$. We consider the case where $tick \in E_{L(G_S)}(s)$. Consider any $\sigma'' \in E_{L(G)}(s) \cap F(t) \subseteq \tilde{F}(C)$. There exists $s'' \in L(G'_S) \cap P^{-1}(t)$ such that $\sigma'' \in E_{L(G'_S)}(s'')$. Controllability of $L(G'_S)$ implies that if $\sigma'' \in \Sigma_{uc}$ then $\sigma'' \in E_{L(G'_S)}(s) \subseteq E_{L(G_S)}(s)$. Also, since $L(G'_S)$ satisfies the first condition of weak observability, if $\sigma'' \in \Sigma_c$ then $\sigma'' \in E_{L(G'_S)}(s) \subseteq E_{L(G_S)}(s)$. So we have $\delta_S(\sigma'', q)!$. In Step 3 of Algorithm 1, the $tick$ transition originating from q is removed, which implies that $tick \notin E_{L(G'_S)}(s)$. □

Lemma 7. Consider the notations of Algorithm 1 for a closed regular language $K \subseteq L(G)$. Then $\sup CN(K) \subseteq L(G'_S) \subseteq K$ holds.

PROOF. By Algorithm 1, $L(G'_S) \subseteq L(G_S) \subseteq K$ holds. Note that $\sup CN(K) = \sup CN(L(G_S)) \subseteq \sup N(L(G_S))$ holds. So it suffices to prove that $\sup N(L(G_S)) \subseteq L(G'_S)$. Since G_S is a strict subautomaton of the normal automaton G, the automaton G''_S with $L(G''_S) = \sup N(L(G_S))$ is constructed as follows (Cho & Marcus, 1989):

- If $q_0 \equiv_Q q$ for some $q \in Q - Q_S$, G''_S is the empty automaton Φ. Otherwise,

  $$G''_S = (Q''_S, \Sigma, \delta''_S, q_0),$$

  where

  $$Q''_S = \{q \in Q_S | \not\exists q' \in Q - Q_S;\ q \equiv_Q q'\}$$

  and $\delta''_S : \Sigma \times Q''_S \to Q''_S$ is defined by

  $$\delta''_S(\sigma, q) = \begin{cases} \delta_S(\sigma, q), & \text{if } \delta_S(\sigma, q) \in Q''_S, \\ \text{undefined, otherwise.} \end{cases}$$

If $G''_S = \Phi$, then $\emptyset = \sup N(L(G_S)) \subseteq L(G'_S)$ holds. We consider the case where $G''_S \neq \Phi$. We suppose that a $tick$ transition originating from a state $q \in Q_S$ is removed in Step 3 of Algorithm 1. Then there exists $\sigma \in \tilde{F}(C)$ such that $q \in C$ and $\delta_S(\sigma, q)!$. Since $\sigma \in \tilde{F}(C)$, there exists $q' \in C \cap Q_S$ such that $\delta(tick, q')!$ and $\neg\delta_S(tick, q')!$. Since G_S is a strict subautomaton of G, we have $q'_t := \delta(tick, q') \in Q - Q_S$. Also, since $q \equiv_Q q'$, we have $q_t \equiv_Q q'_t$ where $q_t := \delta(tick, q)$, which implies that $q_t \notin Q''_S$. So the $tick$ transition originating from $q \in Q_S$ is also removed in the construction of G''_S. Therefore, G''_S is a subautomaton of G'_S, which implies that $\sup N(L(G_S)) \subseteq L(G'_S)$. □

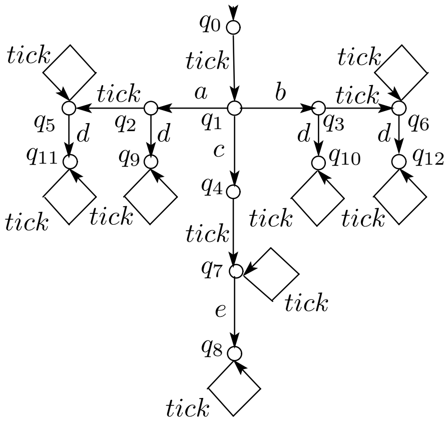

Fig. 1. A TDES model of Example 9.

By the above two lemmas, we have the following theorem.

Theorem 8. Consider the notations of Algorithm 1 for a closed regular language $K \subseteq L(G)$. The language $L(G'_S)$ is a closed, controllable, and weakly observable sublanguage of K larger than $\sup CN(K)$.

Example 9. We consider a TDES modeled by an automaton G shown in Fig. 1. Let $\Sigma_c = \{d, e\}$, $\Sigma_{for} = \{d\}$, and $\Sigma_o = \{d, tick\}$. The observer G^P of G is shown in Fig. 2. Since the state space of G^P is a partition of the state space of G, G is normal. Let $\mathcal{C} = \{C_0, C_1, C_2, C_3, C_4\}$ where $C_0 = \{q_0\}$, $C_1 = \{q_1, q_2, q_3, q_4\}$, $C_2 = \{q_5, q_6, q_7, q_8\}$, $C_3 = \{q_9, q_{10}\}$, and $C_4 = \{q_{11}, q_{12}\}$.

