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 function 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 or 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 $\varepsilon$. 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 $\pi := \{s\}$. 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, $\Sigma$ is the finite set of events, $\Delta$ is the set of output symbols, $f : Q \times \Sigma \rightarrow Q$ is the partial...
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 $\lambda$, 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, $-f(q, s)$ denotes the negation of $f(q, s)$. The generated language $L(G)$ of $G$ is defined by $L(G) = \{s \in \Sigma^* | f(q_0, s)\}$.

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:

- $O(\varepsilon) = \{\varepsilon\}$,
- $(\forall \sigma \in L(G) \ (\sigma \in \Sigma))$
  
  $$O(\sigma) = \{\tau \delta \in \Delta^* | \tau \in O(s), \delta \in \lambda(f(q_0, s), \sigma)\}.$$ 

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

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

That is, $O^{-1}(\tau)$ is the set of event strings for which $\tau$ 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 $M(q, \sigma) = \{M(\sigma)\}$. So our framework using the nondeterministic 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 $\varepsilon = (\varepsilon, \varepsilon)$. Let $t = a_1a_2\ldots a_n \in (\Sigma \times (\Delta \cup \{\varepsilon\}))^*$ 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

- $P_\Sigma(t) = \sigma_1\sigma_2\ldots \sigma_n$,
- $P_\Delta(t) = \delta_1\delta_2\ldots \delta_n$.

For each event string $s = \sigma_1\sigma_2\ldots \sigma_n \in L(G)$, an extended event string $t = a_1a_2\ldots a_n \in (\Sigma \times (\Delta \cup \{\varepsilon\}))^*$ 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
- $P_\Delta(a_i) \in \lambda(f(q_0, \sigma_1\sigma_2\ldots \sigma_{i-1}), \sigma_i)$ ($i = 2, 3, \ldots, n$).

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 $\Pi(\varepsilon) = \{\varepsilon\}$ 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 $\Sigma$ is partitioned into the controllable event set $\Sigma_c$ and the uncontrollable event set $\Sigma_u$, that is, $\Sigma = \Sigma_c \cup \Sigma_u$ [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 $\tau$. 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:

- $\varepsilon \in L_e(S/G)$,
- $(\forall t \in L_e(S/G), a \in \Sigma \times (\Delta \cup \{\varepsilon\}))$
  
  $$ta \in L_e(S/G) \iff ta \in L_e(S/G) \land P_\Sigma(a) \notin S(P_\Delta(t)).$$

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

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

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

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

(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-Permissible 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 $\sigma \in \Sigma_c$,

$$O^{-1}(\tau) \cap K \neq \emptyset \implies \sigma \notin S(\tau).$$
Also, it is said to be anti-permissive if, for each \( \tau \in O(L(G)) \) and \( \sigma \in \Sigma_c \),
\[
(O^{-1}(\tau) \cap K)(\sigma) \cap (L(G) - K) \neq \emptyset \iff \sigma \in S(\tau).
\]
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
\[
(O^{-1}(\tau) \cap K)(\sigma) \cap (L(G) - K) \neq \emptyset \iff \sigma \notin S_A(\tau).
\]
We then have the following proposition whose proof is straightforward.

Proposition 1: For a nonempty closed controllable language \( K \subseteq L(G) \),
\[
L(S_A/G) \subseteq K \subseteq L(S_P/G).
\]  

C. Supervisory Control under the Mask Function

We review the results on the conventional supervisory control under partial event observation. Let \( M : \Sigma \rightarrow \Delta \cup \{ \varepsilon \} \) 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:
\[
\lambda(q, \sigma) = \{ M(\sigma) \}.
\]
A closed language \( K \subseteq L(G) \) is said to be observable [1], [2] (with respect to \( M, \Sigma_c, \) and \( L(G) \)) if, for any \( s, s' \in K \) and \( \sigma \in \Sigma_c \),
\[
M(s) = M(s') \wedge s' \in K \Rightarrow s \in K.
\]
For these supervisors, we have \( L(S'/G) = L(G) \) and \( L(S_A/G) = K \). Only the anti-permissive supervisor achieves the specification in this example.

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 \( \sigma \in \Sigma_c \),
\[
O^{-1}(\tau)(\sigma) \neq \emptyset \implies [O^{-1}(\tau)(\sigma) \cap K] \neq K.
\]

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
\[
O^{-1}(\tau)(\sigma) \neq \emptyset, \quad (O^{-1}(\tau)(\sigma) \cap K) \neq K.
\]

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

By the definition of \( S_P \), we have \( \sigma' \notin S_P(\sigma') \). It follows that \( t'a \in L_G(S_P/G) \). Thus, we have \( t \in L_G(S_P/G) \). Further, since \( \sigma \notin S_P(\tau) \) by (5), we have \( t(\sigma, \delta) \in L_G(S_P/G) \), where \( \delta \in L_G(f(q_0, s), \sigma) \). So we have \( s = L_G(S_P/G) = K \), 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 \( \varepsilon \in L(G) \cap \triangleleft \).
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 \( t\alpha \in L_e(S_P/G) \) such that \( P_2(t) = s \) and \( P_2(\alpha) = \sigma \). Since \( t\alpha \in L_e(S_P/G) \), we have \( \sigma \not\in S_P(P_\Delta(t)) \). By the definition of \( S \), 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

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

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 \( P_2(t) = s \). We have \( s\sigma \in O^{-1}(P_\Delta(t)) \{ \sigma \} \cap K \neq \emptyset \), which implies that \( \sigma \not\in S_P(P_\Delta(t)) \). It follows that \( t(\alpha, \delta) \in L_e(S_P/G) \), where \( \delta \in \lambda(f(\theta_0, \sigma)) \). So we have \( s\sigma \in L(S_P/G) \). Thus, \( K \subseteq L(S_P/G) \).

