Relative Coobservability in Decentralized Supervisory Control of Discrete-Event Systems

Article *in* International Journal of Control · April 2016

DOI: 10.1080/00207179.2017.1397754

3 authors, including:

Renyuan Zhang
Northwestern Polytechnical University

Walter Wonham
University of Toronto

Relative Coobservability in Decentralized Supervisory Control of Discrete-Event Systems

Kai Cai, Renyuan Zhang, and W.M. Wonham

***Abstract*—We study the new concept of *relative coobservability* in decentralized supervisory control of discrete-event systems under partial observation. This extends our previous work on relative observability from a centralized setup to a decentralized one. A fundamental concept in decentralized supervisory control is coobservability (and its several variations); this property is not, however, closed under set union, and hence there generally does not exist the supremal element. Our proposed relative coobservability, although stronger than coobservability, is algebraically well-behaved, and the supremal relatively coobservable sublanguage of a given language exists. We present an algorithm to compute this supremal sublanguage. Moreover, relative coobservability is weaker than conormality, which is also closed under set union; unlike conormality, relative coobservability imposes no constraint on disabling unobservable controllable events.**

***Index Terms*—Supervisory control, discrete-event systems, decentralized supervision, relative coobservability, partial observation, automata.**

I. Introduction

Recently we introduced the new concept of *relative observability* in supervisory control of discrete-event systems (DES) under partial observation (see [1] and its conference precursor [2]; also the timed case [3]). Relative observability is stronger than observability, weaker than normality, and preserved under set union; hence there exists the supremal relatively observable sublanguage of a given language, which may be effectively computed. Relative observability is formulated in a centralized setup where a monolithic supervisor partially observes and controls the plant as a whole.

In this paper and its conference precursor [4], we extend relative observability to a *decentralized* setup where multiple decentralized supervisors operate jointly, each of which observes and controls only part of the plant. Decentralized supervisory control is an effective means of managing computational complexity when DES are large-scale (e.g. [5, Chapter 4]). Our work is motivated by the fact that, in decentralized control under partial observation, there has so far lacked an effective concept for which the supremal decentralized supervisors may be computed, unless normality constraints are imposed which might be overly conservative.

K. Cai is with Urban Research Plaza, Osaka City University, Japan. R. Zhang is with Department of Traffic and Control Engineering, Northwestern Polytechnical University, China. W.M. Wonham is with Department of Electrical and Computer Engineering, University of Toronto, Canada. Emails: kai.cai@eng.osaka-cu.ac.jp, ryzhang@nwpu.edu.cn, wonham@control.utoronto.ca. This work was supported in part by Program to Disseminate Tenure Tracking System, MEXT, Japan; the National Nature Science Foundation of China, Grant no. 61403308; the Natural Sciences and Engineering Research Council, Canada, Grant no. 7399.

The fundamental concept in decentralized supervisory control is *coobservability*, identified in [6] (see also [7]): coobservability and controllability of a language K is necessary and sufficient for the existence of *nonblocking* decentralized supervisors that synthesize K. Here the decentralized supervisors follow a *conjunctive* decision fusion rule: an event is enabled if and only if *all* supervisors 'agree' to enable that event. One may also consider alternative fusion rules, e.g. that of *disjunctive*, or a mix of conjunctive and disjunctive; these lead to variations of coobservability studied in [8]. A further extension called conditional coobservability is reported in [9].

None of the above various versions of coobservability, however, is closed under set union; consequently there generally does not exist the supremal coobservable sublanguage of a given language. In fact, even the existence of a coobservable sublanguage is undecidable in general [10]. On the other hand, *conormality* (or strong decomposability), being stronger than coobservability, is proposed in [6]; it is preserved under set union and the supremal conormal sublanguage may be computed. Conormality, however, imposes the constraint that no decentralized supervisor can disable its unobservable, controllable events, and may therefore be overly conservative in practice. There is a weaker version of conormality studied in [11], which is also closed under set union; however, no algorithm is presented to compute the supremal element.

