'90 韓國自動制御學術會議論文集 1990. 10.26~27

Designing Observation Functions in Supervisory Control

Hangju Cho

The Agency for Defence Development, Daejeon

Abstract

This paper discusses the problem of designing an observation function as coarsest as possible in supervisory control of discrete event dynamic systems. Some algebraic properties of two sets consisting of observation functions for which the given desirable behavior is realizable are investigated: these sets with a partial ordering turn out not to possess largest elements. Two natural methods of obtaining an observation fuction, which is frequently coarser than the identity mapping, are also presented. An example illustrates the use of these methods.

1. Introduction

A Discrete Event Dynamic System (DEDS) is an event- driven dynamic system with a discrete set of states which may take logical or symbolic values. The states are changing in response to events which occur asynchronously at discrete instants of time. While the issue of modeling, analysis and control of DEDS is attracting more and more researchers with various expertise, it seems unlikely to come up with an approach in the near future which resolves the issue in a unified manner. This is because a DEDS is usually a very complex man-made system and can pose many different problems each of which may require different disciplines to be efficiently solved. A reader who is interested in the modeling issue of DEDS is referred to [1].

Among many models for DEDS, an extended finite state machine framework [2] based on automata theory has been proved to be useful in the study of many qualititive control issues. A typical problem in this framework is the supervisor synthesis problem in which we are required to design a supervisor (or controller) so that the closed loop system behavior is "confined" in a desired behavior : the supervisor can observe through an observation function the events occuring in the system and may disable certain controllable events to occur in order to accomplish the required job. The closed loop system is depicted in Fig. 1.

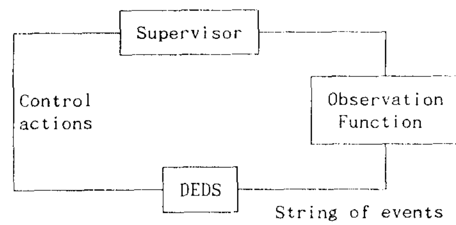


Fig. 1. The closed loop system of a supervisor synthesis problem

In the supervisor systhesis problem, the desired behavior is given in the form of a set consisting of all desirable strings of events which can be generated by the DEDS. The observation function is given a priori and fixed throughout the design procedure. Since there are uncontrollable events in the system such as machine-failure, it is in general impossible to realize the given desired behavior by means of control actions from the supervisor. Thus the supervisor synthesis problem becomes the problem of finding maxmal realizable portion of the desired behavior [2].

In this paper, we are concerned with the converse to the supervisor synthesis problem; that is, given a realizable behavior, design an observation function in an optimal fashion. Optimality here should represent minimal information that the observation stage is required to provide to the supervisor in order for the supervisor to be able to realize the given desired behavior. Thus in Section 3, a partial ordering for the set of observation functions will be introduced and the issue of existence of a largest element in the set will be examined. Section 4 presents a preliminary result for obtaining an observation function (a projection) which may not provide the supervisor with the information about the occurrence of certain events and still enable the supervisor to realize the given desired behavior. In section 2, the extended finite state machine framework will be reviewed.

2. Supervisory control of DEDS

In the extended finite state machine framework, the behavior of a DEDS is represented by the set of all strings of events that can be generated in the system. Such a set of strings of events is called a language, and is often represented by an automaton.

Formally, if Σ denotes a nonempty set of events and Σ^* the set of all finite strings of events in Σ including the empty string ε, any subset of Σ^* is called a language. A language L is said to be closed if it contains all prefixes of strings in L. In this paper, we only consider closed languages. For $s, t \in \Sigma^*$, the product st is the concatenation of two strings s and t. For $L_1, L_2 \subset \Sigma^*$, the product L_1L_2 is defined by

