Decision Fusion and Supervisor Synthesis in Decentralized Discrete-Event Systems

Joseph H. Prosser[†], Moshe Kam[†], and Harry G. Kwatny[‡]

[†]Department of Electrical and Computer Engineering
[‡]Department of Mechanical Engineering and Mechanics
Drexel University, 3141 Chestnut Street
Philadelphia, PA 19104

Abstract

We study a problem in decentralized supervisory control coined the *Global Problem* (GP) by Rudie and Wonham (1992). The objective is to find local supervisors that generate languages that lie in a specified range of closed languages. We present results for a version of this problem called the *Special Global Problem*, (SGP) where the lower end of the range is empty and the higher end of the range is a closed and regular 'legal' language. To this end, on-line local supervisors are synthesized that generate a language that contains the supremal closed controllable and strongly decomposable (*supCCSD*) sublanguage of the legal language. We accomplish this task through a new, more general decentralized supervisory control architecture involving *decision fusion*. We formulate two problems that extend GP and SGP. These are *GP with Fusion* (GPF) and *SGP with Fusion* (SGPF). Our main result is a set of local supervisors that solve SGPF for any *k-out-of-n* type event fusion rule. This result is of particular interest when more than one supervisor has control over the same event. Furthermore, the local supervisor designs are *decoupled*, thus the (on-line) computation of local supervisors can be performed in a distributed manner.

1. Introduction

Under centralized supervision of discrete-event systems, all available information about event occurrences in a discrete-event system is routed to a single supervisor that has the ability to disable any subset of the system's controllable events at any time. In a *decentralized* supervisory control architecture, two or more supervisors simultaneously affect a DES plant.

An excellent overview of the evolution of decentralized supervisory control theory appears in [9]. The theory was originally suggested for rendering large complex centralized supervisory control problems more computationally manageable. In [5] and [11] conditions are given under which a centralized supervisor can be computed in a *modular* fashion. The control task is split into several subtasks that are solved independently, then combined to form a single solution to the original problem. This led to the use of *modularity under partial observation* [2], and to a general decentralized supervision architecture, first presented in [1].

In [1], each local supervisor possesses its own mask function (the supervisor's *local mask function*) and a set of events that it may disable (the supervisor's *local controllable event set*). A local mask function provides event observations to a single local supervisor, and the local controllable event sets are not necessarily disjoint; there may be events that more than one local supervisor can disable.

We introduce into decentralized supervisory control theory the use of *event fusion rules*. These rules determine how events get globally disabled as a function of the local disabling decisions when the decisions are conflicting. To date, only two approaches have been analyzed for resolving these conflicts. One used an *OR rule*, where an event is disabled whenever *any* local supervisor wishes it to be disabled. The other used an *AND rule*, where an event is disabled globally only if *every* local supervisor wishes it to be disabled (that is, every local supervisor that has the event in its local controllable event set.) To date, virtually every study in decentralized supervisory control has assumed that all local disabling decisions are reconciled according to the OR rule.

We find and synthesize local supervisors that generate sublanguages of a specified closed and regular legal language, under a fixed fusion rule. These are solutions to the SGPF. We allow each controllable event to use a fusion rule that is a *k-out-of-n* rule, where n is the number of supervisors that have the event in its controllable event set; an event is disabled if at least k of those supervisors wish to disable it. Results obtained under the OR rule assumption now appear as a special case, when $k = 1$. The generated closed-loop language contains the *supremal controllable and strongly decomposable* sublanguage [9] of the legal language.

This study was supported by the NSF grant no. ECS-9216588 and the Electric Power Research Institute contract no. RP-8030-01. A companion paper appears in the Proceedings of the 1997 Conference on Information Sciences and Systems, Johns Hopkins University, Baltimore, MD.

0-7803-3835-9/97/$10.00 (c) 1997 AACC

2. Preliminaries

We review the necessary background in the Ramadge/Wonham framework for centralized supervisory control of discrete-event systems. Our notation and definitions are based on the references [3], [6], [7], and [10].

