2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Supervisory Control of Discrete Event Systems Modeled by Mealy Automata with Nondeterministic Output Functions

Toshimitsu Ushio and Shigemasa Takai

Abstract— **In the conventional supervisory control framework for discrete event systems with partial event observation, it is assumed that, for each event, the corresponding output symbol is determined deterministically. However, this assumption does not hold in discrete event systems such as a system with sensor errors and a mobile system, where an output symbol depends on not only an event but also a state at which the event occurs. In this paper, we model such a discrete event system by a Mealy automaton with a nondeterministic output function. We consider two kinds of supervisors: one assigns its control action based on a permissive policy and the other based on an anti-permissive one. We present necessary and sufficient conditions for each of them to achieve a given specification. We then present algorithms for verifying these conditions. Moreover, we discuss the relationship between the two supervisors in the case that an output function is deterministic.**

I. Introduction

The supervisory control framework for discrete event systems (DESs) with partial event observation was proposed in [1], [2]. In this conventional framework, partial event observation is represented by the projection function from the event set to the observable event set [1] or the mask function from the set of events to the set of output symbols [2]. It was shown in [1], [2] that there exists a partial observation supervisor that achieves a given specification language if and only if the language is controllable [3] and observable [1], [2]. There are many studies dealing with supervisory control of DESs with partial event observation [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. In state feedback control of DESs, it is assumed that partial information of the current state is available for deciding the control action [18], [19], [20]. Partial state observation is represented by the mask function from the state space to the observation space. Further, supervisory control using partial event and state observations was studied [21]. In all existing work mentioned above, the observed output is determined deterministically based on an event and/or a state.

Recently, a sensor failure tolerant supervisory control problem has been addressed [22]. In this problem, it is assumed that an observable event becomes unobservable after failure of the corresponding sensor. That is, an observed symbol depends on the state of the corresponding sensor. In a mobile system, an output symbol corresponding to an event may depend on the state at which the event occurs, and may not be determined uniquely due to packet loss. State-dependence and nondeterminism of partial event observation are represented by introducing a nondeterministic output function that maps a pair of an event and a state into a set of possible output symbols. Motivated by this, we study supervisory control of DESs modeled by Mealy automata [23] with nondeterministic output functions.

We consider two kinds of supervisors: one assigns its control action based on a permissive policy [24] and the other based on an anti-permissive one [24]. The former is called the permissive supervisor, and the latter is called the anti-permissive supervisor. In the conventional supervisory control framework [1], [2], a given specification language is achieved by the permissive supervisor if and only if it is achieved by the anti-permissive one. We show that, however, there exists a language that can be achieved only by the anti-permissive supervisor under a nondeterministic output function. We introduce notions of P-observability and A-observability to characterize classes of languages achievable by the permissive supervisor and the anti-permissive one, respectively. We provide effective tests to verify P-observability and A-observability of a given language.

Moreover, we discuss the existence of a supervisor that achieves a given specification language. We show that A-observability is weaker than P-observability, and that controllability and A-observability are only sufficient for the existence of a supervisor. That is, there may exist a supervisor that cannot be synthesized by using the permissive nor anti-permissive policy. We then prove that in a special case that an output function is deterministic, these sufficient conditions become necessary and sufficient, and P-observability and A-observability are equivalent.

In this paper, a notation $|A|$ represents the cardinality of a set A. A^* denotes the set of all finite strings of elements of A, including the empty string ε . For a language $K \subseteq A^*$ over a set A, the set of all prefixes of strings in K is denoted by $\overline{K}$. K is said to be (prefix-)closed if $K = \overline{K}$. For a finite string $s \in A^*$, we write $\overline{s} := \overline{\{s\}}$. Also, $|s|$ denotes the length of s.

II. Discrete Event System Model

In this paper, we consider a DES modeled by a Mealy automaton [23] with a nondeterministic output function:

$$G = (Q, \Sigma, \Delta, f, \lambda, q_0),$$

where Q is the set of states, Σ is the finite set of events, Δ is the set of output symbols, $f : Q \times \Sigma \to Q$ is the partial

This work was supported in part by Grant-in-Aid for Scientific Research (No. 17360198 and No. 18560433).

T. Ushio is with the Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka 560-8531, Japan, e-mail: ushio@sys.es.osaka-u.ac.jp

S. Takai is with the Graduate School of Science and Technology, Kyoto Institute of Technology, Sakyo, Kyoto 606-8585, Japan, e-mail: takai@kit.ac.jp

978-1-4244-4524-0/09/$25.00 ©2009 AACC

state transition function, $\lambda : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^{\Delta \cup \{\varepsilon\}}$ is the nondeterministic output function, and $q_0 \in Q$ is the initial state. When an event $\sigma \in \Sigma$ occurs at a state $q \in Q$, an output symbol in $\lambda(q, \sigma) \in 2^{\Delta \cup \{\varepsilon\}}$ is observed nondeterministically by a supervisor. If $\varepsilon \in \lambda(q, \sigma)$, no output symbol may be observed when $\sigma \in \Sigma$ occurs at $q \in Q$. For each $q \in Q$, we let $\lambda(q, \varepsilon) = \{\varepsilon\}$. By introducing the nondeterministic output function λ, both state-dependence and nondeterminism of partial event observation are represented. The state transition function $f : Q \times \Sigma \to Q$ is extended to $f : Q \times \Sigma^* \to Q$ in a natural way. The notation $f(q, s)!$ means that $f(q, s)$ is defined for $q \in Q$ and $s \in \Sigma^*$. Also, $\neg f(q, s)!$ denotes the negation of $f(q, s)!$. The generated language $L(G)$ of G is defined by $L(G) = \{s \in \Sigma^* \mid f(q_0, s)!\}$.