In this paper, we introduce the new concept of *relative coobservability*, which is a natural extension of relative observability to the decentralized supervisory control setup. We prove that relative coobservability is stronger than (any of the known variations of) coobservability, weaker than (weak) conormality, and closed under set union. Moreover, we present an algorithm for computing the supremal relatively coobservable (and controllable, $L_m(\mathbf{G})$-closed) sublanguage of a given language. This algorithm is so far the only one that effectively synthesizes nonblocking controlled behavior that is generally more permissive than the conormal counterpart. The new concept and algorithm are demonstrated with a Guideway example.

We note that [12] introduced three concepts called strong conjunctive coobservability, strong disjunctive coobservability, and strong local observability; the latter two are proved to be closed under set union. First, for strong local observability, we will see that it is in fact a special case of our relative coobservability. Then for strong disjunctive coobservability, although weaker than our relative coobservability, there is no existing finitely convergent algorithm that computes its supremal element. By contrast, we will present an algorithm that effectively computes the supremal relatively coobservable

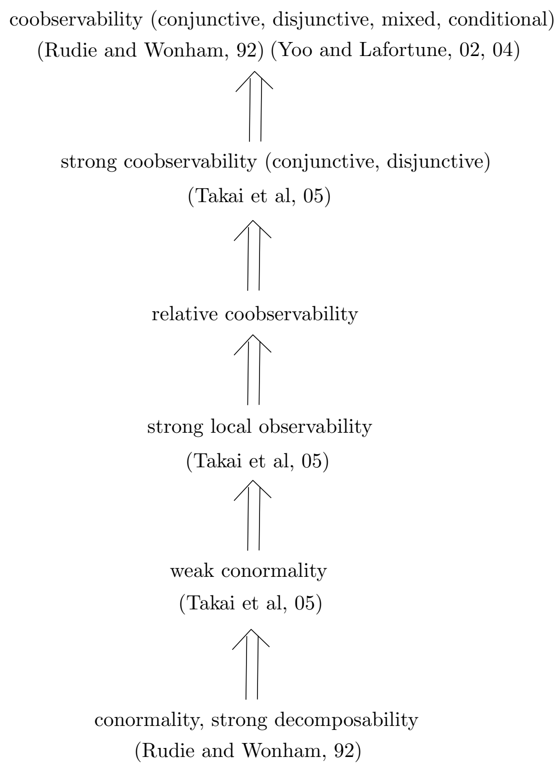


Fig. 1. Observability concepts and their relations in decentralized supervisory control under partial observation: bottom to top, strong to weak. For all coobservability concepts weaker than relative coobservability, no effective algorithm exists that computes the corresponding nonblocking controlled behavior.

sublanguage. The relations of relative coobservability and other concepts reported in decentralized supervisory control are summarized in Fig. 1.

Note also that, for prefix-closed languages, several procedures are developed to compute maximal decentralized supervisors, e.g. [13], [14]. Those procedures are not, however, applicable to non-closed languages, because the resulting decentralized supervisors may be blocking.

Finally we point out that the supremal relatively coobservable sublanguage of a given language K may be empty even if there exists a nonempty coobservable sublanguage of K (whether or not K is prefix-closed). Nevertheless, whenever the supremal relatively coobservable sublanguage is nonempty (and therefore can be computed by our proposed algorithm), it is guaranteed to be coobservable, and nonblocking decentralized supervisors may be constructed accordingly [6].

The rest of the paper is organized as follows. In Section II we introduce the new concept of relative coobservability and show that it is stronger than coobservability (and its variations) and weaker than conormality. In Section III we prove that relative coobservability is closed under set union, and present an algorithm to compute the supremal relatively coobservable sublanguage of a given language. The results are demonstrated with a Guideway example in Section IV. Finally in Section V we state our conclusions.

