On Observability of Discrete-Event Systems

F. LIN

and

W. M. WONHAM

Systems Control Group, Department of Electrical Engineering,
University of Toronto, Toronto, Ontario M5S 1A4, Canada

ABSTRACT

The observability of discrete-event systems is investigated. A discrete-event system \( G \) is modeled as the controlled generator of a formal language \( L_m(G) \) in the framework of Ramadge and Wonham. To control \( G \), a supervisor \( S \) is developed whose action is to enable and disable the controllable events of \( G \) according to a record of occurrences of the observable events of \( G \), in such a way that the resulting closed-loop system obeys some prespecified operating rules embodied in a given language \( K \). A necessary and sufficient condition is found for the existence of a supervisor \( S \) such that \( L_m(S/G) = K \). Based on this condition, a solution of the supervisory control and observation problem (SCOP) is obtained. Two examples are provided.

1. INTRODUCTION

In this paper, we investigate the observability of discrete event systems. The paper is set in the framework of [8] and subsequent articles, with which the reader is assumed to be familiar.

The discrete-event system (DES) to be controlled (i.e. the plant) is modeled by an automaton

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

where \( \Sigma \) is the alphabet of event labels, \( Q \) is the set of states, \( \delta : \Sigma \times Q \rightarrow Q \) is the transition function (a partial function), \( q_0 \) is the initial state, and \( Q_m \subseteq Q \) is the set of marked states. The behavior of the DES is described by the language...
generated by $G$: \(^1\)

$$L(G) := \{ s : s \in \Sigma^* \text{ and } \delta(s, q_0) \}$$

or by the language marked by $G$:

$$L_m(G) := \{ s : s \in L(G) \text{ and } \delta(s, q_0) \in Q_m \}.$$  

We interpret $\Sigma$ as the set of events. Hence the plant generates a stream of events. Some of these events are controllable, i.e., their occurrence can be prevented; and some of these events are observable, i.e., their occurrence can be observed. Thus $\Sigma$ is partitioned as

$$\Sigma = \Sigma_c \cup \Sigma_{uc} = \Sigma_o \cup \Sigma_{uo}$$

where $\Sigma_c$ is the set of controllable events, $\Sigma_{uc}$ is the set of uncontrollable events, $\Sigma_o$ is the set of observable events, and $\Sigma_{uo}$ is the set of unobservable events.

An automaton $G$ over $\Sigma$ can be extended to an automaton $G'$ over $\Sigma'$, with $\Sigma \subseteq \Sigma'$, as

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

The transition function $\delta' : \Sigma' \times Q \rightarrow Q$ is defined by

$$\delta'(\sigma, q) = \begin{cases} \delta(\sigma, q) & \text{if } \sigma \in \Sigma, \\ q & \text{if } \sigma \in \Sigma' - \Sigma. \end{cases}$$

Such a transition function $\delta'$ will be called $(\Sigma' - \Sigma)$-null.

Our objective is to design a controller (or supervisor) whose task is to enable and disable the controllable events based on a record of occurrences of the observable events such that the resulting closed-loop system obeys some pre-specified operating rules. Formally, a supervisor is a pair

$$S = (S, \psi),$$

where $S = (\Sigma, X, \xi, x_0, X_m)$ is an automaton in which

$$\xi : \Sigma \times X \rightarrow X \ (\text{pfn}) \text{ is } \Sigma_{uo}\text{-null},$$

\(^1\delta(s, q_0)! \text{ means that } \delta(s, q_0) \text{ is defined.}\)
and \( \psi: \Sigma \times X \rightarrow \{0,1\} \) is the feedback map, satisfying

\[
\psi(\sigma, x) = 1 \quad \text{if} \quad \sigma \in \Sigma_{uc}, \quad x \in X, \\
\psi(\sigma, x) \in \{0,1\} \quad \text{if} \quad \sigma \in \Sigma_{c}, \quad x \in X.
\]

\( S \) is considered to be driven externally by the stream of event symbols generated by \( G \); while in turn, with \( S \) in state \( x \), the events \( \sigma \) of \( G \) are subject to the control \( \psi(\sigma, x) \). If \( \psi(\sigma, x) = 0 \), then \( \sigma \) is "disabled" (prohibited from occurring), while if \( \psi(\sigma, x) = 1 \), then \( \sigma \) is "enabled" (permitted but not forced to occur). In this way there results a closed-loop feedback structure \( S/G \), called the supervised discrete event system. \( S/G \) is itself an automaton

\[
S/G = (\Sigma, X \times Q, (\xi \times \delta)^{\psi}, (x_0, q_0), X_m \times Q_m),
\]

where \((\xi \times \delta)^{\psi}: \Sigma \times X \times Q \rightarrow X \times Q\) is defined as \(^2\)

\[
(\xi \times \delta)^{\psi}(\sigma, x, q) = \begin{cases} 
(\xi(\sigma, x), \delta(\sigma, q)) & \text{if } \psi(\sigma, x) = 1, \\
\text{undefined} & \text{if } \psi(\sigma, x) = 0.
\end{cases}
\]

Following [8], we say that a supervisor \( S = (S, \psi) \) for \( G \) is complete if the three conditions

1. \( s \in L(S/G), \)
2. \( s\sigma \in L(G), \) i.e. \( \delta(s\sigma, q_0)!, \)
3. \( \psi(\sigma, \xi(s, x_0)) = 1, \) i.e. \( \sigma \) is enabled at \( \xi(s, x_0), \)

together imply that

4. \( s\sigma \in L(S/G), \) i.e. \( \xi(s\sigma, x_0)!. \)

It will always be assumed (or guaranteed) that \( S \) is complete. The behavior of \( S/G \) is described by the languages \( L(S/G) \) and \( L_m(S/G) \). A (complete) supervisor \( S \) is proper if

\[
\overline{L_m(S/G)} = L(S/G).
\]

\(^2:\) \((\xi(\sigma, x), \delta(\sigma, x))! \) iff both \( \xi(\sigma, x)! \) and \( \delta(\sigma, q)! \).
We shall need the following notation and definitions. If \( X \) is a set and \( \theta \) an equivalence relation on \( X \), then for \( x \in X \) we write \( \theta[x] \) for the coset (equivalence class) of \( x \mod \theta \). The index of \( \theta \) (cardinality of the quotient set \( X/\theta \)) will be written \( ||\theta|| \).