Due to the nondeterminism of the output function, the output string is not uniquely determined. For each $s \in L(G)$, a set $O(s)$ of possible output strings is defined inductively as follows:

- $O(\varepsilon) = \{\varepsilon\}$,
- $(\forall s\sigma \in L(G)\ (\sigma \in \Sigma))$

$$O(s\sigma) = \{\tau\delta \in \Delta^* \mid \tau \in O(s),\ \delta \in \lambda(f(q_0, s), \sigma)\}.$$

For a language $K \subseteq L(G)$, the set of all possible output strings is denoted by $O(K)$, that is,

$$O(K) = \bigcup_{s \in K} O(s) \subseteq \Delta^*.$$

Also, for each output string $\tau \in O(L(G))$, the set $O^{-1}(\tau) \subseteq L(G)$ is defined by

$$O^{-1}(\tau) = \{s \in L(G) \mid \tau \in O(s)\}.$$

That is, $O^{-1}(\tau)$ is the set of event strings for which τ can be observed as an output string.

In the conventional framework of supervisory control under partial event observation, for each event, the corresponding output symbol is determined deterministically [1], [2]. This deterministic event observation is modeled by the mask function $M : \Sigma \to \Delta \cup \{\varepsilon\}$ such that $\lambda(q, \sigma) = \{M(\sigma)\}$. So our framework using the nondeterministic output function generalizes the conventional one.

A pair $(\sigma, \delta) \in \Sigma \times (\Delta \cup \{\varepsilon\})$ of an event $\sigma \in \Sigma$ and an output symbol $\delta \in \Delta \cup \{\varepsilon\}$ is called an extended event. The empty extended event string is denoted by $\hat{\varepsilon} = (\varepsilon, \varepsilon)$. Let $t = a_1 a_2 \dots a_n \in (\Sigma \times (\Delta \cup \{\varepsilon\}))^*$ be an extended event string, where each a_i is of the form $a_i = (\sigma_i, \delta_i) \in \Sigma \times (\Delta \cup \{\varepsilon\})$. We define two projection functions $P_\Sigma : (\Sigma \times (\Delta \cup \{\varepsilon\}))^* \to \Sigma^*$ and $P_\Delta : (\Sigma \times (\Delta \cup \{\varepsilon\}))^* \to (\Delta \cup \{\varepsilon\})^*$ by

$$\begin{aligned} P_\Sigma(t) &= \sigma_1\sigma_2 \dots \sigma_n, \\ P_\Delta(t) &= \delta_1\delta_2 \dots \delta_n. \end{aligned}$$

For each event string $s = \sigma_1\sigma_2 \dots \sigma_n \in L(G)$, an extended event string $t = a_1 a_2 \dots a_n \in (\Sigma \times (\Delta \cup \{\varepsilon\}))^*$ is said to be compatible if the following three conditions hold:

- $P_\Sigma(t) = s$,
- $P_\Delta(a_1) \in \lambda(q_0, \sigma_1)$, and
- $P_\Delta(a_i) \in \lambda(f(q_0, \sigma_1\sigma_2 \dots \sigma_{i-1}), \sigma_i)\ (i = 2, 3, \dots, n)$.

Denoted by $\Pi(s)$ is the set of all compatible extended event strings for s. For each compatible extended event string $t \in \Pi(s)$, $P_\Delta(t)$ is an output string that is possibly observed when s is executed. Note that $\Pi(\varepsilon) = \{\hat{\varepsilon}\}$ for the empty event string $\varepsilon \in L(G)$. We define the set $L_e(G) \subseteq (\Sigma \times (\Delta \cup \{\varepsilon\}))^*$ of all compatible extended event strings of G as

$$L_e(G) = \bigcup_{s \in L(G)} \Pi(s).$$

III. Supervisory Control

A. Problem Formulation

The event set Σ is partitioned into the controllable event set Σ_c and the uncontrollable event set Σ_u, that is, $\Sigma = \Sigma_c \dot{\cup} \Sigma_u$ [3]. Formally, a supervisor is defined as a function $S : O(L(G)) \to 2^{\Sigma_c}$. For each possible output string $\tau \in O(L(G))$, $S(\tau)$ is the set of controllable events which are disabled by S after observing τ. Let S/G be the closed-loop system controlled by the supervisor S. The set $L_e(S/G) \subseteq L_e(G)$ of all compatible extended event strings of S/G is defined inductively as follows:

- $\hat{\varepsilon} \in L_e(S/G)$,
- $(\forall t \in L_e(S/G), a \in \Sigma \times (\Delta \cup \{\varepsilon\}))$

$$ta \in L_e(S/G) \Leftrightarrow ta \in L_e(G) \wedge P_\Sigma(a) \notin S(P_\Delta(t)).$$

Also, the generated language $L(S/G)$ of the closed-loop system S/G is defined by

$$L(S/G) = \{s \in \Sigma^* \mid \exists t \in L_e(S/G) : P_\Sigma(t) = s\}. \quad (1)$$

For each event string $s = \sigma_1\sigma_2 \dots \sigma_n \in L(S/G)$, we have $\sigma_1 \notin S(\varepsilon)$, and there exists at least one corresponding output string $\delta_1\delta_2 \dots \delta_n \in (\Delta \cup \{\varepsilon\})^*$ such that $\sigma_i \notin S(\delta_1\delta_2 \dots \delta_{i-1})$ $(i = 2, 3, \dots, n)$. Clearly, $L(S/G) \subseteq L(G)$ holds.