**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 \( S \). In this sense, Theorem 2 can be viewed as a generalization of the result of [22].

### B. Anti-Permissive Supervisor

We note that the problem of synthesizing an anti-permissive supervisor \( S_A \) for the sensor fault tolerant supervisory control problem is not a simple one. 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 \( t' \in \tau \) and \( \sigma' \in \Sigma_c \) satisfying

\[
P_2(t')\sigma' \in \overline{\sigma\sigma},
\]

\[
O^{-1}(P_\Delta(t')) \cap K \{ \sigma' \} \cap (L(G) - K) \neq \emptyset.
\]

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

We next suppose that \( K \) is controllable and \( A \)-observable. Since \( K \) is nonempty and closed, we have \( \epsilon \in L(S_A/G) \cap K \). 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 we have \( s \sigma \in K \). We consider the case that \( \sigma \in \Sigma_c \). Since \( s \sigma \in L(S_A/G) \), there exists an extended event string \( t\alpha \in L_e(S_A/G) \) such that \( P_2(t) = s \) and \( P_2(\alpha) = \sigma \). Since \( t\alpha \in L_e(S_A/G) \), 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 \( t' \in \tau \) and \( \sigma' \in \Sigma_c \) with \( P_2(t')\sigma' \in \overline{\sigma\sigma} \) satisfy (7). We have \( t\alpha \in L_e(S_A/G) \) and \( \sigma \notin S_A(P_\Delta(t)) \), which implies that \( s\sigma \in L(S_A/G) \). Thus, \( K \subseteq L(S_A/G) \).

### 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 \( K \) 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 \( \Sigma = \Sigma_e = \{ \sigma_1, \sigma_2 \} \) 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 \( L(S/G) = K \).

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

Note that this supervisor \( S \) cannot be synthesized by using the permissive nor anti-permissive policy.
In a special case that the output function $\lambda$ 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 $\lambda : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \Delta \cup \{\varepsilon\}$ is deterministic. Then, $K$ is P-observable if and only if it is A-observable. Further, there exists a supervisor $S : O(L(G)) \rightarrow 2^\Sigma_c$ 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 $\lambda$ is deterministic, we have $|\Pi(s)| = 1$. Let $\{t\} = \Pi(s)$. By A-observability of $K$, we have

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

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 $S : O(L(G)) \rightarrow 2^\Sigma_c$ 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 $s\sigma \in K$. Since the output function $\lambda$ is deterministic, we have $s\sigma \notin S(\tau)$. Consider any $s'\sigma \in (O^{-1}(\tau) \cap K)\{\sigma\} \cap L(G)$. Since the output function $\lambda$ 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 $G_K = (Q_K, \Sigma_c, r_K, q_{K,0})$ (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:

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

If $f_P(r_P, \sigma) !$, then $f_P(r_P, \sigma) = (q'_1, q'_{K,1}, q'_2, q'_{K,2})$, where

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

$$q'_{K,i} = \begin{cases} f_K(q_{K,i}, \sigma_i), & \text{if } \sigma_i \notin \varepsilon \\ q_{K,i}, & \text{otherwise} \end{cases} (i = 1, 2).$$

**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, 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 $r_P = (q_1, q_{K,1}, q_2, q_{K,2})$ of the testing automaton $T_P$ and a controllable event $\sigma \in \Sigma_c$ such that $f_K(q_{K,1}, \sigma) !$, $f(q_2, \sigma) !$, and $\lambda(q_1, \sigma) \cap \lambda(q_2, \sigma) \neq \emptyset$.

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

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

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

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

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

$$\Sigma_c(X') = \{ \sigma \in \Sigma_c : \exists(q, q_K) \in X' : f(q, \sigma) !, -f_K(q_K, \sigma) \}.$$
\((2^X - \{\emptyset\}) \times \Delta \rightarrow (2^X - \{\emptyset\})\) is defined as follows: For each \(X' \in 2^X - \{\emptyset\}\) and \(\delta \in \Delta\), let
\[
X'_\delta = \{ (q', q'_K) \in X \mid \exists (q, q_K) \in X', \sigma \in \Sigma : g((q, q_K), \sigma) = (q', q'_K), \delta \in \lambda(q, \sigma) \}.
\]
Then
\[
h(X', \delta) = \begin{cases} 
\epsilon(X'_\delta), & \text{if } X'_\delta \neq \emptyset \\
\text{undefined}, & \text{otherwise}.
\end{cases}
\]

For verifying A-observability, a finite automaton \(T_A = (R_A, \Sigma_A, f_A, r_{A,0})\) is defined as follows:
- \(R_A = Q \times (2^X - \{\emptyset\})\),
- \(r_{A,0} = (q_0, \emptyset)\).
- \(\Sigma_A = \Sigma \times (\Delta \cup \{\emptyset\})\),
- \(f_A : R_A \times \Sigma_A \rightarrow R_A\) is defined as follows: For each \(r_A = (q, X') \in R_A\) and \(\sigma, \delta \in \Sigma_A\),
\[
f_A(r_A, \sigma \delta) = \begin{cases} 
\emptyset, & \text{if and only if } \delta = \emptyset, \\
\{ (q, \emptyset) \}, & \text{otherwise}.
\end{cases}
\]

It can be verified that for the nonempty closed language \(K \subseteq L(G)\), the anti-permissive supervisor \(S_{L_{\epsilon}}\) satisfies \(L_{\epsilon}(S_{A}/G) = L(T_A)\). 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
\[
K = \{ s \in \Sigma^* \mid \exists t \in L(T_A) : P_{21}(t) = s \}.
\]

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