Let \( \pi \) and \( \rho \) be two binary relations on \( X \). We write \( \pi \leq \rho \) (\( \pi \) refines \( \rho \)) if

\[
(\forall x, y \in X) \quad (x, y) \in \pi \Rightarrow (x, y) \in \rho.
\]

Let \( E(X) \) denote the lattice of all equivalence relations on \( X \), under the partial ordering \( (\leq) \). If \( \theta, \theta' \in E(X) \), the meet \( \theta \wedge \theta' \in E(X) \) is given by

\[
(x, y) \in \theta \wedge \theta' \text{ iff } (x, y) \in \theta \text{ and } (x, y) \in \theta',
\]

and the join \( \theta \lor \theta' \in E(X) \) is given by

\[
(x, y) \in \theta \lor \theta'
\]

iff there exist \( k \geq 0 \) and \( z_1, z_2, \ldots, z_k \in X \) such that

\[
(x, z_1) \in \theta \text{ or } (x, z_1) \in \theta',
\]

\[
(z_1, z_2) \in \theta \text{ or } (z_1, z_2) \in \theta',
\]

\[
\vdots
\]

\[
(z_k, y) \in \theta \text{ or } (z_k, y) \in \theta'.
\]

An equivalence relation \( \theta \) on \( \Sigma^* \) is a right congruence if\(^3\)

\[
(\forall s, s', t \in \Sigma^*) \quad s \equiv s' \pmod{\theta} \Rightarrow st \equiv s't \pmod{\theta}.
\]

Given an automaton \( G = (\Sigma, Q, \delta, q_0, Q_m) \), we define a right congruence \( \text{eq}(G) \) on \( \Sigma^* \), called the equiresponse relation of \( G \) [4], by\(^4\)

\[
s \equiv s' \pmod{\text{eq}(G)} \text{ iff } \delta(s, q_0) = \delta(s', q_0).
\]

Conversely, if \( \theta \) is a right congruence on \( \Sigma^* \), then the following automaton \( G \) can be defined:

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

\(^3\)If \( \theta \) is an equivalence relation, we may write \( s \equiv s' \pmod{\theta} \) for \( (s, s') \in \theta \).

\(^4\)If \( \delta(s, q_0) \) is undefined, then \( \delta(s, q_0) = \delta(s', q_0) \) iff \( \delta(s', q_0) \) is also undefined.
where

\[ Q := \{ \theta[s] : s \in \Sigma^* \}, \]

\[ \delta(\sigma, \theta[s]) = \theta[s\sigma], \]

\[ q_0 = \theta[1]. \]

Clearly, \( \text{eq}(G) = \theta \); and \( Q_m \) can be designated arbitrarily.

Let \( \Sigma_o \subseteq \Sigma \). The projection \( P : \Sigma^* \to \Sigma_o^* \) is given by

\[ P1 = 1, \]

\[ Ps = s - \Sigma_o, \quad \sigma \in \Sigma - \Sigma_o, \]

\[ Ps = s, \quad \sigma \in \Sigma_o, \]

\[ P(s\sigma) = P(s)P(\sigma), \quad s \in \Sigma^*, \quad \sigma \in \Sigma. \]

For \( s, s' \in \Sigma^* \) define \( s \equiv s' \pmod{\ker P} \) if \( Ps = Ps' \). Clearly \( \ker P \) is a right congruence on \( \Sigma^* \).

For a language \( L \subseteq \Sigma^* \), the right congruence \( \equiv \pmod{L} \), viz. Nerode equivalence, is defined by

\[ s \equiv s' \pmod{L} \iff (\forall t \in \Sigma^*) \quad st \in L \iff s't \in L. \]

Given \( K \subseteq L_m(G) \), for \( s \in \Sigma^* \) define the active event set

\[ A_K(s) := \begin{cases} \{ \sigma : s\sigma \in \bar{K} \} & \text{if } s \in \bar{K}, \\ \emptyset & \text{otherwise}, \end{cases} \]

and the inactive event set

\[ IA_K(s) := \begin{cases} \{ \sigma : s\sigma \in L(G) - \bar{K} \} & \text{if } s \in \bar{K}, \\ \emptyset & \text{otherwise}. \end{cases} \]

Finally, the binary relation \( \text{act}_K \) on \( \Sigma^* \) is defined as \( (s, s') \in \text{act}_K \) iff

(i) \( A_K(s) \cap IA_K(s') = \emptyset = A_K(s') \cap IA_K(s) \), and
(ii) \( s \in \bar{K} \cap L_m(G) \wedge s' \in \bar{K} \cap L_m(G) \Rightarrow (s \in K \iff s' \in K) \).
Equivalently, for all $s, s' \in \Sigma^*$, $(s, s') \in \text{act}_K$ iff

(i) $(\forall \sigma) \; s\sigma \in \overline{K} \land s' \in \overline{K} \land s'\sigma \in L(G) \Rightarrow s'\sigma \in \overline{K}$, and
(ii) $s \in K \land s' \in \overline{K} \land L_m(G) = s' \in K$, and
(iii) (i) and (ii) hold with $s$ and $s'$ interchanged.

We note that $\text{act}_K$ is a tolerance relation on $\Sigma^*$, i.e. is reflexive and symmetric but not transitive.

2. OBSERVABILITY

In this section, we obtain a necessary and sufficient condition for the existence of a supervisor $S$ for $G$ such that $L_m(S/G) = K$. We assume that $G$ is trim, i.e., $G$ is reachable and

$$\overline{L_m(G)} = L(G).$$

We first prove the following lemma.