In this paper, we assume that a control specification is given as a nonempty closed language $K \subseteq L(G)$, and consider a problem of synthesizing a supervisor $S : O(L(G)) \to 2^{\Sigma_c}$ such that $L(S/G) = K$. A closed language K is said to be controllable [3] (with respect to Σ_u and $L(G)$) if

$$K\Sigma_u \cap L(G) \subseteq K. \quad (2)$$

The following lemma can be easily obtained as in the conventional supervisory control under partial event observation [1], [2].

Lemma 1: Let $K \subseteq L(G)$ be a nonempty closed language. If there exists a supervisor $S : O(L(G)) \to 2^{\Sigma_c}$ such that $L(S/G) = K$, K is controllable.

B. Permissive and Anti-Permissive Supervisors

We define two kinds of supervisors with different control policies.

Definition 1: A supervisor $S : O(L(G)) \to 2^{\Sigma_c}$ is said to be *permissive* if, for each $\tau \in O(L(G))$ and $\sigma \in \Sigma_c$,

$$O^{-1}(\tau)\{\sigma\} \cap K \neq \emptyset \Leftrightarrow \sigma \notin S(\tau).$$

Also, it is said to be *anti-permissive* if, for each $\tau \in O(L(G))$ and $\sigma \in \Sigma_c$,

$$(O^{-1}(\tau) \cap K)\{\sigma\} \cap (L(G) - K) \neq \emptyset \Leftrightarrow \sigma \in S(\tau).$$

Hereafter, the permissive and anti-permissive supervisor are denoted by S_P and S_A, respectively. It is clear from the definition of S_A that

$$(O^{-1}(\tau) \cap K)\{\sigma\} \cap L(G) \subseteq K \Leftrightarrow \sigma \notin S_A(\tau).$$

We then have the following proposition whose proof is straightforward.

Proposition 1: For a nonempty closed controllable language $K \subseteq L(G)$,

$$L(S_A/G) \subseteq K \subseteq L(S_P/G). \tag{3}$$

C. Supervisory Control under the Mask Function

We review the results on the conventional supervisory control under partial event observation. Let $M : \Sigma \to \Delta \cup \{\varepsilon\}$ be the mask function. For each $q \in Q$ and $\sigma \in \Sigma$, the output symbol is uniquely determined by the mask function M as follows:

$$\lambda(q, \sigma) = \{M(\sigma)\}. \tag{4}$$

A closed language $K \subseteq L(G)$ is said to be observable [1], [2] (with respect to M, Σ_c, and $L(G)$) if, for any s, $s' \in K$ and $\sigma \in \Sigma_c$,

$$M(s) = M(s') \wedge s\sigma \in K \wedge s'\sigma \in L(G) \Rightarrow s'\sigma \in K.$$

Theorem 1: [1], [2] Assume that the output function $\lambda : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^{\Delta \cup \{\varepsilon\}}$ is given by (4). Then, the following four statements are equivalent for a nonempty closed language $K \subseteq L(G)$.

1) K is controllable and observable.
2) There exists a supervisor $S : O(L(G)) \to 2^{\Sigma_c}$ such that $L(S/G) = K$.
3) For the permissive supervisor S_P, $L(S_P/G) = K$.
4) For the anti-permissive supervisor S_A, $L(S_A/G) = K$.

However, in a general case that the output function λ is nondeterministic, the equivalence of 3) and 4) of Theorem 1 does not necessarily hold as shown in the following example.

Example 1: We consider a DES modeled by a Mealy automaton shown in Fig. 1, where $\Sigma = \Sigma_c = \{\sigma_1, \sigma_2\}$ and $\Delta = \{\delta_1, \delta_2\}$. A label of an arc is of the form σ/A, where $\sigma \in \Sigma$ represents an event, and $A \subseteq \Delta \cup \{\varepsilon\}$ denotes the set of possible output symbols. For the generated language $L(G) = \overline{\{\sigma_1, \sigma_2\}^2}$, we consider a control specification given by a nonempty closed language $K = L(G) - \{\sigma_2\sigma_1\}$. Then, the permissive supervisor S_P and the anti-permissive supervisor S_A are given as follows:

$$S_P(\tau) = \begin{cases} \emptyset, & \text{if } \tau \in \{\varepsilon, \delta_1, \delta_2\} \\ \Sigma_c, & \text{otherwise,} \end{cases}$$

$$S_A(\tau) = \begin{cases} \{\sigma_1\}, & \text{if } \tau = \delta_2 \\ \emptyset, & \text{otherwise.} \end{cases}$$

For these supervisors, we have $L(S_P/G) = L(G)$ and $L(S_A/G) = K$. Only the anti-permissive supervisor achieves the specification in this example.