The system to be controlled is modeled by a deterministic finite automaton (DFA) or *generator* $G = (Q, \Sigma, \delta, q_0)$, where Q is a finite set of states containing the initial state q_0, Σ is a nonempty finite set of events such that the *null event* $\epsilon \notin \Sigma$, and δ is a state transition function (a partial function) that maps a subset of $\Sigma \times Q$ into Q. Events that occur in sequence are *concatenated* to form *strings*. The set $\Sigma^* = \{\epsilon\} \cup \Sigma \cup \Sigma\Sigma \cup \ldots$ is the set of all strings made from concatenating any number of events in Σ. The null event ϵ satisfies for all $s \in \Sigma^*$, $s\epsilon = \epsilon s = s$.

The plant's transition function is extended in its domain over a subset of $\Sigma^* \times Q$ in the natural way. A *language* is any subset of Σ^*. A language L is said to be *prefix-closed* or just *closed* if every prefix of every string in L is also a member of L. Corresponding to the DFA G we define a language $L(G)$ (called the *plant language*) defined to be the set

$$L(G) := \{s \mid s \in \Sigma^* \wedge \delta(s, q_0) \text{ is defined}\}. \tag{1}$$

The language $L(G)$ is closed, and we say that the plant G *generates* $L(G)$. If $L \subseteq L(G)$ we say that L is a *sublanguage of* $L(G)$ or just that L is a sublanguage.

We partition Σ into two subsets: Σ_c, the set of controllable events; and Σ_u, the set of uncontrollable events. The plant is controlled by a *supervisor* that can prevent controllable events from occurring based on its observation of (past) event occurrences. A supervisor is a function $\gamma : L(G) \to 2^{\Sigma_c}$ defined for all $s \in L(G)$, where $\gamma(s)$ is the set of events that are disabled after the occurrence of the string s. Under the event-disabling action of the supervisor γ, the plant G generates a *closed-loop language* that is (by construction) a closed sublanguage of $L(G)$. The closed-loop language, denoted $L(G, \gamma)$, is defined by setting $\epsilon \in L(G, \gamma)$ and

$$s\sigma \in L(G, \gamma) \iff s \in L(G, \gamma) \wedge s\sigma \in L(G) \wedge \sigma \notin \gamma(s). \tag{2}$$

If $L(G, \gamma) = L$ we say that the supervisor γ *generates* L. We extend γ in its domain over strings.

For all $J \subseteq L(G)$, the extended function γ, denoted $\bar{\gamma}$, is given by

$$\bar{\gamma}(J) := \bigcup_{s \in J} \gamma(s). \tag{3}$$

Assume that $L \subseteq L(G)$ is prefix-closed. L is *controllable* (with respect to Σ_u) if

$$(\forall s \in L, \sigma \in \Sigma_u)\, [s\sigma \in L(G) \implies s\sigma \in L]. \tag{4}$$

If L is a closed sublanguage of $L(G)$ then a supervisor exists such that $L(G, \gamma) = L$ if and only if L is controllable [6]. Every closed language $L \subseteq L(G)$ contains a supremal closed controllable sublanguage [10]. The supremal closed controllable sublanguage of L is denoted $L^{\uparrow}$. An effective algorithm for the computation of $L^{\uparrow}$ (when L is regular) is given in [10]. When $L \subseteq L(G)$ is closed and regular, so is $L^{\uparrow}$.

A *mask* [1] is a function $M : \Sigma \to \Lambda \cup \{\epsilon\}$, where $\epsilon \notin \Lambda$, M is defined for all $\sigma \in \Sigma$, and Λ is another event set. Events in Λ are called *observed events*. The mask M is extended in its domain over strings through concatenation and assigning $M(\epsilon) := \epsilon$. The mask function is extended in its domain over sets of events and sets of strings in the natural way.

The *inverse mask function* is a map $M^{-1} : 2^{\Lambda^*} \to 2^{\Sigma^*}$ defined for all $\Delta \subseteq \Lambda^*$ by

$$M^{-1}(\Delta) := \{s \in \Sigma^* \mid M(s) \in \Delta\}. \tag{5}$$

For $l \in \Lambda^*$, the set of strings in Σ^* whose mask value is equal to l is given by $M^{-1}(\{l\})$. The set Σ_λ is the set of events $\sigma \in \Sigma$ that satisfy $M(\sigma) = \lambda$ (for $\lambda \in \Lambda \cup \{\epsilon\}$). The *unobservable events* are those in Σ_ϵ. All other events (i.e., ones in $\Sigma - \Sigma_\epsilon$) are called *observable events*. For simplicity of notation we omit parentheses and braces and write $M\sigma$ for $M(\sigma)$, $ML(G)$ for $M(L(G))$, $M^{-1}l$ for $M^{-1}(\{l\})$, etc.; it is always clear from the context what is meant.