$$L_1L_2 = \{ w_1w_2 : w_1 \in L_1,\ w_2 \in L_2 \}.$$

A (deterministic) Finite Automaton (FA) is a 4-tuple $G = (Q, \Sigma, \delta, q_0)$, where Q is a finite set of states, q_0 initial state, and $\delta: \Sigma \times Q \rightarrow Q$ is a partial function. The state transition function δ is extended to $\Sigma^* \times Q$ in an obvious manner [2]. The language L(G) defined by

$$L(G) : \{ w \in \Sigma^* : \delta(w, q_0) \text{ is defined} \}$$

is called the language generated by G. A DEDS is represented by a FA G, and L(G) corresponds to the open loop behavior of the DEDS. The observation function in Fig.1. is a mapping $M: \Sigma \rightarrow \Delta \cup \{\varepsilon\}$, where Δ is a set of output symbols which the supervisor can observe. Again, M is extended to Σ^* in a natural manner [3].

Now we partition Σ into uncontrollable and controllable events: $\Sigma = \Sigma_u \cup \Sigma_c$. The controllable events can be disabled by control actions from the supervisor. The supervisor has no control over uncontrollable events, examples of which are machine-failure, packet-loss, and so forth. A subset γ of Σ satisfying $\Sigma_u \subset \gamma$ is called a control pattern, and is the mathematical model of a control action taken by the supervisor. The DEDS under the control pattern γ can execute only the events in γ. Thus the supervisor in Fig.1 is formally a mapping $f: M(L(G)) \rightarrow \Gamma$ [4], where Γ is the set of all control patterns. The closed loop behavior L_f is then described by

i) $\varepsilon \in L_f$

ii) $w\sigma \in L_f$ iff $w \in L_f$, $\sigma \in f(M(w))$ and $w\sigma \in L(G)$.

In practice, f is realized by a pair $\underline{S} = (S, \phi)$, where $S = (X, \Delta, \xi, x_0)$ is a FA and $\phi: X \rightarrow \Gamma$ is an output mapping of S. S is frequently a copy of a FA which generates L_f [2]. Clearly, $\underline{S}$ realizes f if for each $w \in L_f$, $\phi(\xi(M(w), x_0)) = f(M(w))$.

When a desired closed loop behavior is given in the form of a language $L \subset L(G)$, a typical supervisor synthesis problem can be posed as follows: given L, design f (or $\underline{S}$) such that $L_f = L$. Frequently, it is impossible to realize L exactly, i.e., to have f with $L_f = L$. In this case, we are forced to find a maximal realizable language L_{max} such that $L_{max} \subset L$ (For more detailed discussion on this point, see [5]). The necessary and sufficient conditions for the existance of f for which $L_f = L$ are the followings [2], [3]:

(a) L is $(\Sigma_u, L(G))$-invariant; i.e., $L\Sigma_u \cap L(G) \subset L$

(b) L is $(M, \Sigma_c, L(G))$-controllable; i.e.,

$s, t \in L$, $\sigma \in \Sigma_c$, $s\sigma \in L$, $t\sigma \in L(G)$, $M(s) = M(t) \Rightarrow t\sigma \in L$

Effective methods of computing $(\Sigma_u, L(G))$-invariance and $(M, \Sigma_c, L(G))$-controllablity for a given L can be found in [6] and [7], respectively. When a language satisfies the above conditions, a supervisor $\underline{S}$ can be constructed in a systematic way ([2], [3]) to realize the language.

3. Observation Function Design

3.1. Observation Function Design Problem

If we are given a closed language $L \subset L(G)$ and required to realize it, and if there is no restriction in choosing the observation function, the first step we take is frequently to extract a largest subset L' of L which is $(\Sigma_u, L(G))$-invariant : this is because only $(\Sigma_u, L(G))$-invariant languages can be realized by supervisory control described in Section 2 regardless of what mapping is employed for the observation stage, and because a larger language represents the more jobs which the DEDS can perform. Once the largest sublanguage L' is obtained, it can be readily realized by using the identity mapping as an observation function: recalling the necessary and sufficient conditions in Section 2 for the existence of a supervisor, we can easily see that the identity mapping always satisfies the condition (b). An interesting fact, however, is that there might be many different observation functions M for which the language L' is $(M, \Sigma_c, L(G))$-controllable. This fact has been already illustrated by an example in [5].