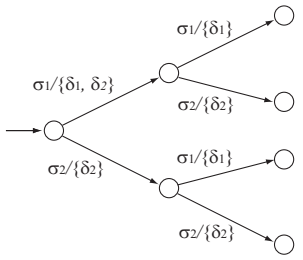

Fig. 1. Controlled discrete event system G of Example 1.

IV. Existence of Supervisors

A. Permissive Supervisor

We present necessary and sufficient conditions under which the permissive supervisor S_P achieves a specification language K. We introduce a notion of P-observability, which can be seen as a direct extension of the conventional observability condition.

Definition 2: A closed language $K \subseteq L(G)$ is said to be *P-observable* if, for any $\tau \in O(K)$ and $\sigma \in \Sigma_c$,

$$[O^{-1}(\tau)\{\sigma\} \cap K \neq \emptyset] \Rightarrow [(O^{-1}(\tau) \cap K)\{\sigma\} \cap L(G) \subseteq K].$$

The following theorem shows that controllability and P-observability are necessary and sufficient conditions for the permissive supervisor S_P to achieve the specification language K.

Theorem 2: Let $K \subseteq L(G)$ be a nonempty closed language. The permissive supervisor S_P satisfies $L(S_P/G) = K$ if and only if K is controllable and P-observable.

Proof: We first suppose that the permissive supervisor S_P satisfies $L(S_P/G) = K$. By Lemma 1, K is controllable. We prove by contradiction that K is P-observable. If K is not P-observable, then there exist $\tau \in O(K)$ and $\sigma \in \Sigma_c$ such that

$$O^{-1}(\tau)\{\sigma\} \cap K \neq \emptyset, \tag{5}$$

$$(O^{-1}(\tau) \cap K)\{\sigma\} \cap (L(G) - K) \neq \emptyset. \tag{6}$$

By (6), there exists $s \in K$ such that $s\sigma \in (O^{-1}(\tau) \cap K)\{\sigma\} \cap (L(G) - K)$. Then, there exists a compatible extended event string $t \in L_e(G)$ such that $P_\Sigma(t) = s$ and $P_\Delta(t) = \tau$. We prove by the induction that $t \in L_e(S_P/G)$. For the empty extended event string $\hat{\varepsilon} \in \overline{t}$, we have $\hat{\varepsilon} \in L_e(S_P/G)$. We consider $t' \in \overline{t}$ and $a := (\sigma', \delta') \in \Sigma \times (\Delta \cup \{\varepsilon\})$ such that $t' \in L_e(S/G)$ and $t'a \in \overline{t}$. Since $t \in L_e(G)$, we have $t'a \in L_e(G)$. Also, since $P_\Sigma(t'a) = P_\Sigma(t')\sigma' \in \overline{s} \subseteq K$, we have

$$P_\Sigma(t')\sigma' \in O^{-1}(P_\Delta(t'))\{\sigma'\} \cap K \neq \emptyset.$$

By the definition of S_P, we have $\sigma' \notin S_P(P_\Delta(t'))$. It follows that $t'a \in L_e(S_P/G)$. Thus, we have $t \in L_e(S_P/G)$. Further, since $\sigma \notin S_P(\tau)$ by (5), we have $t(\sigma, \delta) \in L_e(S_P/G)$, where $\delta \in \lambda(f(q_0, s), \sigma)$. So we have $s\sigma \in L(S_P/G) = K$, and this is a contradiction. We can conclude that K is P-observable.

We next suppose that K is controllable and P-observable. Since K is nonempty and closed, we have $\varepsilon \in L(S_P/G) \cap$

K. We consider any $s \in L(S_P/G) \cap K$ and $\sigma \in \Sigma$. We first assume that $s\sigma \in L(S_P/G)$. By controllability of K, if $\sigma \in \Sigma_u$, then $s\sigma \in K$. We consider the case that $\sigma \in \Sigma_c$. Since $s\sigma \in L(S_P/G)$, there exists an extended event string $ta \in L_e(S_P/G)$ such that $P_\Sigma(t) = s$ and $P_\Sigma(a) = \sigma$. Since $ta \in L_e(S_P/G)$, we have $\sigma \notin S_P(P_\Delta(t))$. By the definition of S_P, we have

$$O^{-1}(P_\Delta(t))\{\sigma\} \cap K \neq \emptyset.$$

Since $P_\Delta(t) \in O(s) \subseteq O(K)$, it follows from P-observability of K that

$$s\sigma \in (O^{-1}(P_\Delta(t)) \cap K)\{\sigma\} \cap L(G) \subseteq K.$$

Thus, we have $L(S_P/G) \subseteq K$. Conversely, we assume that $s\sigma \in K$. Since $s \in L(S_P/G)$, there exists an extended event string $t \in L_e(S_P/G)$ such that $P_\Sigma(t) = s$. We have $s\sigma \in O^{-1}(P_\Delta(t))\{\sigma\} \cap K \neq \emptyset$, which implies that $\sigma \notin S_P(P_\Delta(t))$. It follows that $t(\sigma, \delta) \in L_e(S_P/G)$, where $\delta \in \lambda(f(q_0, s), \sigma)$. So we have $s\sigma \in L(S_P/G)$. Thus, $K \subseteq L(S_P/G)$ holds. ■

Remark 1: The sensor failure tolerant supervisory control problem studied by [22] can be transformed into a problem of controlling a Mealy automaton with a deterministic output function by the permissive supervisor S_P. In this sense, Theorem 2 can be viewed as a generalization of the result of [22].