A *partial information supervisor* is a function $\gamma_M : ML(G) \to 2^{\Sigma_c}$ defined for all $l \in ML(G)$. The closed-loop language of a partial information supervisor is computed by assigning

$$(\forall s \in L(G))\ \gamma(s) := \gamma_M(Ms), \tag{6}$$

and then computing $L(G, \gamma)$ as in (2). We shall always use the subscript M for partial information supervisors (that use the mask M) so we are permitted to write $L(G, \gamma_M)$.

3. Decentralized Supervisory Control

We study a supervisory control architecture where a number of independent partial information supervisors simultaneously affect the operation of the same DES plant. Let $\{\gamma_{M_i}\}_{i=1}^n$ be a set of n partial information supervisors $\gamma_{M_i} : M_i L(G) \to 2^{\Sigma_{ci}}$, where $M_i : \Sigma \to \Lambda_i \cup \{\epsilon\}$ is a mask function for supervisor i, and $\Sigma_{ci} \subseteq \Sigma$ is a set of events that supervisor i may decide to disable. The sets Σ_{ci}, $i = 1, \ldots, n$ are not necessarily disjoint. The partial information supervisors $\{\gamma_{M_i}\}_{i=1}^n$ are called *local supervisors*. Similarly, the mask function used by local supervisor i (i.e., M_i) is called a *local mask function*, and the set of events that supervisor i plays a role in disabling (i.e., Σ_{ci}) are called *locally controllable events*. The set of *locally uncontrollable events* is given by $\Sigma_{ui} = \Sigma - \Sigma_{ci}$. For every i, and for each $s \in L(G)$, the value of $\gamma_{M_i}(M_i s)$ is a subset of Σ_{ci} and is called a *local disabling decision*.

0-7803-3835-9/97/$10.00 (c) 1997 AACC

We define the controllable and uncontrollable event sets Σ_c and Σ_u as $\Sigma_c = \Sigma_{c1} \cup \ldots \cup \Sigma_{cn}$ and $\Sigma_u = \Sigma - \Sigma_c$.

As the same controllable event (say σ) may be locally controllable for more than one supervisor, (e.g., $\sigma \in \Sigma_{ci} \cap \Sigma_{cj}$, $i \neq j$) one local supervisor may decide to disable σ while another may decide *not* to disable σ. For this reason, a rule must be established that reconciles conflicting decisions among local supervisors.

For each event $\sigma \in \Sigma_c$, let $I_\sigma = \{i \mid \sigma \in \Sigma_{ci}\}$. The set I_σ is the set of indicies of local supervisors that have σ in their local controllable event sets. Let $n_\sigma = |I_\sigma|$, the number of local supervisors that have σ as a locally controllable event.

An *event fusion rule* is a map $\mathcal{F}_\sigma : 2^{I_\sigma} \to \{\emptyset, \{\sigma\}\}$ that satisfies for all $P_1, P_2 \subseteq I_\sigma$ the following monotonicity property:

$$P_1 \subseteq P_2 \implies \mathcal{F}_\sigma(P_1) \subseteq \mathcal{F}_\sigma(P_2). \tag{7}$$

Suppose the event σ is disabled by some set of supervisors whose indicies are in the set $P \subseteq I_\sigma$. If $\sigma \in \mathcal{F}_\sigma(P)$ (i.e., $\mathcal{F}_\sigma(P) = \{\sigma\}$) then the event σ is globally disabled. If $\sigma \notin \mathcal{F}_\sigma(P)$ (i.e., $\mathcal{F}_\sigma(P) = \emptyset$) then the event σ is globally enabled.

For the event σ, the event fusion rule $\mathcal{F}_\sigma$ determines precisely how conflicting disabling decisions among the local supervisors (concerning σ) get reconciled. The monotonicity requirement ensures that if a group of supervisors' disable decisions is sufficient to globally disable an event, then an additional local decision of another supervisor to disable the event will not reverse the global decision.