When many observation functions satisfying the condition (b) are available, we naturally seek for a "coarsest" observation function which represents the transmission of minimal amount of information. The minimal amount of information here will in fact correspond to a minimum number of sensors or communication lines, or some combination of both. Thus we consider the following observation function design problem (OFDP).

(OFDP) Given a closed $(\Sigma_u, L(G))$-invariant language $L \subset L(G)$, design an observation function M as coarsest as possible while it makes L $(M, \Sigma_c, L(G))$-controllable.

To make it clear what we mean by coarser observation functions, we introduce a partial ordering on a set of mappings in the next subsection.

3.2. Sets of mappings E and E_c

Recall that an observation function is a mapping $M: \Sigma \rightarrow \Delta \cup \{\varepsilon\}$, where Δ is a set of output symbols. In what follows, all mappings have the same domain Σ and take values from the fixed set $\Delta \cup \{\varepsilon\}$ unless otherwise specified. An element in $M^{-1}(\varepsilon)$ thus represents an event which cannot be seen by the supervisor. Also, if $M(\sigma)=M(\sigma')$, then the supervisor cannot distinguish between events σ and σ'. When we consider the set of all such mappings, denoted by EM, it is useful to introduce the following equivalence relation.

<u>Definition 1</u> : Two Mappings M_1 and M_2 are said to be <u>equivalent</u>, written $M_1 \equiv M_2$, if (i) $M^{-1}(\varepsilon)=M^{-1}(\varepsilon)=\Lambda$ and (ii) for all $\sigma_1, \sigma_2 \in \Sigma - \Lambda$, $M_1(\sigma_1)=M_1(\sigma_2)$ iff $M_2(\sigma_1)=M_2(\sigma_2)$.

The set $M^{-1}(\varepsilon)$ is sometimes said to be the <u>null set</u> of the mapping M. The condition (ii) above can be rephased as follows; i.e., $\Sigma - \Lambda$ has the same partition or the same equivalence relation induced by these mappings. Thus if M_1 and M_2 are equivalent, they convey exactly the same amount of information to the supervisor. Therefore we do not need to distinguish equivalent mappings between them. In this context, we consider the set E of all equivalence classes of EM under $\equiv$. For convenience in the later development, however, we consider E as the set of all mappings each of which is the representative of an equivalence class of EM. Thus if M_1 and M_2 are in E, then they are not equivalent. We include the identity mapping in E, which is the representative of all one-to-one mappings from Σ into Δ.

Now we are ready to introduce a partial ordering on the set E. Define a relation $\leq$ on E as follows: $M_1 \leq M_2$ if

(i) $\Lambda_1 \subset \Lambda_2$, and

(ii) for all $\sigma_1, \sigma_2 \in \Sigma - \Lambda_2$,
$M_1(\sigma_1)=M_1(\sigma_2) \Rightarrow M_2(\sigma_1)=M_2(\sigma_2)$,

where Λ_1 and Λ_2 are the null sets of M_1 and M_2, respectively. It can be easily verified that the relation $\leq$ is a partial ordering. Thus $(E, \leq)$ is a partially ordered set. In fact, we have

<u>Proposotion 1</u>: The partially ordered set $(E, \leq)$ is a complete lattice.

Note that the smallest element in E is the identity mapping and the largest element the mapping which has Σ as its null set. The physical meaning of the partial ordering $\leq$ is self-explanatory in its definition: for example, the larger the null set of a mapping is, the smaller number of sensors we need to install to observe the occurrence of events in the system. Also, when M_1 and M_2 have the same null set, the relationship $M_1 \leq M_2$ might imply that the number of transmission channels for M_2 required for feeding the corresponding information to the supervisor is not greater than that for M_1.