B. Anti-Permissive Supervisor

We show necessary and sufficient conditions for the anti-permissive supervisor S_A to achieve a specification language K. We define a notion of A-observability.

Definition 3: A closed language $K \subseteq L(G)$ is said to be *A-observable* if, for any $s \in K$ and $\sigma \in \Sigma_c$ such that $s\sigma \in K$, there exists an extended event string $t \in \Pi(s)$ that satisfies the following condition.

- For any $t' \in \overline{t}$ and $\sigma' \in \Sigma_c$ such that $P_\Sigma(t')\sigma' \in \overline{s\sigma}$,

$$(O^{-1}(P_\Delta(t')) \cap K)\{\sigma'\} \cap L(G) \subseteq K. \qquad (7)$$

The following theorem shows that controllability and A-observability are necessary and sufficient conditions under which the anti-permissive supervisor S_A achieves a specification language K.

Theorem 3: Let $K \subseteq L(G)$ be a nonempty closed language. The anti-permissive supervisor S_A satisfies $L(S_A/G) = K$ if and only if K is controllable and A-observable.

Proof: We first suppose that the anti-permissive supervisor S_A satisfies $L(S_A/G) = K$. By Lemma 1, K is controllable. We suppose for contradiction that K is not A-observable. Then, there exist $s \in K$ and $\sigma \in \Sigma_c$ such that $s\sigma \in K$, and for any $t \in \Pi(s)$, there exist $t' \in \overline{t}$ and $\sigma' \in \Sigma_c$ satisfying

$$P_\Sigma(t')\sigma' \in \overline{s\sigma},$$
$$(O^{-1}(P_\Delta(t')) \cap K)\{\sigma'\} \cap (L(G) - K) \neq \emptyset.$$

By the definition of S_A, we have $\sigma' \in S_A(P_\Delta(t'))$. It follows that $s\sigma \notin L(S_A/G)$. This contradicts the assumption that $L(S_A/G) = K$. Thus, K is A-observable.

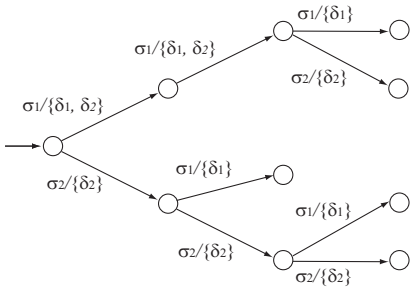

Fig. 2. Controlled discrete event system G of Example 2.

We next suppose that K is controllable and A-observable. Since K is nonempty and closed, we have $\varepsilon \in L(S_A/G) \cap K$. We consider any $s \in L(S_A/G) \cap K$ and $\sigma \in \Sigma$. We first assume that $s\sigma \in L(S_A/G)$. By controllability of K, if $\sigma \in \Sigma_u$, then $s\sigma \in K$. We consider the case that $\sigma \in \Sigma_c$. There exists an extended event string $ta \in L_e(S_A/G)$ such that $P_\Sigma(t) = s$ and $P_\Sigma(a) = \sigma$. Since $ta \in L_e(S_A/G)$, we have $\sigma \notin S_A(P_\Delta(t))$. By the definition of S_A, we have

$$s\sigma \in (O^{-1}(P_\Delta(t)) \cap K)\{\sigma\} \cap L(G) \subseteq K.$$

Thus, we have $L(S_A/G) \subseteq K$. Conversely, we assume that $s\sigma \in K$. If $\sigma \in \Sigma_u$, we have $s\sigma \in L(S_A/G)$. We consider the case that $\sigma \in \Sigma_c$. Since K is A-observable, there exists $t \in \Pi(s)$ such that any $t' \in \overline{t}$ and $\sigma' \in \Sigma_c$ with $P_\Sigma(t')\sigma' \in \overline{s\sigma}$ satisfy (7). We have $t \in L_e(S_A/G)$ and $\sigma \notin S_A(P_\Delta(t))$, which implies that $s\sigma \in L(S_A/G)$. Thus, $K \subseteq L(S_A/G)$ holds. ■

C. Discussions

We discuss the existence of a supervisor S such that $L(S/G) = K$ for a given nonempty closed language $K \subseteq L(G)$. We can easily verify that if K is P-observable, then it is A-observable. By Theorem 3, controllability and A-observability are sufficient conditions for the existence of a supervisor. However, as shown in the following example, these are not necessary conditions.