We consider hybrid *k-out-of-n* type event fusion rules. For every $\sigma \in \Sigma_c$, let $1 \leq k_\sigma \leq n_\sigma$. For every $\sigma \in \Sigma_c$ and for all $P \subseteq I_\sigma$, let

$$\mathcal{F}_\sigma^{k_\sigma}(P) = \begin{cases} \{\sigma\} & \text{if } |P| \geq k_\sigma \\ \emptyset & \text{otherwise.} \end{cases} \tag{8}$$

In $\mathcal{F}_\sigma^{k_\sigma}(P)$, the set P would contain the indices of local supervisors that wish σ to be disabled, and if at least k_σ local supervisors wish to disable σ, then $|P| \geq k_\sigma$ and the event σ would get globally disabled.

The event fusion rule $\mathcal{F}_\sigma^1$ is the *OR rule*, where it only takes one local decision to disable to globally disable σ. If $k_\sigma = n_\sigma$, then $\mathcal{F}_\sigma^{k_\sigma}$ is the *AND rule*, where it takes n_σ disable-decisions (the number of supervisors that have σ as a locally controllable event) to globally disable σ. If $k_\sigma = n_\sigma/2$ then the event fusion rule $\mathcal{F}_\sigma^{k_\sigma}$ is a *majority rule*.

All the event fusion rules get combined into a single rule that involves all controllable events. Given $\{k_\sigma\}_{\sigma \in \Sigma_c}$, for all $R_1 \subseteq \Sigma_{c1}, \ldots, R_n \subseteq \Sigma_{cn}$, define the *global fusion rule* corresponding to the event fusion rules $\{\mathcal{F}_\sigma^{k_\sigma}\}_{\sigma \in \Sigma_c}$ by

$$\mathcal{F}(R_1, \ldots, R_n) := \bigcup_{\sigma \in \Sigma_c} \mathcal{F}_\sigma^{k_\sigma}(\{i \mid \sigma \in R_i,\ i = 1, \ldots, n\}). \tag{9}$$

Here, each R_i represents a possible local disabling decision of local supervisor i. The set $\{i \mid \sigma \in R_i,\ i = 1, \ldots, n\}$ is a subset of I_σ and it consists of the indicies of the local supervisors for which σ is an element of the respective local disabling decision. For every string $s \in L(G)$, the global fusion rule is applied to reconcile a set of local disabling decisions $\gamma_{M_1}(M_1 s), \ldots, \gamma_{M_n}(M_n s)$ by computing the set $\mathcal{F}(\gamma_{M_1}(M_1 s), \ldots, \gamma_{M_n}(M_n s)) \subseteq \Sigma_c$.

Of special interest are the global OR rule, denoted $\mathcal{O}$, for which $k_\sigma = 1$ for all $\sigma \in \Sigma_c$, and the global AND rule, denoted $\mathcal{A}$, for which $k_\sigma = n_\sigma$ for all $\sigma \in \Sigma_c$. It can be shown that for all $R_1 \subseteq \Sigma_{c1}, \ldots, R_n \subseteq \Sigma_{cn}$,

$$\mathcal{O}(R_1, \ldots, R_n) = R_1 \cup \ldots \cup R_n. \tag{10}$$

A *decentralized supervisor*, denoted $(\mathcal{F}, \{\gamma_{M_i}\}_{i=1}^n)$ (or for simplicity just $\{\gamma_M\}_\mathcal{F}$) is pair of a global fusion rule and a set of n partial information supervisors. The closed-loop language generated by a decentralized supervisor is computed by submitting, for each $s \in L(G)$, the local disabling decisions $\{\gamma_{M_i}(M_i s)\}_{i=1}^n$ to the fusion rule $\mathcal{F}$ to get the global disabled-event set (as a function of s). The closed-loop language generated by the decentralized supervisor $(\mathcal{F}, \{\gamma_{M_i}\}_{i=1}^n)$, denoted $L(G, \{\gamma_M\}_\mathcal{F})$, is defined to be the closed-loop language $L(G, \gamma_\mathcal{F})$, where (for all $s \in L(G)$)

$$\gamma_\mathcal{F}(s) := \mathcal{F}(\gamma_{M_1}(M_1 s), \ldots, \gamma_{M_n}(M_n s)). \tag{11}$$

The functions $M_1, \ldots, M_n$ are all mask functions by our centralized definition, so they are extended in their domains and their inverses are defined accordingly. We denote by Σ_λ^i (where $\lambda \in \Lambda_i \cup \{\epsilon\}$) the set of events $\sigma \in \Sigma$ that satisfy $M_i \sigma = \lambda$.