We consider a language $K \subseteq L(G)$ which is generated by an automaton G_K shown in Fig. 3. For $tick \in P(K)$, the second condition of weak observability does not hold. Thus, there does not exist a supervisor that exactly achieves K. Since $tick \cdot a \in K$, $tick \in E_{L(G)}(tick \cdot a)$, $tick \cdot c \in P^{-1}P(tick \cdot a) \cap K$, and $\{d\} = E_K(tick \cdot a) \cap$

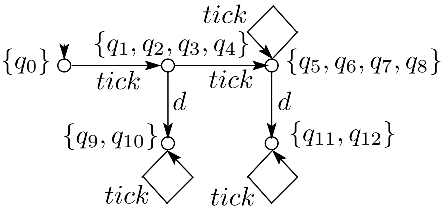

Fig. 2. The observer G^P.

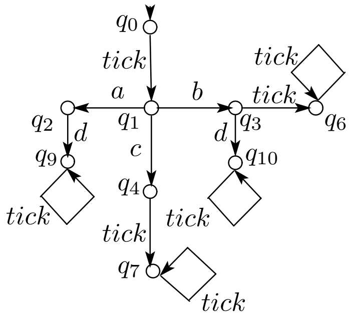

Fig. 3. An automaton G_K of Example 9.

$\Sigma_{for} \neq E_K(tick \cdot c) \cap \Sigma_{for} = \emptyset$, K is not coherent, which implies that the synthesis method of Lin and Wonham (1995) cannot be applied to this example. We compute a closed, controllable, and weakly observable sublanguage of K by using Algorithm 1.

Since G_K is a strict subautomaton of G such that $L(G_K)$ satisfies the controllability condition, the first condition of weak observability, and $\emptyset = \sup CN(K) \subseteq L(G_K) \subseteq K$, we let $G_S := G_K$ in Step 1. We proceed to Step 2. We have $C_i \cap Q_S \neq \emptyset$ $(i = 0, 1, 2, 3)$, and let $\tilde{F}(C_1) = \{d\}$ and $\tilde{F}(C_i) = \emptyset$ $(i = 0, 2, 3)$. For $q_2 \in C_1 \cap Q_S$ such that $\delta(tick, q_2)!$ and $\neg\delta_S(tick, q_2)!$, there exists $d \in \tilde{F}(C_1)$ with $\delta_S(d, q_2)!$. Also, for C_i $(i = 0, 2, 3)$, there does not exist $q \in C_i \cap Q_S$ such that $\delta(tick, q)!$ and $\neg\delta_S(tick, q)!$. So, each $\tilde{F}(C_i)$ $(i = 0, 1, 2, 3)$ satisfies the requirement of Step 2. Finally, we proceed to Step 3. For $q_3 \in Q_S$ with $\delta_S(tick, q_3)!$, there exists $d \in \tilde{F}(C_1)$ such that $q_3 \in C_1$ and $\delta_S(d, q_3)!$. So the *tick* transition originating from q_3 is removed. For other states $q \in Q_S$ with $\delta_S(tick, q)!$, there does not exist $\sigma \in \tilde{F}(C)$ such that $q \in C$ and $\delta_S(\sigma, q)!$, which implies that the *tick* transition originating from q is not removed. The resulting subautomaton G'_S is shown in Fig. 4. In this example, the sublanguage $L(G'_S)$ computed by Algorithm 1 is strictly larger than $\sup CN(K) = \emptyset$.

5. CONCLUSION

We have presented an algorithm that effectively computes a closed, controllable, and weakly observable sublanguage for TDESs. By using the computed sublanguage, we can synthesize a supervisor, which is more permissive than the one that achieves the supremal closed, controllable, and normal sublanguage.

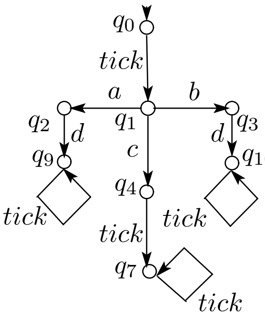

Fig. 4. The automaton G'_S.

ACKNOWLEDGMENTS

This work was supported in part by MEXT under Grant-in-Aid for Scientific Research (No. 15760321).

REFERENCES

Brandin, B. A., & Wonham, W. M. (1994). Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, *39*(2), 329–342.

Cho, H., & Marcus, S. I. (1989). On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation. *Mathematics of Control, Signals, and Systems*, *2*(1), 47–69.

Cieslak, R., Desclaux, C., Fawaz, A. S., & Varaiya, P. (1988). Supervisory control of discrete-event processes with partial observations. *IEEE Transactions on Automatic Control*, *33*(3), 249–260.

Lin, F., & Wonham, W. M. (1995). Supervisory control of timed discrete-event systems under partial observation. *IEEE Transactions on Automatic Control*, *40*(3), 558–562.

Ramadge, P. J., & Wonham, W. M. (1987). Supervisory control of a class of discrete-event processes. *SIAM Journal on Control and Optimization*, *25*(1), 206–230.

Takai, S., & Ushio, T. (2003). Effective computation of an $L_m(G)$-closed, controllable, and observable sublanguage arising in supervisory control. *Systems & Control Letters*, *49*(3), 191–200.

Takai, S., & Ushio, T. (2004). A new class of supervisors for timed discrete event systems under partial observation. *Proceedings of the 43rd IEEE Conference on Decision and Control* (pp. 497–502).

Wonham, W. M., & Ramadge, P. J. (1987). On the supremal controllable sublanguage of a given language. *SIAM Journal on Control and Optimization*, *25*(3), 637–659.