Example 2: We consider a DES modeled by a Mealy automaton shown in Fig. 2, where $\Sigma = \Sigma_c = \{\sigma_1, \sigma_2\}$ and $\Delta = \{\delta_1, \delta_2\}$. Let K be a nonempty closed language generated by an automaton shown in Fig. 3. We can verify that K is not A-observable. For $s := \sigma_2\sigma_2 \in K$, we have $\Pi(s) = \{(\sigma_2, \delta_2)(\sigma_2, \delta_2)\}$. Then, for $t' := (\sigma_2, \delta_2)(\sigma_2, \delta_2)$ and $\sigma_2 \in \Sigma_c$, we have

$$(O^{-1}(P_\Delta(t')) \cap K)\{\sigma_2\} \cap (L(G) - K) = \{\sigma_1\sigma_1\sigma_2\} \neq \emptyset,$$

which implies that K is not A-observable. However, the following supervisor S satisfies $L(S/G) = K$.

$$S(\tau) = \begin{cases} \emptyset, & \text{if } \tau \in \{\varepsilon, \delta_1, \delta_2\delta_2\} \\ \{\sigma_1\}, & \text{if } \tau = \delta_2 \\ \{\sigma_2\}, & \text{if } \tau = \delta_1\delta_1 \\ \Sigma_c, & \text{otherwise.} \end{cases}$$

Note that this supervisor S cannot be synthesized by using the permissive nor anti-permissive policy.

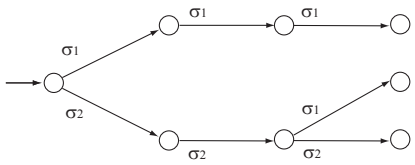

Fig. 3. Automaton representation of control specification of Example 2.

In a special case that the output function λ is deterministic, the following theorem shows that A-observability and P-observability are equivalent, and controllability and P-observability are necessary and sufficient conditions for the existence of a supervisor.

Theorem 4: Let $K \subseteq L(G)$ be a nonempty closed language. Assume that the output function $\lambda : Q \times (\Sigma \cup \{\varepsilon\}) \to \Delta \cup \{\varepsilon\}$ is deterministic. Then, K is P-observable if and only if it is A-observable. Further, there exists a supervisor $S : O(L(G)) \to 2^{\Sigma_c}$ such that $L(S/G) = K$ if and only if K is controllable and P-observable (equivalently, A-observable).

Proof: To prove the equivalence of P-observability and A-observability, it suffices to show that if K is A-observable, then it is P-observable. Consider any $\tau \in O(K)$ and $\sigma \in \Sigma_c$ such that $O^{-1}(\tau)\{\sigma\} \cap K \neq \emptyset$. Then, there exists $s \in O^{-1}(\tau)$ such that $s\sigma \in K$. Since the output function λ is deterministic, we have $|\Pi(s)| = 1$. Let $\{t\} = \Pi(s)$. By A-observability of K, we have

$$(O^{-1}(P_\Delta(t)) \cap K)\{\sigma\} \cap L(G) \subseteq K.$$

Since $P_\Delta(t) = \tau$, K is P-observable.

Next, we show that controllability and P-observability are necessary and sufficient conditions for the existence of a supervisor. By Theorem 2, controllability and P-observability are sufficient conditions for the existence of a supervisor. We prove that if there exists $S : O(L(G)) \to 2^{\Sigma_c}$ such that $L(S/G) = K$, then K is controllable and P-observable. By Lemma 1, K is controllable. Consider any $\tau \in O(K)$ and $\sigma \in \Sigma_c$ such that $O^{-1}(\tau)\{\sigma\} \cap K \neq \emptyset$. There exists $s \in O^{-1}(\tau)$ such that $s\sigma \in K = L(S/G)$. Since the output function λ is deterministic, we have $\sigma \notin S(\tau)$. Consider any $s'\sigma \in (O^{-1}(\tau) \cap K)\{\sigma\} \cap L(G)$. Since the output function λ is deterministic, and $\sigma \notin S(\tau)$, we have $s'\sigma \in L(S/G) = K$. Thus, K is P-observable. ■

V. Verification Results

In this section, we present algorithms for verifying P-observability and A-observability. Throughout this section, we assume that the system G has the finite state set Q, and a nonempty closed language $K \subseteq L(G)$ is generated by a finite automaton $G_K = (Q_K, \Sigma, f_K, q_{K,0})$ (without the output function). The proofs of the theorems in this section are omitted due to page limits.

A. Verification of P-Observability

In [5], it was shown that observability can be verified in polynomial-time. We generalize this result to verify P-observability. We construct a testing automaton $T_P = (R_P, \Sigma_P, f_P, r_{P,0})$ as follows:

- $R_P = Q \times Q_K \times Q \times Q_K$.
- $r_{P,0} = (q_0, q_{K,0}, q_0, q_{K,0})$.
- $\Sigma_P = (\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\}) - \{(\varepsilon, \varepsilon)\}$.
- $f_P : R_P \times \Sigma_P \to R_P$ is defined as follows: For each $r_P = (q_1, q_{K1}, q_2, q_{K2}) \in R_P$ and $\sigma_P = (\sigma_1, \sigma_2) \in \Sigma_P$, $f_P(r_P, \sigma_P)!$ if and only if
  - $f(q_i, \sigma_i)!$ and $f_K(q_{K_i}, \sigma_i)!$ if $\sigma_i \neq \varepsilon$ $(i = 1, 2)$, and
  - $\lambda(q_1, \sigma_1) \cap \lambda(q_2, \sigma_2) \neq \emptyset$.