Assume that $L \subseteq L(G)$ is prefix-closed. L is *co-observable* (with respect to $M_1, M_2, \ldots, M_n$) if for all $s, s_1, \ldots, s_n \in L$ and $\sigma \in \Sigma$,

$$(\forall i)[(s_i \sigma \in L \wedge M_i s_i = M_i s) \vee \sigma \notin \Sigma_{ci}] \wedge s\sigma \in L(G) \implies s\sigma \in L \tag{12}$$

The property of co-observability is a generalization of observability to when many supervisors operate simultaneously and the global OR rule is used.

We present a theorem on the existence of a decentralized supervisor under the OR rule.

Theorem 1 *Assume that $L \subseteq L(G)$ is prefix-closed. For all i, let $M_i : \Sigma \to \Lambda_i \cup \{\epsilon\}$ and $\Sigma_{ci} \subseteq \Sigma$.*

There exists a decentralized supervisor $\{\gamma_M\}_\mathcal{O}$ such that $L(G, \{\gamma_M\}_\mathcal{O}) = L$ if and only if L is both controllable with respect to Σ_u and co-observable with respect to $M_1, \ldots, M_n$.

For a proof, see [1], [8], or [4].

As may be expected (because of its relationship to observability) the property of co-observability is not closed

0-7803-3835-9/97/$10.00 (c) 1997 AACC

under union, so there may not exist a supremal closed controllable and co-observable sublanguage of the legal language L. For this reason, other language properties have been used for decentralized supervision. A language property that is closed under union is *strong decomposability*.

Definition 1 *Assume that $L \subseteq L(G)$ is prefix-closed. L is* **strongly decomposable** *(with respect to $M_1, \ldots, M_n$) if*

$$L = L(G) \cap \left[M_1^{-1} M_1 L \cup \ldots \cup M_n^{-1} M_n L\right]. \tag{13}$$

If a language is strongly decomposable, then it is co-observable. It is easy to show that a language is strongly decomposable if and only if it is normal with respect to every local mask function. In other words, when a language L is strongly decomposable, then every local supervisor can ascertain the membership of every string in L based on its own local observation of the string.

Both strong decomposability and controllability are closed under union, so there exists a supremal closed controllable and strongly decomposable sublanguage of any closed sublanguage L [9]. This language is denoted *supCCSD*(L).

4. The Global Problem with Fusion

We define the Global Problem with Fusion (GPF). Given $n > 0$ and:

1. the plant $G = (Q, \Sigma, \delta, q_0)$,
2. n local mask functions $M_i : \Sigma \to \Lambda_i \cup \{\epsilon\}$ $(1 \leq i \leq n)$,
3. n local controllable event sets $\Sigma_{ci} \subseteq \Sigma$, $(1 \leq i \leq n)$,
4. closed languages A, L, with $A \subseteq L \subseteq L(G)$, and
5. a set of event fusion rules $\{\mathcal{F}_\sigma\}_{\sigma \in \Sigma_c}$, where $\mathcal{F}_\sigma : 2^{I_\sigma} \to \{\emptyset, \{\sigma\}\}$,

find a set of local supervisors $\{\gamma_{M_i}\}_{i=1}^n$, where $\gamma_{M_i} : M_i L(G) \to 2^{\Sigma_{ci}}$, such that

$$A \subseteq L(G, \{\gamma_M\}_\mathcal{F}) \subseteq L. \tag{14}$$

The Special Global Problem with Fusion (SGPF) is obtained from GPF by setting $A = \emptyset$.

We assume that $L \subseteq L(G)$ is closed and regular. Let $K = L^\uparrow$. Assume that K is not empty. Both K and L are closed regular languages. Let γ be such that $L(G, \gamma) = K$ and for all $s \in K$, $\sigma \in \gamma(s)$ implies that $s\sigma \in L(G)$. (Hence $\gamma(s)$ contains no 'extra' events.)

4.1. Local supervisor synthesis

We use the following local supervision scheme, identical for all local supervisors. Suppose local supervisor j observes the string l. It concludes that some string in $K \cap M_j^{-1} l$ has occurred, so it decides to disable every event $\sigma \in \Sigma_{cj}$ for which there exists a string $s \in K \cap M_j^{-1} l$ such that $s\sigma \in L(G)$ and $s\sigma \notin K$. Note that this supervision scheme does not depend on the (given) fusion rule.