**Lemma 2.1.** Given a language $K$, $\emptyset \neq K \subseteq L_m(G)$. There exists a proper supervisor $S$ such that $L_m(S/G) = K$ if and only if the following conditions are both satisfied:

(1) $\overline{K} \Sigma_{uc} \cap L(G) \subseteq \overline{K}$.
(2) There exists a right congruence $\gamma$ such that

$$\ker P \leqslant \gamma \leqslant \text{act}_K.$$

**Proof.** "Only if": Let $S = (S, \psi)$ be a proper supervisor such that $L_m(S/G) = K$. By Theorem 6.1 of [8],

$$\overline{K} \Sigma_{uc} \cap L(G) \subseteq \overline{K}.$$

Define $\gamma = \text{eq}(S')$. Since $\xi$ is $\Sigma_{uo} - \text{null},$

$$\ker P \leqslant \gamma.$$

Finally, to show that $\gamma \leqslant \text{act}_K$, let $s = s' \pmod{\gamma}$. If either $\xi(s, x_0)$ or $\xi(s', x_0)$ is undefined, then $(s, s') \in \text{act}_K$ is automatic. Otherwise we have $s, s' \in \overline{K}$ and write $x = \xi(s, x_0) = \xi(s', x_0)$. 

(i) We claim that $A_K(s) \cap IA(s') = \emptyset$. Otherwise

$$(\exists \sigma \in \Sigma) \quad s\sigma \in \overline{K} \land s'\sigma \in \overline{K} \land s'\sigma \in L(G) - \overline{K}.$$ 

If $\psi(\sigma, x) = 1$, then $s'\sigma \in L(S/G) \land s'\sigma \not\in \overline{K}$, and therefore $L(S/G) \neq \overline{K}$; whereas if $\psi(\sigma, x) = 0$, then $s\sigma \in L(S/G) \land s\sigma \in \overline{K}$, and therefore $L(S/G) \neq \overline{K}$. But then $L_m(S/G) \neq K$, a contradiction. Similarly, $A_K(s') \cap IA_K(s) = \emptyset$.

(ii) If $s \in \overline{K} \cap L_m(G) \land s' \in \overline{K} \cap L_m(G)$ then $s' \in \overline{K} \subseteq L(S/G)$, so that $\delta(s', q_0)$ and $\xi(s', x_0)$, and then

$$s \in K \implies x = \xi(s, x_0) - \xi(s', x_0) \in X_m,$$

$$s' \in L_m(G) \implies s' \in L(G) \subseteq \emptyset,$$

hence $s' \in K$. Similarly $s' \in K \implies s \in K$. Hence

$$s \in \overline{K} \cap L_m(G) \land s' \in \overline{K} \cap L_m(G) \implies (s \in K \iff s' \in K).$$

"If": Let $\gamma$ be a right congruence such that $\ker P \leq \gamma \leq \text{act}_K$; and let $S = (\Sigma, X, \xi, x_0, X_m)$ be the automaton with $\text{eq}(S) = \gamma$ and with

$$X_m = \{ x : x \in X \land (\exists s \in K \land \xi(s, x_0) = x) \}.$$

For all $x \in X$, let

$$\Sigma_x^0 := \bigcup \{ IA_K(s) : \xi(s, x_0) = x \},$$

$$\Sigma_x^1 := \bigcup \{ A_K(s) : \xi(s, x_0) = x \}.$$

Then $\gamma \leq \text{act}_K$ implies $\Sigma_x^0 \cap \Sigma_x^1 = \emptyset$; furthermore

$$\overline{K} \Sigma_{uc} \cap L(G) \subseteq \overline{K} \implies \Sigma_x^0 \subseteq \Sigma_c.$$

So we can define $\psi : \Sigma \times X \to \{0, 1\}$ to be any function such that

$$\psi(\sigma, x) = 1 \quad \text{if} \quad \sigma \in \Sigma_x^1,$$

$$\psi(\sigma, x) = 0 \quad \text{if} \quad \sigma \in \Sigma_x^0.$$

Because $\xi$ is a total function, $S = (S, \psi)$ is a complete supervisor for $G$. 
Next we show that \( L(S/G) = \bar{K} \), using induction on the length of words. For \( \sigma \in \Sigma \),

\[
\sigma \in \bar{K} \quad \Rightarrow \quad \sigma \in \text{L}(G) \land \sigma \in \text{A}_K(1) \quad (\bar{K} \subseteq \text{L}(G))
\]
\[
\Rightarrow \quad \sigma \in \text{L}(G) \land \psi(\sigma, x_0) = 1 \quad \text{(definition of } \psi) \]
\[
\Rightarrow \quad \sigma \in \text{L}(S/G).
\]

On the other hand,

\[
\sigma \in \text{L}(S/G) \quad \Rightarrow \quad \sigma \in \text{L}(G) \land \psi(\sigma, x_0) = 1
\]
\[
\Rightarrow \quad \sigma \in \text{L}(G) \land \sigma \notin \text{IA}_K(1) \quad \text{(definition of } \psi) \]
\[
\Rightarrow \quad \sigma \in \text{L}(G) \land \sigma \notin \text{L}(G) - \bar{K} \quad (1 \in \bar{K})
\]
\[
\Rightarrow \quad \sigma \in \bar{K}.
\]

Now suppose that for all \( s, |s| \leq n, s \in \text{L}(S/G) \Leftrightarrow s \in \bar{K} \). Fix \( s \) with \( |s| = n \) and write \( x = \xi(s, x_0) \). For \( \sigma \in \Sigma \),

\[
s\sigma \in \bar{K} \quad \Rightarrow \quad s\sigma \in \text{L}(G) \land \sigma \in \text{A}_K(s) \land s \in \bar{K}
\]
\[
\Rightarrow \quad s\sigma \in \text{L}(G) \land \sigma \in \text{A}_K(s) \land s \in \text{L}(S/G)
\quad \text{(induction hypothesis)}
\]
\[
\Rightarrow \quad s\sigma \in \text{L}(G) \land \psi(\sigma, x) = 1 \land s \in \text{L}(S/G)
\quad \text{(definition of } \psi)
\]
\[
\Rightarrow \quad s\sigma \in \text{L}(S/G).
\]

Conversely,

\[
s\sigma \in \text{L}(S/G)
\]
\[
\Rightarrow \quad s \in \text{L}(S/G) \land s\sigma \in \text{L}(G) \land \psi(\sigma, x) = 1
\]
\[
\Rightarrow \quad s \in \bar{K} \land s\sigma \in \text{L}(G) \land \sigma \notin \text{IA}_K(s)
\quad \text{(induction hypothesis)}
\]
\[
\Rightarrow \quad s \in \bar{K} \land s\sigma \in \text{L}(G) \land s\sigma \notin \text{L}(G) - \bar{K}
\quad \text{and definition of } \psi)
\]
\[
\Rightarrow \quad s\sigma \in \bar{K}.
\]
Finally, to prove that $S$ is proper, we must show that $L_m(S/G) = K$. We have

\[ s \in L_m(S/G) \Rightarrow s \in L(S/G) \land s \in L_m(G) \land \xi(s, x_0) \in X_m \]

\[ \Rightarrow s \in L(S/G) \land s \in L_m(G) \land (\exists s') \ (s' \in K \land \xi(s, x_0) = \xi(s', x_0)) \]