When we are given a closed language $L \subset L(G)$, not all members of E satisfy the $(M, \Sigma_c, L(G))$-controllability condition. We thus need to restrict ourselves to a smaller set $E_c \subset E$ of mappings M for which L is $(M, \Sigma_c, L(G))$-controllable. Then E_c with the partial odering $\leq$ inherited from $(E, \leq)$ becomes another partially ordered set. Now, the problem (OFDP) can be rephased as follows: find a mapping in E_c which is as large as possible. It is clear that the partially ordered set $(E_c, \leq)$ has a maximal element, since E_c has a finite number of elements in it. But, is there a largest element in $(E_c, \leq)$? The following example illustrates that $(E_c, \leq)$ is not a lattice and does not possess a largest element in general.

<u>Example 1</u> Let $\Sigma = \Sigma_c = \{a, b, c\}$, $L(G) = \overline{ab(a+c)^*}$, $L = \overline{abc^*}$ (these expressions for the languages L(G) and L are called <u>regular expressions</u> [8]. The upper bar denotes the prefix-closure [2], [5]). Fig.2. displays the FA's G and G_s, where $L(G_s) = L$.

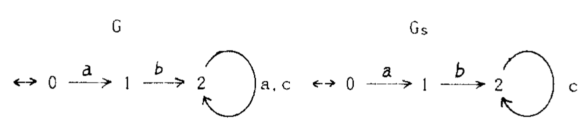


Fig.2. The FA's G and G_s ($L(G_s)$ = L)

Consider two mappings M_1 and M_2 defined as follows:

M_1: $M1 = M_1(a) = M_1(c) = \varepsilon$, $M_1(b) = \delta$.

M_2: $M_2 = M_2(b) = M_2(c) = \varepsilon$, $M_1(a) = \delta$.

Using the computation method in [7], we can easily verify that L is $(M_1, \Sigma_c, L(G))$-controllable and $(M_2, \Sigma_c, L(G))$-controllable. In other words, M_1 and M_2 belong to the set E_c (Clearly, M_1 and M_2 are distinct elements in E_c).

We now check if M_1 or M_2 has an upper bound in E_c. Glancing at the definition of M_1 and M_2, we immediately see that the only possible candidate for an upper bound of M_1 or M_2, excluding M_1 or M_2 itself, is the mapping M with the null set Σ, i.e., $M(a)=M(b)=M(c)=\varepsilon$. It turns out, however, that L is not $(M, \Sigma_c, L(G))$-controllable (try the strings $s=\varepsilon$, $t=ab$, $\sigma=a$ to see if the $(M, \Sigma_c, L(G))$-controllablity condition for this M is satisfied). Thus M_1 or M_2 does not have an upper bound other than themselves. Therefore there is no upper bound of the set $\{M_1, M_2\}$ either; hence the partially odered set $(E_c, \leq)$ for this problem is not a lattice and does not possess a largest element. Note that M_1 and M_2 are in fact maximal elements in $(E_c, \leq)$. (end of example)

The absence of a largest element forces us to consider the maximal elements of E_c as alternative solutions. However, it is not clear at this point what procedure would compute maximal elements of E_c. In the next section, we restrict our attention to a subset of E_c which consists of projections of Σ. Two natural methods of obtaining a coaser projection will be presented. Before leaving the section, we note that if we impose a stricter condition, namely $(M, L(G))$-recognizability [3], [7], instead of $(M, \Sigma_c, L(G))$-controllability for the membership of E_c, the algebraic property of the set does not improve; i.e., it is still not a lattice and does not have a largest element.

4. Methods of Computing Coarser Projections

4.1. Projections of Σ

Consider the mappings $P: \Sigma \rightarrow \Sigma \cup \{\varepsilon\}$ deifned in such a way that for some $\Lambda \subset \Sigma$, $P(\sigma)=\varepsilon$ if $\sigma \in \Lambda$ and $P(\sigma)=\sigma$ otherwise. Such mappings are called projections of Σ. Again, the set Λ will be called the null set of the projection P. In this section, we will only consider projections of Σ.