Theorem 2 *Define a set of n local supervisors as follows:*

$$(\forall i) \ \left[(\forall l \in M_i K) \ \gamma_{M_i}(l) := \Sigma_{ci} \cap \bar{\gamma}(K \cap M_i^{-1} l)\right]. \tag{15}$$

Then for any global hybrid fusion rule $\mathcal{F}$,

$$supCCSD(K) \subseteq L(G, \{\gamma_M\}_\mathcal{F}) \subseteq K. \tag{16}$$

Proof Let (for all i)

$$(\forall l \in M_i K) \ \gamma_{M_i}(l) = \Sigma_{ci} \cap \bar{\gamma}(K \cap M_i^{-1} l). \tag{17}$$

Let $N_\mathcal{F} = L(G, \{\gamma_M\}_\mathcal{F})$ and let $N_\mathcal{A} = L(G, \{\gamma_M\}_\mathcal{A})$.

Let $\bar{K}$ be any nonempty closed controllable and strongly decomposable sublanguage of K. We show that $\bar{K} \subseteq N_\mathcal{F}$. Suppose that $s\sigma \in \bar{K}$, $s \in \bar{K}$, and $s \in N_\mathcal{F}$. Suppose, for the sake of contradiction, that $s\sigma \notin N_\mathcal{F}$. Then there is an i such that $\sigma \in \gamma_{M_i}(M_i s)$. Hence $\sigma \in \Sigma_{ci}$ and $\sigma \in \bar{\gamma}(K \cap M_i^{-1} M_i s)$. So there exists a string s' for which $M_i s' = M_i s$, $s'\sigma \in L(G)$, $s' \in K$ and and $s'\sigma \notin K$. But $s\sigma \in \bar{K}$ and $M_i(s'\sigma) = M_i(s\sigma)$, hence $s'\sigma \in M_i^{-1} M_i \bar{K}$. Since $\bar{K}$ is strongly decomposable, we have that $s'\sigma \in \bar{K}$, and since $\bar{K} \subseteq K$, this is a contradiction with $s'\sigma \notin K$. Hence $s\sigma \in N_\mathcal{F}$. We conclude that every closed controllable and strongly decomposable sublanguage of K is a subset of $N_\mathcal{F}$, and in particular, so is *supCCSD*(K).

We now show that $N_\mathcal{F} \subseteq N_\mathcal{A} \subseteq K$. Suppose that $s \in N_\mathcal{F}$, $s \in N_\mathcal{A}$, and $s\sigma \in N_\mathcal{F}$. We want to show that $s\sigma \in N_\mathcal{A}$. Suppose for the sake of contradiction that $s\sigma \notin N_\mathcal{A}$, or equivalently, that $\sigma \in \gamma_\mathcal{A}(s)$. Hence for all $j \in I_\sigma$, we have that $\sigma \in \gamma_{M_j}$. By the definition of a *k-out-of-n* event fusion rule, we have that $\sigma \in \gamma_\mathcal{F}(s)$, since $n_\sigma \geq k_\sigma$. Consequently $s\sigma \notin N_\mathcal{F}$. This is a contradiction, hence $s\sigma \in N_\mathcal{A}$. Thus $N_\mathcal{F} \subseteq N_\mathcal{A}$. Let $s \in N_\mathcal{A}$, $s \in K$, and $s\sigma \in N_\mathcal{A}$. Then $\sigma \notin \gamma_\mathcal{A}(s)$. If $\sigma \in \Sigma_u$, then $s\sigma \in K$ since K is controllable. If $\sigma \in \Sigma_c$, then there exists a j such that $\sigma \in \Sigma_{cj}$ and $\sigma \notin \gamma_{M_j}(M_j s)$. Hence (by definition of γ_{M_j}) there does not exist a string $s' \in K$ for which $s'\sigma \notin K$, $s'\sigma \in L(G)$ and $M_j s' = M_j s$. If $s\sigma \notin K$, then $s_j = s$ is such a string. This is a contradiction, so it must be that $s\sigma \in K$. Hence $N_\mathcal{A} \subseteq K$. □

The local supervisors defined in (15) satisfy several desirable properties. First, the designs of the local supervisors are *decoupled*. In general, the design of the local supervisors is coupled. A fusion rule takes as input local disabling decisions of many supervisors; the local disabling decisions *taken together* determine the global decision and should generally be designed simultaneously and interdependently. In our design, however, the local supervisors are designed independently, and it is therefore possible to distribute the computation of local supervisors among the n supervisors.

