On-Line Decentralized Supervisory Control of Discrete Event Systems

Shigemasa Takai[†] and Toshimitsu Ushio[‡]
[†] Department of Opto-Mechatronics, Wakayama University,
930 Sakaedani, Wakayama 640-8510, Japan
takai@sys.wakayama-u.ac.jp
[‡] Department of Systems and Human Science, Osaka University,
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
ushio@sys.es.osaka-u.ac.jp

Abstract

In this paper, we study decentralized supervisory control of discrete event systems where local disabling actions are fused by the OR rule. We generalize an on-line procedure for synthesizing decentralized supervisors proposed by Prosser.

1 Introduction

In this paper, we study *on-line* decentralized control of discrete event systems (DESs) in the framework of supervisory control proposed by Ramadge and Wonham [3]. Kozák and Wonham have proposed an on-line decentralized supervisor, called the fully decentralized supervisor [1]. Since the fully decentralized supervisor is synthesized without using the past control actions, it is somewhat restrictive. By taking account of the past control actions, Prosser have proposed an on-line procedure for synthesizing a decentralized supervisor where, for each controllable event, local disabling actions are fused by the OR, AND or majority rules [2]. In this paper, we generalize the Prosser's procedure when all events are controlled under the OR rule.

2 Preliminaries

We consider a DES modeled by an automaton $G = (Q, \Sigma, \delta, q_0)$, where Q is the set of states, Σ is the finite set of events, a partial function $\delta : \Sigma \times Q \to Q$ is the transition function, and $q_0 \in Q$ is the initial state. Let Σ^* be the set of all finite strings of elements of Σ, including the empty string ε. The function δ can be generalized to $\delta : \Sigma^* \times Q \to Q$ in the natural way. The language generated by G is denoted by $L(G)$. This paper assumes that a control specification is given by a (prefix-)closed sublanguage $K \subseteq L(G)$.

The event set Σ is decomposed into two subsets Σ_c and Σ_u of controllable and uncontrollable events, respectively [3]. A (centralized) supervisor is defined by a map $\gamma : L(G) \to 2^{\Sigma_c}$. For each string $s \in L(G)$, $\gamma(s)$ is the set of controllable events which are disabled by the supervisor γ after s has occurred. The language generated under the control actions of γ [3] is denoted by $L(G, \gamma)$. Without loss of generality, we assume that a supervisor γ satisfies, for any $s \in L(G)$,

$$\gamma(s) = \begin{cases} \{\sigma \in \Sigma_c |\ s\sigma \in L(G) - L(G, \gamma)\} \\ \qquad \text{if } s \in L(G, \gamma), \\ \emptyset \\ \qquad \text{otherwise.} \end{cases}$$

Let $K^\uparrow$ be the supremal closed and controllable sublanguage of K. Throughout this paper, we assume that $K^\uparrow \neq \emptyset$. Let $\gamma^\uparrow : L(G) \to 2^{\Sigma_c}$ be the supervisor such that $L(G, \gamma) = K^\uparrow$.

In this paper, we study decentralized supervisory control where n local supervisors jointly control the system G. We assume that Σ_c is decomposed into n subsets Σ_{ic} ($i \in I = \{1, 2, \cdots, n\}$), which are not necessarily pairwise disjoint. The set Σ_{ic} consists of controllable events for an ith local supervisor. Let $M_i : \Sigma \to \Lambda_i \cup \{\varepsilon\}$ be the local observation mask from the event set Σ to the locally observable event set Λ_i. The mask M_i is generalized to $M_i : \Sigma^* \to \Lambda_i^*$ in the natural way. Let $\Sigma_i^{\lambda_i} = \{\sigma \in \Sigma |\ M_i(\sigma) = \lambda_i\}$ for each $\lambda_i \in \Lambda_i$ and $\Sigma_i^{\varepsilon} = \{\sigma \in \Sigma |\ M_i(\sigma) = \varepsilon\}$. An ith local supervisor γ_i is formally defined by $\gamma_i : M_i(L(G)) \to 2^{\Sigma_{ic}}$. Each γ_i disables events in $\gamma_i(M_i(s))$ after a string $s \in L(G)$ has occurred. A set $\{\gamma_i\}_{i \in I}$ of n local supervisors is called a decentralized supervisor. An event $\sigma \in \Sigma_c$ is disabled in the closed-loop system if at least one local supervisor γ_i disables it. The language generated by the closed-loop system [5] is denoted by $L(G, \{\gamma_i\}_{i \in I})$.