(by definition of $X_m$)

\[ \Rightarrow s \in L(S/G) \land s \in L_m(G) \land (\exists s') \ (s' \in K \land s \equiv s' \ (\text{mod } \gamma)) \]

\[ \Rightarrow s \in L(S/G) \land s \in L_m(G) \land (\exists s') \ s' \in K \land s' \in L_m(G) \land (s, s') \in \text{act}_K. \]

But $s' \in K \land s \in \bar{K} \cap L_m(G) \land (s, s') \in \text{act}_K \Rightarrow s \in K$ by condition (ii) of the definition of $\text{act}_K$. On the other hand

\[ s \in K \Rightarrow s \in \bar{K} \land s \in K \cap L_m(G) \]

\[ \Rightarrow s \in L(S/G) \land \xi(s, x_0) \in X_m \land s \in L_m(G) \] (definition of $X_m$)

\[ \Rightarrow s \in L_m(S/G) \]

Recall from [8] that a language $K \subseteq L_m(G)$ is controllable [with respect to $L(G)$] if

\[ \bar{K} \Sigma_{uc} \cap L(G) \subseteq \bar{K}. \]

We shall define a language $K \subseteq L_m(G)$ to be $P$-observable [with respect to $L(G)$], or simply observable, if

\[ \ker P \leq \text{act}_K. \]

The following theorem is now immediate.

**Theorem 2.1.** Let $K$ be a nonempty sublanguage of $L_m(G)$. There exists a proper supervisor $S$ such that $L_m(S/G) = K$ if and only if $K$ is both controllable and observable.
Proof. By Lemma 2.1 and the fact that \( \ker P \) is a right congruence.

We have already shown that a supervisor \( S = (S, \psi) \) can be constructed from a right-congruence \( \gamma \) on \( \Sigma^* \) such that

\[
\ker P \leq \gamma \leq \text{act}_K.
\]

Since there could be many such right congruences, supervisors are not unique. Let us define

\[
\Gamma := \{ \gamma : \gamma \text{ is a right congruence such that } \ker P \leq \gamma \leq \text{act}_K \}.
\]

Because \( \gamma_1 \in \Gamma \) and \( \gamma_2 \in \Gamma \) do not imply \( \gamma_1 \vee \gamma_2 \in \Gamma \), we cannot conclude that \( \Gamma \) has a supremal element. But the following result implies that supervisors exist that are "optimal" in an evident weak sense.

**Theorem 2.2.** \( \Gamma \) has at least one maximal element.

*Proof. Since \( \ker P \) is a right congruence, \( \Gamma \neq \emptyset \). Because every chain \( C \subseteq \Gamma \) has an upper bound in \( \Gamma \), by Zorn's lemma there exists a maximal element of \( \Gamma \).*

Our next result casts additional light on the structure of supervisors, but will not be used in the sequel.

**Theorem 2.3.** For all \( S = (S, \psi) \) such that \( L(S/G) = \overline{K} \),

\[
(\text{eq}(S) \wedge \equiv_{L(G)})|\overline{K} \leq (\equiv_{\overline{K}})|\overline{K}.
\]

*Proof. For simplicity we write \((\bmod \ {\overline{K}})\) etc. in place of \((\bmod \equiv_{\overline{K}})\). Let \( s, s' \in \overline{K} \). If \( s \equiv s' \ (\text{mod } \text{eq}(S)) \) and \( s \equiv s' \ (\text{mod } L(G)) \) but \( s \not\equiv s' \ (\text{mod } \overline{K}) \), then

\[
(\exists t \in \Sigma^*) \quad |t| > 1 \land st \in \overline{K} \land s't \not\in \overline{K}.
\]

Let \( t'\sigma \) be the longest prefix of \( t \) such that

\[
s't' \in \overline{K} \land s't'\sigma \not\in \overline{K}.
\]
In particular \( st' \sigma \in L(G) \). Since \( \text{eq}(S) \) and \( \equiv_{L(G)} \) are right congruences,

\[
st' = s't' \mod \text{eq}(S) \quad \text{and} \quad st' \sigma = s't' \sigma \mod L(G);
\]

hence we can write \( x = \xi(st', x_0) = \xi(s't', x_0) \) and \( s't' \sigma \in L(G) \). By definition of \( A_K(st') \) and \( IA_K(s't') \)

\[
st' \sigma \in \overline{K} \Rightarrow \sigma \in A_K(st')
\]

and

\[
s't' \in \overline{K} \land s't' \sigma \in L(G) \land s't' \sigma \not\in \overline{K} \Rightarrow \sigma \in IA_K(s't').
\]

Finally,

\[
\sigma \in A_K(st') \land L(S/G) = \overline{K} \Rightarrow \psi(\sigma, x) = 1,
\]

\[
\sigma \in IA_K(s't') \land L(S/G) = \overline{K} \Rightarrow \psi(\sigma, x) = 0.
\]

With this contradiction the result is proved. \( \square \)

To conclude this section, we show how to deal with a nondeterministic generator \( G \). It has been assumed so far that distinct transitions of \( G \) out of any state carry distinct event labels \( (\delta : \Sigma \times Q \rightarrow Q) \). However, the “noisy” or nondeterministic case \( (\delta : \Sigma \times Q \rightarrow 2^Q) \), where labels needn’t be distinct, can be brought within our framework by introducing new unobservable events.

For example, suppose we have a nondeterministic transition \( \alpha \) as shown below:

![Diagram](attachment:image.png)

Then we can introduce two new unobservable events \( \alpha_1, \alpha_2 \) and replace the
previous transitions by the following:

Because $\alpha_1$, $\alpha_2$ are unobservable, from the supervisor's point of view the transition $\alpha$ is still nondeterministic. However, in the formal description of G the transitions $\alpha$, $\alpha_1$, $\alpha_2$ are now deterministic.