Another desirable property of the local supervisors defined in (15) is that they may be synthesized *on-line*.

0-7803-3835-9/97/$10.00 (c) 1997 AACC

In general, disabling decisions in the centralized case are affected only by disabling decisions made in the past. In contrast the decentralized case, a disabling decision for one local supervisor in the present is affected by another local supervisor's *future* disabling decision. This undesirable property is the result of events not being always observed by all supervisors. Hence, computing local supervisors is not a causal operation – future information is needed for a disabling decision in the present. In our design, the design can be performed (that is, the local supervisors can be synthesized) in a causal manner. This means that the values of every function γ_{M_i} can be obtained by (i) computing $\gamma_{M_i}(\epsilon)$, then (ii) computing $\gamma_{M_i}(l\lambda)$ (where $l\lambda \in M_iL(G)$) at the time λ is observed. The functions $\{\gamma_{M_i}\}$ can be synthesized on-line.

5. Conclusions and Future Work

We have presented a new framework for decentralized supervisory control of discrete-event systems. In this framework, local supervisors do not directly disable events in the plant; rather, they submit their disabling decisions to a *fusion rule* that combines them into a single global disabling decision. In previous work, it was assumed that local disabling decisions follow an OR rule: an event is disabled globally whenever any local supervisor wishes to disabled it. With these new results, other fusion rules are possible.

We have shown how to construct a set of local supervisors (in an on-line distributed manner) that generate a sublanguage of the legal language *for any fusion rule in a family of fusion rules*. In other words, as long as the given fusion rule is a member of a certain family, these local supervisors will yield a closed-loop language that is a sublanguage of the legal language. This particular solution actually solves a (seemingly) more difficult problem, that is, how do we design local supervisors if only the *structure* of fusion rule is known, but the rule itself *is not known*. Specifically, we consider the case where each event is globally disabled according to *any* hybrid *k-out-of-n* type fusion rule (where different controllable events may have different values of k and n.) The languages that we generate satisfy (for any fusion rule in this family) the property that they contain a language proposed in the past as a sublanguage that can be generated by a set of local supervisors. This language is the *supremal closed controllable and strongly decomposable* sublanguage of L, where L is the legal language.

The next step in this study will be to develop supervisors for classes of languages larger than those synthesized here.

References

[1] Randy Cieslak, C. Desclaux, Ayman S. Fawaz, and Pravin Varaiya. Supervisory control of discrete-event processes with partial observations. *IEEE Trans. Autom. Ctrl.*, 33(3):249–260, March 1988.

[2] F. Lin and W. M. Wonham. Decentralized supervisory control of discrete-event systems. *Info. Sci.*, 44:199–224, 1988.

[3] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Info. Sci.*, 44:173–198, 1988.

[4] Joseph H. Prosser. *Supervisor Synthesis for Partially Observed Discrete-Event Systems.* PhD thesis, Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA, September 1996.

[5] P. J. Ramadge and W. M. Wonham. Modular feedback logic for discrete-event systems. *SIAM J. Ctrl. Optim.*, 25(5):1202–1218, September 1987.

[6] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM J. Ctrl. Optim.*, 25(1):206–230, January 1987.

[7] Peter J. G. Ramadge and W. Murray Wonham. The control of discrete-event systems. *Proc. IEEE*, 77(1):81–97, January 1989.

[8] Karen Rudie and W. Murray Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Trans. Autom. Ctrl.*, 37(11):1692–1708, November 1992.

[9] Karen Gail Rudie. *Decentralized Control of Discrete-Event systems.* PhD thesis, Graduate Department of Electrical Engineering, University of Toronto, Toronto, Ontario, Canada, 1992.

[10] W. M. Wonham and P. J. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM J. Ctrl. Optim.*, 25(3):637–659, May 1987.

[11] W. M. Wonham and P. J. Ramadge. Modular supervisory control of discrete-event systems. *Math. Ctrl., Signals, and Systems*, 1(1):13–30, 1988.

0-7803-3835-9/97/$10.00 (c) 1997 AACC