Let EP denote the set of all projections of Σ. The set EP can be put into one-to-one correspondence with the power set $P(\Sigma)$. In fact, two partially ordered sets $(EP, \leq)$ and $(P(\Sigma), \subset)$ are isomorphic: indeed, we have that $P_1 \leq P_2$ iff $\Lambda_1 \subset \Lambda_2$, where Λ_1 and Λ_2 are the null sets of P_1 and P_2, respectively. Therefore we can immediately see that $(EP, \leq)$ is a complete lattice. Now consider the set EP_c consisting of projections P for which L is $(P, \Sigma_c, L(G))$-controllable. The same example in Section 3 (the mappings M_1 and M_2 in the example are equivalent to the projections with null sets $\{a, c\}$ and $\{b, c\}$, respectively) shows that $(EP_c, \leq)$ has a maximal element and does not have a largest element in general. However, the set EP_c has a better algebraic property, which we show in the following.

Lemma 1 Let P and P′ be projections with $P' \leq P$. Then for all $s, t \in \Sigma^*$, $P'(s)=P'(t) \Rightarrow P(s)=P(t)$.

Proof: We use induction on the length of the string $P'(s)$. Let Λ and Λ' be the null sets of P and P′, respectively. If $P'(s)=\varepsilon$, then $P'(s)=P'(t)$ implies that $s, t \in \Lambda'$. Since $\Lambda' \subset \Lambda$, we have $s, t \in \Lambda$ so that $P(s)=P(t)=\varepsilon$. Suppose now that $P'(s)=\sigma_1\sigma_2 \ldots \sigma_n\sigma_{n+1}=P'(t)$. Let x and y be the largest prefixes of s and t, respectively, such that $P'(x)=\sigma_1\sigma_2 \ldots \sigma_n=P'(y)$. Then $s=x\sigma_{n+1}u$ and $t=y\sigma_{n+1}\nu$ for some $u, \nu \in \Sigma^*$ with $P'(u)=\varepsilon=P'(\nu)$. By the induction hypothesis, we have that $P(x)=P(y)$ and $P(u)=P(\nu)$. Therefore $P(s)=P(x)P(\sigma_{n+1})P(u)=P(y)P(\sigma_{n+1})P(\nu)=P(t)$. Q.E.D.

Proposition 2 If $P \in EP_c$ and if $P' \leq P$, then $P' \in EP_c$.

Proof: Suppose that $s, t \in L$, $\sigma \in \Sigma_c$, $s\sigma \in L$, $t\sigma \in L(G)$ and $P'(s)=P'(t)$. By Lemma 1, $P(s)=P(t)$. Since $P \in EP_c$, L is $(P, \Sigma_c, L(G))$-controllable. Thus $t\sigma \in L$, which implies that L is $(P', \Sigma_c, L(G))$-controllable. Therefore $P' \in EP_c$. Q.E.D.

The conclusion in Proposition 2 is intuitively rather obvious: when $P' \leq P$, the supervisor observes more events with P′. Thus the supervisor in this case should be able to realize the given language whenever this is possible with P. The above conclusion, however, fails when EP_c is replace by E_c: some difficulty in ensuring the conclusion arises when the null sets of mappings in E_c do not coincide.

4.2. Methods of Obtaining A Projection

In this subsection, we present two natural ways of obtaining a projection in EP_c which is, in many cases, coarser than the identity mapping.

Let G be a FA for a DEDS and let the language $L \subset L(G)$ be $(\Sigma_u, L(G))$-invariant. If the observation function is the identity mapping, there exists a supervisor that realizes L. We denote this supervisor by $\underline{S} = (S, \phi)$, where $S=(X, \Sigma, \xi, x_0)$. Now let Σ_{tr} be the set of events that cause state transitions into states different from previously occupied states; i.e.,

$$\Sigma_{tr} = \{\sigma : \xi(\sigma, x)=y \text{ for some } x, y \in X \text{ with } x \neq y\}.$$

Thus the supervisor stays at the same state and maintains the same control pattern when it observes events in $\Sigma - \Sigma_{tr}$. Therefore it is reasonable to believe that the supervisor can do the job without observing the events in $\Sigma - \Sigma_{tr}$. Indeed, we have

Proposition 3 Let P be a projection with the null set $\Lambda = \Sigma - \Sigma_{tr}$. Then L is $(P, \Sigma_c, L(G))$-controllable.