We consider the problem to synthesize an on-line decentralized supervisor $\{\gamma_i\}_{i \in I}$ such that $L(G, \{\gamma_i\}_{i \in I}) \subseteq K$ for the closed language specification $K \subseteq L(G)$.

3 On-Line Decentralized Control

We assume that $K^{\uparrow}$ is regular, which implies that there exists a finite automaton $S = (X, \Sigma, \xi, x_0)$ such that $L(S) = K^{\uparrow}$. The transition function ξ is generalized to $\xi : 2^{\Sigma^*} \times 2^X \to 2^X$ as follows:

$$\xi(L, X') = \{x \in X |\ \exists s \in L,\ x' \in X';\ \xi(s, x') = x\}.$$

Without loss of generality, we assume that S is a subautomaton of G. Then, the supervisor $\gamma^{\uparrow} : L(G) \to 2^{\Sigma_c}$ is equivalently defined as $\gamma^{\uparrow} : X \to 2^{\Sigma_c}$ with

$$\gamma^{\uparrow}(x) := \gamma^{\uparrow}(s),$$

where $x = \delta(s, x_0)$. For any subset $X' \subseteq X$, let $\gamma^{\uparrow}(X') := \bigcup_{x \in X'} \gamma^{\uparrow}(x)$.

Let $\{\gamma_i\}_{i \in I}$ be any decentralized supervisor. Each local supervisor γ_i estimates the current state of S based on a locally observable event string in $M_i(L(G))$ in order to compute the control action. For all $i \in I$, let $\hat{\xi}_{\gamma_i}(\varepsilon) := \{x_0\}$ and

$$\hat{\xi}_{\gamma_i}(l_i \lambda_i) := \xi(\Sigma_i^{\lambda_i} - \gamma_i(l_i), \hat{\xi}^+_{\gamma_i}(l_i)) \quad (\forall l_i \lambda_i \in M_i(L(G))),$$

where $\hat{\xi}^+_{\gamma_i}(l_i) = \xi([\Sigma_i^{\varepsilon} - \gamma_i(l_i)]^*, \hat{\xi}_{\gamma_i}(l_i))$. Each γ_i estimates that the current state of S is in $\hat{\xi}_{\gamma_i}(l_i)$ just after it observes $l_i \in M_i(L(G))$.

Recall that an event is disabled in the closed-loop system if at least one local supervisor disables it. So we define a subset $\tilde{\Sigma}_{ic} \subseteq \Sigma_{ic}$ for all $i \in I$ such that

$$\bigcup_{i \in I} \tilde{\Sigma}_{ic} = \Sigma_c$$

and

$$\tilde{\Sigma}_{ic} \cap \tilde{\Sigma}_{jc} = \emptyset \ (i \neq j).$$

Intuitively, $\tilde{\Sigma}_{ic}$ is the set of controllable events such that the ith local supervisor has to control them. For each $i \in I$, we define a map $\tilde{M}_i : \Sigma^* \to 2^{\Sigma^*}$ as follows [4]:

$$\tilde{M}_i(s) = \begin{cases} \{\varepsilon\} \\ \qquad \text{if } s = \varepsilon, \\ \{t'\sigma \in \Sigma^* |\ M_i(t') = M_i(s')\} \\ \qquad \text{if } s = s'\sigma\ (\sigma \in \Sigma). \end{cases}$$