  If $f_P(r_P, \sigma_P)!$, then $f_P(r_P, \sigma_P) = (q_1', q_{K1}', q_2', q_{K2}')$, where

$$q_i' = \begin{cases} f(q_i, \sigma_i), & \text{if } \sigma_i \neq \varepsilon \\ q_i, & \text{otherwise,} \end{cases}$$

$$q_{Ki}' = \begin{cases} f_K(q_{Ki}, \sigma_i), & \text{if } \sigma_i \neq \varepsilon \\ q_{Ki}, & \text{otherwise,} \end{cases} \quad (i = 1, 2).$$

Remark 2: In the algorithm for verifying observability [5], three automata, that is, one copy of G and two copies of G_K, are composed to characterize the violation of observability. On the other hand, we compose two copies of G and two copies of G_K. In a Mealy automaton G, the next output symbol depends on the current state. In order to track two strings $s_1, s_2 \in K$ such that $O(s_1) \cap O(s_2) \neq \emptyset$, we need to know states reached by executing s_1 and s_2 in G. This is a reason why we need two copies of G to verify P-observability.

Theorem 5: A nonempty closed language $K \subseteq L(G)$ is not P-observable if and only if there exists a reachable state $r_P = (q_1, q_{K1}, q_2, q_{K2})$ of the testing automaton T_P and a controllable event $\sigma \in \Sigma_c$ such that $f_K(q_{K1}, \sigma)!$, $f(q_2, \sigma)!$, and $\neg f_K(q_{K2}, \sigma)!$.

By Theorem 5, P-observability is verified in polynomial-time with respect to $|Q|$ and $|Q_K|$.

B. Verification of A-Observability

We construct the synchronous composition $G \| G_K = (X, \Sigma, g, x_0)$ of G and G_K, where $X := Q \times Q_K$, $x_0 = (q_0, q_{K,0})$, and

$$g((q, q_K), \sigma) = \begin{cases} (f(q, \sigma), f_K(q_K, \sigma)), & \text{if } f(q, \sigma)! \text{ and } f_K(q_K, \sigma)! \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Then, $L(G \| G_K) = L(G) \cap L(G_K) = K$ holds. For a subset $X' \subseteq X$, we define $\varepsilon(X') \subseteq X$ inductively as follows:

- $X' \subseteq \varepsilon(X')$,
- $(\forall (q, q_K) \in \varepsilon(X'), \forall \sigma \in \Sigma)$

$$g((q, q_K), \sigma)! \wedge \varepsilon \in \lambda(q, \sigma) \Rightarrow g((q, q_K), \sigma) \in \varepsilon(X').$$

Also, we define $\Sigma_c(X') \subseteq \Sigma_c$ as

$$\Sigma_c(X') = \{\sigma \in \Sigma_c \mid \exists (q, q_K) \in X' : f(q, \sigma)!, \neg f_K(q_K, \sigma)!\}.$$

We construct an *observer* automaton $(G \| G_K)_O = (2^X - \{\emptyset\}, \Delta, h, \varepsilon(\{x_0\}))$ of $G \| G_K$. The transition function h :

$(2^X - \{\emptyset\}) \times \Delta \to (2^X - \{\emptyset\})$ is defined as follows: For each $X' \in 2^X - \{\emptyset\}$ and $\delta \in \Delta$, let

$$\begin{aligned} X'_\delta &= \{(q', q'_K) \in X \mid \exists (q, q_K) \in X', \sigma \in \Sigma : \\ &\quad g((q, q_K), \sigma) = (q', q'_K), \delta \in \lambda(q, \sigma)\}. \end{aligned}$$

Then

$$h(X', \delta) = \begin{cases} \varepsilon(X'_\delta), & \text{if } X'_\delta \neq \emptyset \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

For verifying A-observability, a finite automaton $T_A = (R_A, \Sigma_A, f_A, r_{A,0})$ is defined as follows:

- $R_A = Q \times (2^X - \{\emptyset\})$,
- $r_{A,0} = (q_0, \varepsilon(\{x_0\}))$,
- $\Sigma_A = \Sigma \times (\Delta \cup \{\varepsilon\})$,
- $f_A : R_A \times \Sigma_A \to R_A$ is defined as follows: For each $r_A = (q, X') \in R_A$ and $\sigma_A = (\sigma, \delta) \in \Sigma_A$, $f_A(r_A, \sigma_A)!$ if and only if
  - $f(q, \sigma)!$,
  - $h(X', \delta)!$ if $\delta \neq \varepsilon$,
  - $\delta \in \lambda(q, \sigma)$, and
  - $\sigma \notin \Sigma_c(X')$.

  If $f_A(r_A, \sigma_A)!$, then

$$f_A(r_A, \sigma_A) = \begin{cases} (f(q, \sigma), h(X', \delta)), & \text{if } \delta \neq \varepsilon \\ (f(q, \sigma), X'), & \text{otherwise.} \end{cases}$$

It can be verified that for the nonempty closed language $K \subseteq L(G)$, the anti-permissive supervisor S_A satisfies $L_e(S_A/G) = L(T_A)$. Then, we have the following theorem.

Theorem 6: Assume that a nonempty closed language $K \subseteq L(G)$ is controllable. K is A-observable if and only if

$$K = \{s \in \Sigma^* \mid \exists t \in L(T_A) : P_\Sigma(t) = s\}.$$

To verify A-observability using Theorem 6, we have to construct the observer automaton $(G \| G_K)_O$. The complexity of constructing $(G \| G_K)_O$ is exponential with respect to $|Q| \times |Q_K|$. It is an open problem whether A-observability can be tested in polynomial-time.