3. SUPERVISORY CONTROL AND OBSERVATION PROBLEM

Let $\Sigma_{co} = \Sigma_c \cup \Sigma_o$, and let $T$ be the projection $\Sigma \to \Sigma_{co}^*$, i.e., $T$ erases all the events that are neither controllable nor observable (hence cannot be manipulated or processed by the supervisor). Since in this section only the language $L(S/G)$ will be synthesized, we assume that all the languages introduced are closed. Following [8], let languages $A, E \subseteq \Sigma_{co}^*$ be given with

$$A \subseteq E \subseteq TL(G).$$

We interpret $E$ as "legal behavior" (i.e., each word of $E$ is a "legal task"), and $A$ as "minimal acceptable behavior" (i.e., control of the plant G in such a way that a language smaller than $A$ is generated is considered inadequate). The condition $A, E \subseteq \Sigma_{co}^*$ amounts to assuming that a specification involving the uncontrollable and unobservable events can typically be restated as a specification involving only the controllable or observable events. We now introduce the

SUPERVISORY CONTROL AND OBSERVATION PROBLEM (SCOP). Construct a supervisor $S$ for $G$ such that

$$A \subseteq TL(S/G) \subseteq E.$$
To discuss the solvability of SCOP, we define, for a (closed) language \( L \subseteq TL(G) \subseteq \Sigma_{co}^* \),

\[
C(L) := \{ K : K \subseteq L \text{ and } K \text{ is closed and controllable w.r.t. } TL(G) \},
\]

\[
Q(L) := \{ K : K \supseteq L \text{ and } K \text{ is closed and observable w.r.t. } TL(G) \}.
\]

It is clear from the definition of \( act_K \) that if the language \( K \) is closed, then condition (ii) for \( act_K \) is satisfied automatically, a fact that we shall exploit throughout this section. Also, in this section all the languages involved \([A, E, TL(G), \text{etc.}]\) belong to \( \Sigma_{co}^* \), and controllability and observability are to be understood with respect to \( TL(G) \). Hence the relevant superset of events is \( \Sigma_{co} \) rather than \( \Sigma \).

It was shown in [8] that \( C(L) \) is a nonempty class of languages that is closed under arbitrary unions. Hence

\[
sup C(L)
\]

exists and is the supremal controllable sublanguage contained in the given language \( L \). Notice that \( sup C(L) \subseteq TL(G) \subseteq \Sigma_{co}^* \). Dually we have the following.

**Proposition 3.1.** \( Q(L) \) is a nonempty class of languages that is closed under arbitrary intersection.

**Proof.** Since \( TL(G) \in Q(L), Q(L) \) is nonempty. To show that \( Q(L) \) is closed under arbitrary intersection, let \( K_1, K_2 \in Q(L) \), i.e.

\[
K_1 \text{ is closed } \land K_1 \supseteq L \land ker P \leq act_{K_1},
\]

\[
K_2 \text{ is closed } \land K_2 \supseteq L \land ker P \leq act_{K_2}.
\]

Then

\[
K_1 \cap K_2 \text{ is closed } \land K_1 \cap K_2 \supseteq L.
\]

It remains to prove that \( ker P \leq act_{K_1 \cap K_2} \). Let \( s, s' \in \Sigma_{co}^* \) with \( Ps = Ps' \), and consider condition (i) for \( act_{K_1 \cap K_2} \):

(i) \( A_{K_1 \cap K_2}(s) \cap IA_{K_1 \cap K_2}(s') = \emptyset \).

If this condition fails, then for some \( \sigma \in \Sigma_{co} \),

\[
s\sigma \in K_1 \cap K_2 \land s' \in K_1 \cap K_2 \land s'\sigma \in TL(G) - K_1 \cap K_2.
\]
Also

\[ s'\sigma \in TL(G) - K_1 \cap K_2 \iff s'\sigma \in TL(G) \land (s'\sigma \notin K_1 \lor s'\sigma \notin K_2). \]

If \( s'\sigma \notin K_1 \), then

\[ s\sigma \in K_1 \land s' \in K_1 \land s'\sigma \in TL(G) - K_1, \]

namely,

\[ A_{K_1}(s) \cap IA_{K_1}(s') \neq \emptyset, \]

which contradicts the hypothesis \( \ker P \leq \text{act}_{K_1} \). If \( s'\sigma \notin K_2 \), similarly

\[ A_{K_2}(s) \cap IA_{K_2}(s') \neq \emptyset, \]

which contradicts the hypothesis \( \ker P \leq \text{act}_{K_2} \), so (i) is true after all. By similar reasoning

\[ A_{K_1 \cap K_2}(s') \cap IA_{K_1 \cap K_2}(s) = \emptyset. \]

From the above proposition, we conclude that \( \inf Q(L) \) exists and is the infimal observable sublanguage containing the given language \( L \). Of course \( \inf Q(L) \subseteq TL(G) \subseteq \Sigma_c^* \).

**Theorem 3.1.** SCOP is solvable if and only if

\[ \inf Q(A) \subseteq \sup C(E). \]

The proof of this theorem requires three lemmas.

**Lemma 3.1.** For a given language \( L \subseteq TL(G) \), the set of languages

\[ \underline{CC}(L) = \{ K : K \supseteq L \text{ and } K \text{ is closed and controllable w.r.t. } TL(G) \} \]

is nonempty and closed under arbitrary interactions.

**Proof.** We have \( TL(G) \subseteq \underline{CC}(L) \). If \( K_1, K_2 \in \underline{CC}(L) \), then \( K_1 \cap K_2 \supseteq L \), \( K_1 \cap K_2 \) is closed, and

\[ (K_1 \cap K_2) \Sigma_{uc} \cap TL(G) = K_1 \Sigma_{uc} \cap K_2 \Sigma_{uc} \cap TL(G) \subseteq K_1 \cap K_2. \]
By Lemma 3.1, \( \inf \text{CC}(L) \) exists and is the infimal closed controllable sublanguage of \( TL(G) \) containing the given language \( L \). It can even be written down explicitly, as follows.