Proof Let $s, t \in L$ and $P(s)=P(t)$. Since $s, t \in L$, $\xi(s, x_0)$ and $\xi(t, x_0)$ are defined. Moreover, $P(s)=P(t)$ implies that $\xi(s, x_0)=\xi(t, x_0)$: this can be easily seen by recalling the definitions of P and Σ_{tr}. Now suppose that $\sigma \in \Sigma$, $s\sigma \in L$ and $t\sigma \in L(G)$. Again, recalling the definition of the closed loop language and the supervisor realization, we see that $s\sigma \in L$ implies $\sigma \in \phi(\xi(s, x_0))$. Thus $\sigma \in \phi(\xi(t, x_0))$ (since $\xi(s, x_0)=\xi(t, x_0)$), which in turn implies that $t\sigma \in L$. Hence we have shown that L is $(P, \Sigma_c, L(G))$-controllable. Q.E.D.

The above method of obtaining a projection is easy and straightforward. It is, however, heavily dependent upon the structure of the supervisor which we started with. Since there are many supervisors $\underline{S}$ which realize the same mapping f, it become very important in this method what kind of supervisor $\underline{S}$ is to be used when computing Σ_{tr}: for example, if the supervisor is more "lumped", it is more likely to come up with a coarser projection. In the following, we present a refinement of the above method which does not show such degree of structural dependency.

Definition 2 Two states x and y in X are said to be control-equivalent, written $x \sim y$, if for all $s \in \Sigma^*$, $\phi(\xi(s, x)) = \phi(\xi(s, y))$ whenever both $\xi(s, x)$ and $\xi(s, y)$ are defined.

Remark It is easy to see that $\sim$ is an equivalence relation on X. Also, if $x \sim y$, then $\xi(s, x) \sim \xi(s, y)$ whenever they are defined.

Now let

$$\Sigma_{ce} = \{\sigma: \xi(\sigma, x)=y \text{ for some } x, y \in X \text{ with } x \sim y\}.$$

Define a projection P_{ce} by letting its null set to be $\Sigma - \Sigma_{ce}$.

Lemma 2 If $P_{ce}(s) = P_{ce}(t)$ for $s, t \in L$, then $\xi(s, x_0) \sim \xi(t, x_0)$.

Proof We use induction on the length of $P_{ce}(s)$. If $P_{ce}(s)=\varepsilon=P_{ce}(t)$, then $x_0 \sim \xi(s, x_0)$ and $x_0 \sim \xi(t, x_0)$. By transitivity of equivalence relations, $\xi(s, x_0) \sim \xi(t, x_0)$. Now suppose that $P_{ce}(s)=\sigma_1\sigma_2 \ldots \sigma_n\sigma_{n+1} = P_{ce}(t)$. Let s_1 and t_1 be the largest prefixes of s and t, respectively, such that $P_{ce}(s_1)=\sigma_1\sigma_2 \ldots \sigma_n = P_{ce}(t_1)$. Then $s=s_1\sigma_{n+1}u$ and $t=t_1\sigma_{n+1}v$ where $P_{ce}(u)=\varepsilon=P_{ce}(v)$. Let $x=\xi(s_1, x_0)$ and $y=\xi(t_1, x_0)$. By the induction hypothesis, $x \sim y$. Thus $\xi(\sigma_{n+1}, x) \sim \xi(\sigma_{n+1}, y)$ (see the Remark after Definition 2). Moreover, $P_{ce}(u) = \varepsilon$ implies that $\xi(u, \xi(\sigma_{n+1}, x))$ $(=\xi(\sigma_{n+1}u, x)) \sim \xi(\sigma_{n+1}, x)$. Similarly, $\xi(\sigma_{n+1}v, y) \sim \xi(\sigma_{n+1}, y)$. Hence $\xi(\sigma_{n+1}u, x) \sim \xi(\sigma_{n+1}v, y)$. But $\xi(\sigma_{n+1}u, x)=\xi(\sigma_{n+1}u, \xi(s_1, x_0))=\xi(s_1\sigma_{n+1}u, x_0) = \xi(s, x_0)$. Similarly, $\xi(\sigma_{n+1}v, y)=\xi(t, x_0)$. Thus we have established that $\xi(s, x_0) \sim \xi(t, x_0)$. Q.E.D.