VI. Conclusion

In this paper, we studied a supervisory control problem for DESs modeled by Mealy automata with nondeterministic output functions. We introduced two kinds of supervisors: the permissive and anti-permissive supervisor. We presented necessary and sufficient conditions for each of the two supervisors to achieve a given specification language. However, even if these conditions are not satisfied, there may exist a supervisor that achieves the specification language. Such a supervisor cannot be synthesized by using the permissive nor anti-permissive policy. So the conditions presented in this paper are only sufficient for the existence of a supervisor. It is important future work to derive necessary and sufficient conditions for the existence of a supervisor.

References

[1] F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Inf. Sci.*, vol. 44, no. 3, pp. 173–198, 1988.

[2] R. Cieslak, C. Desclaux, A. S. Fawaz, and P. Varaiya, "Supervisory control of discrete event processes with partial observation," *IEEE Trans. Autom. Control*, vol. 33, no. 2, pp. 249–260, 1988.

[3] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Contr. Optim.*, vol. 25, no. 1, pp. 206–230, 1987.

[4] H. Cho and S. I. Marcus, "On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation," *Math. Control Signals Syst.*, vol. 2, no. 1, pp. 47–69, 1989.

[5] J. N. Tsitsiklis, "On the control of discrete-event dynamical systems," *Math. Control Signals Syst.*, vol. 2, no. 2, pp. 95–107, 1989.

[6] K. Inan, "Nondeterministic supervision under partial observation," in *Proc. 11th International Conf. Analysis and Optimization of Systems*, Lecture Notes in Control and Inf. Sci., No. 199, pp. 39–48, 1994.

[7] M. Heymann and F. Lin, "On-line control of partially observed discrete event systems," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 4, no. 3, pp. 221–236, 1994.

[8] F. Lin and W. M. Wonham, "Supervisory control of timed discrete-event systems under partial observation," *IEEE Trans. Autom. Control*, vol. 40, no. 3, pp. 558–562, 1995.

[9] N. Ben Hadj-Alouane, S. Lafortune, and F. Lin, "Centralized and distributed algorithms for on-line synthesis of maximal control policies under partial observation," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 6, no. 4, pp. 379–427, 1996.

[10] R. Kumar and M. A. Shayman, "Centralized and decentralized supervisory control of nondeterministic systems under partial observation," *SIAM J. Contr. Optim.*, vol. 35, no. 2, pp. 363–383, 1997.

[11] T. Ushio, "On-line control of discrete event systems with a maximally controllable and observable sublanguage," *IEICE Trans. Fundamentals*, vol. E82-A, no. 9, pp. 1965–1970, 1999.

[12] S. Takai, "Robust supervisory control of a class of timed discrete event systems under partial observation," *Syst. Control Lett.*, vol. 39, no. 4, pp. 267–273, 2000.

[13] S. Takai and T. Ushio, "Weak normality for nonblocking supervisory control of discrete event systems under partial observation," *IEICE Trans. Fundamentals*, vol. E84-A, no. 11, pp. 2822–2828, 2001.

[14] S. Takai and T. Ushio, "Effective computation of an $Lm(G)$-closed, controllable, and observable sublanguage arising in supervisory control," *Syst. Control Lett.*, vol. 49, no. 3, pp. 191–200, 2003.

[15] R. Kumar, S. Jiang, C. Zhou, and W. Qiu, "Polynomial synthesis of supervisor for partially observed discrete-event systems by allowing nondeterminism in control," *IEEE Trans. Autom. Control*, vol. 50, no. 4, pp. 463–475, 2005.

[16] S. Takai and T. Ushio, "Supervisory control of a class of concurrent discrete event systems under partial observation," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 15, no. 1, pp. 7–32, 2005.

[17] S. Takai and T. Ushio, "A new class of supervisors for timed discrete event systems under partial observation," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 16, no. 2, pp. 257–278, 2006.

[18] R. Kumar, V. K. Garg, and S. I. Marcus, "Predicates and predicate transformers for supervisory control of discrete event dynamical systems," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 232–247, 1993.

[19] Y. Li and W. M. Wonham, "Control of vector discrete-event systems–I:The base model," *IEEE Trans. Autom. Control*, vol. 38, no. 8, pp. 1214–1227, 1993.

[20] S. Takai, T. Ushio, and S. Kodama, "Static state feedback control of discrete-event systems under partial observation," *IEEE Trans. Autom. Control*, vol. 40, no. 11, pp. 1950–1954, 1995.

[21] S. Takai, T. Ushio, and S. Kodama, "Supervisory control of discrete event systems using partial event and state observations," *Int. J. Intelligent Control Syst.*, vol. 2, no. 3, pp. 453–466, 1998.

[22] K. R. Rohloff, "Sensor failure tolerant supervisory control," in *Proc. 44th IEEE Conf. Decision and Control, and European Control Conf. 2005*, pp. 3493–3498, 2005.

[23] G. H. Mealy, "A method to synthesizing sequential circuits," *Bell System Technical J.*, vol. 34, no. 5, pp. 1045–1079, 1955.

[24] T.-S. Yoo and S. Lafortune, "A general architecture for decentralized supervisory control of discrete-event systems," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 12, no. 3, pp. 335–377, 2002.