**Lemma 3.2.**

\[
\inf \text{CC}(L) = TL(G) \cap L \Sigma^*_uc.
\]

**Proof.** Write \( M = TL(G) \cap L \Sigma^*_uc \). Clearly \( M \supseteq L \) and \( M \) is closed. To show that \( M \) is controllable w.r.t. \( TL(G) \), we have

\[
M \Sigma^*_uc \cap TL(G) = TL(G) \Sigma^*_uc \cap L \Sigma^*_uc \Sigma^*_uc \cap TL(G)
\]

\[
\subseteq L \Sigma^*_uc \cap TL(G)
\]

\[
= M.
\]

If now \( K \supseteq L \) and \( K \) is closed and controllable w.r.t. \( TL(G) \), then

\[
L \Sigma^*_uc \cap TL(G) \subseteq K \Sigma^*_uc \cap TL(G) \subseteq K,
\]

and by induction on \( r \) it follows that

\[
L \Sigma^*_uc \cap TL(G) = L \Sigma^*_uc \Sigma^*_uc \cap TL(G) \subseteq K,
\]

so that \( M \subseteq K \), i.e., \( M \) is infimal. \( \blacksquare \)

**Lemma 3.3.** If \( L \subseteq TL(G) \) is observable, then \( \inf \text{CC}(L) \) is also observable.

**Proof.** Write \( M = TL(G) \cap L \Sigma^*_uc \). By Lemma 3.2 we need only show that

\[
\ker P \leq \text{act}_L \implies \ker P \leq \text{act}_M.
\]

Assume \( \ker P \leq \text{act}_L \) and \( s, s' \in \Sigma^*_co \) with \( P_s = P_{s'} \). We check condition (i) for \( (s, s') \in \text{act}_M \). If \( A_M(s) \cap 1A_M(s') \neq \emptyset \), then for some \( \sigma \in \Sigma^*_co \),

\[
s\sigma \in M \quad \land \quad s' \in M \quad \land \quad s'\sigma \in TL(G) - M.
\]

By controllability of \( M \), \( s' \in M \land s'\sigma \in M \) implies \( \sigma \in \Sigma^*_uc \). Hence \( s\sigma \in L \) and \( s'\sigma \in L \). We claim that \( s' \in L \). Otherwise, if \( s' \notin L \), let \( w'\sigma' \) be the longest prefix of \( s' \) such that \( w' \in L \land w'\sigma' \notin L \). Then \( s' \in L \Sigma^*_uc \Rightarrow \sigma \in \Sigma^*_uc \Rightarrow \sigma' \in \Sigma^*_o \). So there exists a prefix \( w\sigma' \) of \( s \) such that \( Pw = P_{w'} \). Since \( L \) is
observable,

\[ Pw = Pw' \land w\sigma' \in TL(G) \land w'\sigma' \in TL(G) \land w\sigma' \in L \Rightarrow w'\sigma' \in L. \]

But this contradicts the fact that \( w'\sigma' \not\in L \), so that \( s' \in L \) after all. Therefore

\[ s\sigma \in L \land s' \in L \land s'\sigma \in TL(G) - L, \]

namely, \( A_L(s) \cap IA_L(s') \neq \emptyset \). This contradicts the assumption that \( \ker P \leq \text{act}_L \). Similarly, \( A_M(s') \cap IA_M(s) = \emptyset \).

**Proof of Theorem 3.1.** “Only if”: If SCOP is solvable, then there is a supervisor \( S \) such that

\[ A \subseteq TL(S/G) \subseteq E. \]

By Theorem 2.1, \( TL(S/G) \) is controllable and observable. Hence

\[ \inf Q(A) \subseteq TL(S/G) \subseteq \sup \mathcal{C}(E). \]

“If”: If \( \inf Q(A) \subseteq \sup \mathcal{C}(E) \) then

\[ \inf Q(A) \subseteq \inf \overline{CC}(\inf Q(A)) \subseteq \sup \mathcal{C}(E) \]

Now \( K := \inf \overline{CC}(\inf Q(A)) \) is controllable by definition, and is observable by Lemma 3.3. The result follows by Theorem 2.1.

The requirement in this section that the relevant languages be closed cannot be dropped. If they are not closed and for \( L \subseteq TL_m(G) \) we consider instead of \( \overline{CC}(L) \) the family

\[ \overline{NCC}(L) := \{ K : K \supseteq L \text{ and } K \text{ is controllable w.r.t. } TL_m(G) \}, \]

then \( \inf \overline{NCC}(L) \) need not exist. Suppose for instance that

\[ \Sigma = \Sigma_o = \{ \alpha, \beta, \gamma \}, \quad \Sigma_c = \{ \beta, \gamma \}, \quad \Sigma_{uc} = \{ \alpha \}, \]

\[ L_m(G) = 1 + \alpha \beta + \alpha \gamma, \quad L = 1. \]

Since \( \alpha \in L\Sigma_u \cap L(G) \) and \( \alpha \not\in L \), \( L \) is not controllable. Because \( L_m(G) \) is...
controllable, if \( \inf NCC(L) \) exists then \( L \subseteq \inf NCC(L) \subseteq L_m(G) \). Therefore the possible candidates for \( \inf NCC(L) \) are

\[
1 + \alpha \beta, \quad 1 + \alpha \gamma, \quad \text{or} \quad 1 + \alpha \beta + \alpha \gamma,
\]

but none of these is infimal.

The following example shows that if \( A \) and \( E \) are not closed, a solution to SCOP need not exist, even if \( A \) is observable and \( E \) is controllable. Let

\[
\Sigma = \{ \alpha, \beta \}, \quad \Sigma_o = \{ \alpha \}, \quad \Sigma_c = \{ \beta \},
\]

\[
L(G) = \overline{(1 + \beta) \alpha \beta}, \quad L_m(G) = L(G) - \{1\},
\]

as displayed below:

We take

\[
A = \beta, \quad E = \alpha + \beta + \beta \alpha \beta.
\]

Now

\[
\overline{A} = \{1, \beta\}, \quad \beta \in L_m(G),
\]

so \( A = \overline{A} \cap L_m(G) \), i.e., \( A \) is \( L_m(G) \)-closed. Also

\[
A_A(1) = \beta, \quad IA_A(1) = \alpha,
\]

\[
A_A(\beta) = \emptyset, \quad IA_A(\beta) = \alpha;
\]

hence \( A \) is observable. However, as

\[
\beta \in \overline{A}, \quad \alpha \in \Sigma_o, \quad \beta \alpha \in L(G) - \overline{A},
\]
$A$ is not controllable. Next, since $\bar{E} = L(G)$, $E$ is controllable; however, as

$$
\alpha, \beta \in \bar{E}, \quad P\alpha = \alpha = P(\beta \alpha),
$$

$$
A_E(\alpha) = \emptyset, \quad IA_E(\alpha) = \{ \beta \},
$$

$$
A_E(\beta \alpha) = \{ \beta \}, \quad IA_E(\beta \alpha) = \emptyset,
$$

it follows that $E$ is not observable.