Proposition 4 L is $(P_{ce}, \Sigma_c, L(G))$-controllable.

Proof Let $s, t \in L$ and $P_{ce}(s)=P_{ce}(t)$. Then, by Lemma 2, $\phi(\xi(s, x_0))=\phi(\xi(t, x_0))$. The rest of the proof is identical to that of Proposition 3. Q.E.D.

In the following, we demonstrates the use of the above results through an example.

Example 2 Consider two FA's G and G_s in Fig.3. Let

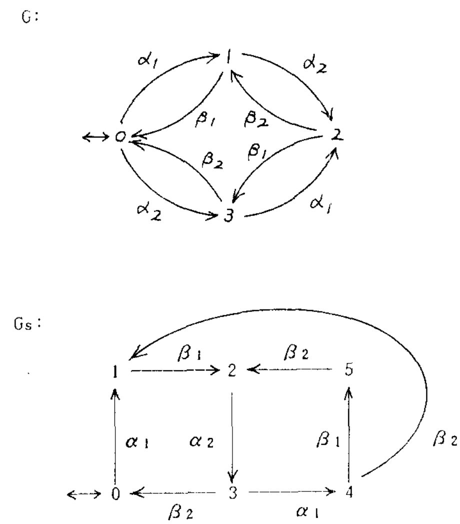
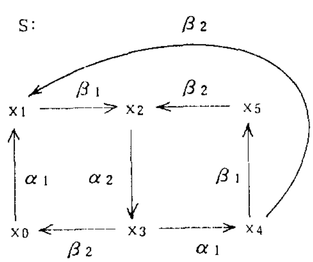


Fig.3. The FA's G and Gs

$L(G_s)= L$ and let $\Sigma_c = \{\alpha_1, \alpha_2\}$. Then $L \subset L(G)$ and L is $(\Sigma_u, L(G))$-invariant. Fig.4. displays a supervisor $\underline{S} = (S, \phi)$ which realizes L when the identity mapping is used as the observation function.



ϕ:

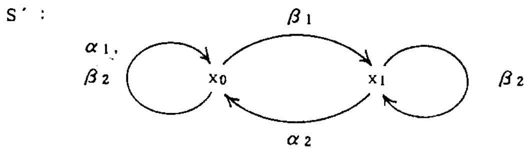
$$\phi(x_i) = \begin{cases} \gamma_1, & i=0,1,3,4 \\ \gamma_2, & i=2,5. \end{cases}$$

$\gamma_1=\{\alpha_1, \beta_1, \beta_2\}$

$\gamma_2=\{\alpha_2, \beta_1, \beta_2\}$

Fig.4. Supervisor $\underline{S} = (S, \phi)$

Now it is easy to verify that the subsets $\{x_2, x_5\}$ and $\{x_0, x_1, x_3, x_4\}$ of X are two equivalence classes of X under $\sim$. Therefore $\Sigma_{ce} = \{\alpha_2, \beta_1\}$. Thus we obtain a projection P_{ce} which is coarser than the identity mapping. As indicated in [5], there is another supervisor $\underline{S}'=(S', \phi')$ where S' is structurally much simpler than S. Fig.5. displays $\underline{S}'$.