Theorem 1 Let $\tilde{\Sigma}_{ic} \subseteq \Sigma'_{ic} \subseteq \Sigma_{ic}$ $(\forall i \in I)$. For each $l_i \in M_i(L(G, \{\gamma_i\}_{i \in I}))$ $(\forall i \in I)$, let $\Gamma_i^{l_i} \subseteq \Sigma'_{ic}$,

$$\bar{\Gamma}_i^{l_i} = \Gamma_i^{l_i} \cap \gamma^{\uparrow}(\xi([\Sigma_i^{\varepsilon} - \Gamma_i^{l_i}]^*, \hat{\xi}_{\gamma_i}(l_i))),$$

and

$$\gamma_i(l_i) = \Sigma'_{ic} \cap \gamma^{\uparrow}(\xi([\Sigma_i^{\varepsilon} - \bar{\Gamma}_i^{l_i}]^*, \hat{\xi}_{\gamma_i}(l_i))).$$

Then, $K_{low} \subseteq L(G, \{\gamma_i\}_{i \in I}) \subseteq K$ holds, where

$$K_{low} := K^{\uparrow} - \left\{ \bigcup_{i \in I} \tilde{M}_i((L(G) - K^{\uparrow}) \cap K^{\uparrow}\Sigma'_{ic}) \right\} \Sigma^*.$$

See [6] for a proof of the theorem. Note that the Prosser's procedure [2] is a special case of Theorem 1 where $\Sigma'_{ic} = \Sigma_{ic}$. By choosing the sets $\tilde{\Sigma}_{ic}$, Σ'_{ic} and $\Gamma_i^{l_i}$ for each $l_i \in M_i(L(G, \{\gamma_i\}_{i \in I}))$ $(\forall i \in I)$, we obtain a decentralized supervisor $\{\gamma_i\}_{i \in I}$. For each $i \in I$, Σ'_{ic} is the set of controllable events such that γ_i controls them. The condition $\tilde{\Sigma}_{ic} \subseteq \Sigma'_{ic}$ ensures that each controllable event is controlled by at least one local supervisor. The set $\Gamma_i^{l_i}$ is a parameter introduced for a technical reason.

Remark 1 The lower bound K_{low}, which does not appear in [2], is larger than the generated language under fully decentralized supervision [1].

The decentralized supervisor given in Theorem 1 can be computed on-line. Each local supervisor independently computes the next control action when it observes a locally observable event. The computational complexity of computing the next control action is $O(|\Sigma||X|)$ [2].

4 Concluding Remarks

We have generalized the on-line procedure for synthesizing decentralized supervisors proposed by Prosser [2]. By using the generalized procedure, we can achieve a sublanguage of a specification which is not achieved by a class of decentralized supervisors synthesized by the Prosser's procedure [6].

References

[1] P. Kozák and W. M. Wonham, "Fully decentralized solutions of supervisory control problems," *IEEE Trans. Automat. Contr.*, vol. 40, no. 12, pp. 2094–2097, 1995.

[2] J. H. Prosser, "Supervisor synthesis for partially observed discrete event systems," Ph.D dissertation, Dept. of Elec. and Comp. Eng., Drexel Univ., 1996.

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

[4] K. Rudie and W. M. Wonham, "The infimal prefix-closed and observable superlanguage of a given language," *Syst. Contr. Lett.*, vol. 15, no. 5, pp. 361–371, 1990.

[5] K. Rudie and W. M. Wonham, "Think globally, act locally: decentralized supervisory control," *IEEE Trans. Automat. Contr.*, vol. 37, no. 11, pp. 1692–1708, 1992.

[6] S. Takai and T. Ushio, "On-line synthesis of decentralized supervisors for discrete event systems," *IEICE Trans. Fundamentals*, vol. E83-A, no. 11, 2000.