Thus neither $A$ nor $E$ is a solution of SCOP. Finally, if

$$
A \subset K \subset E, \quad A \neq K \neq E,
$$

then

$$
K = K_1 := \beta + \alpha \quad \text{or} \quad K = K_2 := \beta + \beta \alpha \beta,
$$

but neither $K_1$ nor $K_2$ is controllable, and we conclude that SCOP is not solvable. Of course, if $E$ is replaced by $\bar{E} = L(G)$, then SCOP is trivially solvable by $\bar{E}$, and nontrivially by

$$
K = 1 + \alpha + \beta + \beta \alpha.
$$

In general, if $E$ is not closed, SCOP may fail to be solvable simply because $E$ has too few sublanguages.

4. SUPERVISORY CONTROL AND NORMALITY

Recall from [6] that a language $K$, $K \subseteq L(G)$, is normal [with respect to $L(G)$ and $P$] if

$$
K = L(G) \cap P^{-1}P(K).
$$

$K$ is normal just when it is the largest sublanguage of $L(G)$ whose projection is $P(K)$, and so is determined uniquely by its projection. Intuitively, therefore, one expects normality to be stronger than observability. In fact we have

**Proposition 4.1.** Let $K$ be a closed or an $L_m(G)$-closed sublanguage of $L(G)$. If $K$ is normal then $K$ is observable.

**Proof.** Suppose $K = L(G) \cap P^{-1}P(K)$. Let $s, s' \in \Sigma^*$ with $Ps = Ps'$. We show that $(s, s') \in \text{act}_K$ by checking the two conditions for $\text{act}_K$. 
(i) We claim that \( A_K(s) \cap IA_K(s') = \emptyset \). Otherwise

\[
(\exists \sigma \in \Sigma) \quad s\sigma \in K \wedge s' \in K \text{ and } s'\sigma \in L(G) - K.
\]

But \( s'\sigma \in L(G) \) and \( P(s'\sigma) = P(s\sigma) \in PK \) imply \( s'\sigma \in L(G) \cap P^{-1}P(K) = K \), a contradiction. Similarly, \( A_K(s') \cap IA_K(s) \neq \emptyset \).

(ii) By definition, if \( K \) is \( L_m(G) \)-closed, then

\[
K = \overline{K} \cap L_m(G)
\]

and condition (ii) holds automatically. Similarly, the result is obvious if \( K \) is closed.

A controllable language, even one that is closed, need not be normal; and \( E \) normal does not imply that \( \sup \mathcal{C}(E) \) is normal. For instance let

\[
\Sigma = \{ \alpha, \beta, \gamma, \lambda \}, \quad \Sigma_c = \{ \beta \}, \quad \Sigma_o = \{ \alpha, \beta, \gamma \}
\]

and

\[
L(G) = \overline{\beta \alpha + \lambda \beta \gamma}, \quad E = \overline{\beta + \lambda \beta \gamma}.
\]

Then \( E \) is closed and normal but controllable, while

\[
\sup \mathcal{C}(E) = \overline{\lambda \beta \gamma}
\]

is closed and controllable but not normal.

Similarly, a closed and observable language need not be normal. For instance let

\[
\Sigma = \{ \alpha, \beta, \lambda, \mu \}, \quad \Sigma_o = \{ \alpha, \beta \}
\]

(i.e. \( P\lambda = P\mu = 1 \)); and let

\[
L(G) = \overline{(1 + \lambda + \mu) \alpha + \beta},
\]

\[
J = 1 + \mu + \mu \alpha, \quad K = 1 + \alpha + \lambda + \lambda \alpha.
\]

Then \( K \) is closed and observable,

\[
PK = 1 + \alpha = PJ,
\]
but

\[ L(G) \cap P^{-1}PK = J + K, \]

namely, \( K \) is not normal.

The class of normal sublanguages of \( L(G) \) (for a fixed projection \( P \)) is algebraically better behaved than the observable sublanguages, as illustrated by the following proposition, whose proof may be left to the reader.

**Proposition 4.2.** Let \( B \subseteq L(G) \), and consider the classes of languages

\[ N(B) := \{ K \subseteq B \mid K \text{ is normal} \}, \]

\[ \overline{CN}(B) := \{ K \subseteq B \mid K \text{ is closed and normal} \}, \]

\[ \overline{RCN}(B) := \{ K \subseteq B \mid K \text{ is } L_m(G)\text{-closed and } \overline{K} \text{ is normal} \}. \]

Then \( N(B), \overline{CN}(B), \) and \( \overline{RCN}(B) \) are closed under arbitrary unions. In particular, if \( X(B) \) stands for any of these classes, then \( \varnothing \in X(B) \), namely \( X(B) \neq \varnothing \), and therefore \( \sup X(B) \) always exists in \( X(B) \).