$\phi'(x_0)=\{\alpha_1, \beta_1, \beta_2\}$, $\phi'(x_1)=\{\alpha_2, \beta_1, \beta_2\}$

Fig.5. Supervisor $\underline{S}'$

Note that for this supervisor $\underline{S}'$, $\Sigma_{tr}=\{\alpha_2, \beta_1\}$. Thus, using the first method (Proposition 3), we get the same projection as before. It should be, however, noted that the first method applied to $\underline{S}$ results in the identity mapping; on the other hand, the second method applied to $\underline{S}'$ yields the same P_{ce} as before. In this respect, we also note that it is quite possible to devise a procedure which "reduces" a supervisor by using the concept of control-equivalence; for example, this procedure would in fact reduces the supervisor $\underline{S}$ to $\underline{S}'$.

(end of example)

As suggested in the above example, the problem of obtaining a coarser projection is closely related to that of supervisor reduction [9]. In this context, it is worth noting that we could define the output mapping ϕ differently with which the supervisor $\underline{S}$ still be able to realize L: this is because at some state x, the inclusion/exclusion of certain events into/from $\phi(x)$ does not make any difference in the closed loop language of the system. However, different output mappings yield different partitions of X (induced by $\sim$), and therefore could result in different projections P_{ce}. The extra freedom in choosing ϕ described above has played a key role in the development of a supervisor reduction method in [9]. It is thought at this stage that the concept of "covers" in [9] could enable us to develop yet another procedure for obtaining a coaser projection; the projection obtained in this way is expected to be coarser than that obtained by the second method in this paper, since the concept of cover is a generalization of the notion of control-equivalent classes.

5. Concluding Remarks

In this paper, the problem of designing an observation function has been discussed. A partial ordering $\leq$ on the set of observation functions has been introduced in order to compare the amounts of information conveyed by various observation functions. The partially ordered set $(E_c, \leq)$ or $(EP_c, \leq)$ consisting of observation functions for which the given language can be realized turns out not to possess a largest element. As a first step towards getting a maximal observation fuction, two natural methods of obtaining a projection have been presented in Section 4. Maximality of solutions in these methods has not been investigated.

There are some remaining problems: as discussed in Section 4, it would be interesting to see how the problem of obtaining a (maximal) projection is related to that of achieving a (optimal) supervisor reduction studied in [9]. It will be also worth trying to characterize a maximal projection by using an "efficient" supervisor in [2]. Some form of extension of the results in Section 4 to the case of general mappings is again remaining to be seen.

References

[1] Y.C. Ho, Ed., "Special Issue on Dynamics of Discrete Event Systems", Proc.IEEE, V77, No.1, 1989

[2] P.J. Ramadge, W.M. Wonham, "Supervisory Control of a Class of Discrete Event Process", SIAM J. Contr. Optimz., V25, pp 206-230, 1987

[3] R.Cieslak, C.Desclaux, A. Fawaz, and P. Varaiya, "Supervisory Control of Discrete Event Processes with Partial Observations", IEEE Trans. AC-33, pp 249-260, 1988

[4] P.J. Ramadge and W.M. Wonham, "The Control of Discrete Event Systems", Proc. IEEE, V77, No.1, pp 81-98, 1989

[5] H. Cho, "Supervisory Control of Discrete Event Dynamical Systems", 1989

[6] W.M. Wonham, P.J. Ramadge, "On the Supremal Controllable Sublanguage of a Given Language", SIAM J. Contr. Optimiz. V25,No.3, pp637-659, 1987

[7] H. Cho, S.I. Marcus, "Supremal and Maximal Sublanguages Arising in Supervisor Synthesis Problems with Partial Observations", Math. Systems Theory, V22, pp 177-211, 1989

[8] J.E. Hopcroft, J.D. Ullman, "Introduction to Theory, Language, and Computation", Addison-Wesley, 1979

[9] A.F.Vaz, W.M. Wonham, "On Supervisor Reduction in Discrete-Event Systems", Int. J. Control, V44, pp 475-491, 1986