It follows from Propositions 4.1, 4.2 and the definition of controllability that the class \( Z(B) \) of sublanguages of \( B \subseteq L(G) \) defined by

\[ Z(B) := C(B) \cap \overline{RCN}(B) \]

\[ = \{ K \subseteq B \mid K \text{ is normal, } K \text{ is } L_m(G) \text{ is closed and controllable} \} \]

is nonempty and closed under arbitrary unions. Thus \( Z(B) \) constitutes a natural and convenient class of sublanguages to be considered as candidates for the solution of problems of the type of SCOP. For instance, with \( A \subseteq E \subseteq L(G) \) given, the condition

\[ \sup Z(E) \supseteq A \]

is necessary and sufficient for the existence of a solution to SCOP in \( Z(E) \), and is thus sufficient for the existence of a solution to SCOP with no particular restrictions placed on \( A \) and \( E \).
5. **EXAMPLES**

**EXAMPLE 1.** Consider a rail network with two stations $(0,1)$:

![Diagram of rail network with two stations](image)

Let $G_i$ be the "process" that generates trains that depart from station $i$ for $j$ (event $d_i$) and arrive at $j$ from $i$ (event $a_i$). Let $\Sigma_i := \{ a_i, d_i \}$. In the absence of further restrictions, the language generated by $G_i$ would be

$$L(G_i) = \{ w \in \Sigma_i^* : |a_i|(w) \leq |d_i|(w) \}.$$

Suppose for simplicity that control is imposed by supervisor $S_i$, say, so that at most one train is running from $i$ to $j$ at any time. The sequence of arrivals and departures at station $0$ is now determined by the generator

$$G = \hat{G}_0 \| \hat{G}_1,$$

where $\hat{G}_i$ refers to $G_i$ controlled by the $S_i$, namely

$$L(\hat{G}_i) = (d_i/a_i)^*.$$

Consider a control for station 0 that enforces the capacity constraint that at most two trains be in the station at any time. Assume that this controller, $S_0$, say, will only observe events that occur at station 0, i.e.,

$$\Sigma_o = \{ d_0^1, a_1^0 \},$$

and can only control the departure of a train, i.e.,

$$\Sigma_c = \{ d_0^1, d_1^0 \}.$$

On the assumption that initially one train is present at station 0 and no trains are en route, the constraint is expressed by the legal language $E_o \subseteq \Sigma^*$
having the transition structure

with adjunction of a self-loop at each state, labeled $a_0^1$.

It can be shown that the language $L(G) \cap E_o$ is both controllable and observable. By Theorem 2.1, there exists a supervisor $S$ such that

$$L(S/G) = L(G) \cap E_o.$$ 

In fact, we can take $S = (S, \psi)$ where $S$ is the generator for $E_o$ as given above and $\psi$ is defined as

<table>
<thead>
<tr>
<th>$x$</th>
<th>$a = d_0^1$</th>
<th>$d_1^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

EXAMPLE 2. A production system consists of a workshop and an inventory. Raw materials are ordered from the dealer and delivered to the inventory and then sent to the workshop when they are required:
We consider the problem of inventory control. Assume that there are two types of raw materials, 1 and 2. The inventory has a capacity of \( n \) units. The order size is \( n_i \) units for type \( i \). Raw material is sent to the workshop in unit amounts. To model and control the system, we designate the following events:

- \( \alpha_i \): type-\( i \) material is ordered from the dealer;
- \( \beta_i \): type-\( i \) material is delivered to the inventory;
- \( \gamma_i \): type-\( i \) material is sent to the workshop.

Suppose that the controller (supervisor) is located at the inventory site whereas material is ordered from the workshop site, and assume that the controller observes only events that occur at its site, i.e.

\[
\Sigma_o = \{ \beta_i, \gamma_i : i = 1, 2 \}.
\]

Assume that the controller can disable \( \gamma_i \) as well as \( \alpha_i \), i.e.

\[
\Sigma_c = \{ \alpha_i, \gamma_i : i = 1, 2 \}.
\]

Assume finally that each order is filled with negligible time delay. Hence \( G \) is realized by the following:

In the transition diagrams, a self-loop labeled \( \{ \gamma_1, \gamma_2 \} \) is understood to be adjoined at each node.

Our task is to design a controller so that:

1. The (virtual) quantity \( k \) of raw material in the inventory never exceeds the inventory's capacity, i.e. \( 0 \leq k \leq n \).
2. Type-1 material has priority over type 2, i.e., whenever \( k > m \) for some fixed threshold \( m < n \), type-2 material is rejected, i.e. cannot be added to inventory.
The legal language $E$ is determined by these two rules. In case $n_1 = n_2 = 3$, $n = 8$, $m = 3$, we have $E = E' \cap L(G)$, where $E'$ is realized by the following generator:

![Diagram](image)

To specify $A$, we impose the following minimal requirements on the closed-loop system:

(3) The situation $k = n$ is possible, i.e., the inventory can be fully utilized. 
(4) Whenever $k \leq m$, type-2 material will not be rejected.

For the listed parameter values, $A$ is realized by the following generator:

![Diagram](image)

It can be checked that $A$ is observable, i.e. $\inf Q(A) = A$; while $E$ is not controllable, but $\sup C(E) = A$. That is,

$$\inf Q(A) = A = \sup C(E).$$

By Theorem 3.1, SCOP is solvable. A supervisor $S = (S, \psi)$ can be modeled on the inventory structure as follows.

$S$ has $n + 1$ states, $X = \{x_0, x_1, \ldots, x_n\}$, $x_0$ is the initial state, and every state is marked. The state transitions are the following:

(a) If the current state is $x_k$ and material of type $i$ is delivered to the inventory, then the next state is $x_{k+n_i}$. 
(b) If the current state is $x_k$ and material of type $i$ is sent to the workshop, then the state changes from $x_k$ to $x_{k-1}$.

The state feedback map $\psi(\sigma, x)$ is defined as follows:

(i) at state $x_0$ disable $\gamma_i$,
(ii) at state $x_k$, $k > m$, disable $\alpha_2$,
(iii) at state $x_n$, disable $\alpha_1$.

In case $n_1 = n_2 = 3$, $n = 8$, $m = 3$, the supervisor has the following structure:

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$x_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

6. CONCLUSION

A definition of observable language was framed in the setting of the discrete-event system theory of Ramadge and Wonham. Under certain restrictions, the existence problem for a supervisory controller (respecting both controllability and observability constraints) was solved abstractly. It was pointed out that the "normal" subclass of the observable languages has certain advantages as a source of control solutions. In future papers the computational aspects of the theory and more substantial application will be reported.

7. BIBLIOGRAPHICAL NOTE

The results reported herein were essentially completed in April 1986, and presented at that time in the second author's graduate seminar ELE1637S. We note that a similar definition of observable language was reported independently by Cieslak et al. [3]. The supervisory control and observation problem (SCOP) solved herein does not, however, appear in the Cieslak report.

REFERENCES


Received 28 February 1987