II. Relative Coobservability

The plant to be controlled is modeled by a generator

$$\mathbf{G} = (Q, \Sigma, \delta, q_0, Q_m) \tag{1}$$

where Q is the finite state set; $q_0 \in Q$ the initial state; $Q_m \subseteq Q$ the subset of marker states; Σ the finite event set; $\delta : Q \times \Sigma \to Q$ the (partial) state transition function. In the usual way, δ is extended to $\delta : Q \times \Sigma^* \to Q$, and we write $\delta(q, s)!$ to mean that $\delta(q, s)$ is defined. The *closed behavior* of $\mathbf{G}$ is the language

$$L(\mathbf{G}) := \{s \in \Sigma^* | \delta(q_0, s)!\} \subseteq \Sigma^* \tag{2}$$

and the *marked behavior* is

$$L_m(\mathbf{G}) := \{s \in L(\mathbf{G}) | \delta(q_0, s) \in Q_m\} \subseteq L(\mathbf{G}). \tag{3}$$

A string s_1 is a *prefix* of a string s, written $s_1 \leq s$, if there exists s_2 such that $s_1 s_2 = s$. The *(prefix) closure* of $L_m(\mathbf{G})$ is $\overline{L_m(\mathbf{G})} := \{s_1 \in \Sigma^* \mid (\exists s \in L_m(\mathbf{G})) s_1 \leq s\}$. In this paper we assume $\overline{L_m(\mathbf{G})} = L(\mathbf{G})$; namely $\mathbf{G}$ is *nonblocking*. A language $K \subseteq \Sigma^*$ is $L_m(\mathbf{G})$-*closed* if $\overline{K} \cap L_m(\mathbf{G}) = K$.

For partial observation, let the event set Σ be partitioned into Σ_o, the observable event subset, and Σ_{uo}, the unobservable subset (i.e. $\Sigma = \Sigma_o \dot\cup \Sigma_{uo}$). Bring in the *natural projection* $P : \Sigma^* \to \Sigma_o^*$ defined according to

$$\begin{aligned} &P(\epsilon) = \epsilon, \quad \epsilon \text{ is the empty string;} \\ &P(\sigma) = \begin{cases} \epsilon, & \text{if } \sigma \notin \Sigma_o, \\ \sigma, & \text{if } \sigma \in \Sigma_o; \end{cases} \\ &P(s\sigma) = P(s)P(\sigma), \quad s \in \Sigma^*, \sigma \in \Sigma. \end{aligned} \tag{4}$$

In the usual way, P is extended to $P : Pwr(\Sigma^*) \to Pwr(\Sigma_o^*)$, where $Pwr(\cdot)$ denotes *powerset*. Write $P^{-1} : Pwr(\Sigma_o^*) \to Pwr(\Sigma^*)$ for the *inverse-image function* of P.

Let $\Sigma_{o,i} \subseteq \Sigma$ and the natural projections $P_i : \Sigma^* \to \Sigma_{o,i}^*$, $i \in \mathcal{I}$ ($\mathcal{I}$ is some index set). Also let $\Sigma_{c,i} \subseteq \Sigma$. We consider decentralized supervisory control where each decentralized supervisor $i \in \mathcal{I}$ observes events only in $\Sigma_{o,i}$, and controls events only in $\Sigma_{c,i}$. Then let $\Sigma_c := \cup_{i \in \mathcal{I}} \Sigma_{c,i}$ be the total controllable event subset, and $\Sigma_u := \Sigma \setminus \Sigma_c$ the uncontrollable subset. A language $K \subseteq \Sigma^*$ is *controllable* with respect to $\mathbf{G}$ if

$$\overline{K}\Sigma_u \cap L(\mathbf{G}) \subseteq \overline{K}. \tag{5}$$

For conceptual simplicity let us first consider the case of two decentralized supervisors, i.e. $\mathcal{I} = \{1, 2\}$. The (conjunctive) coobservability is defined as follows [6]. A language $K \subseteq L_m(\mathbf{G})$ is *coobservable* with respect to $\mathbf{G}$, P_1, P_2, $\Sigma_{c,1}$, $\Sigma_{c,2}$ if

$$(\forall s, s', s'' \in \Sigma^*)\ P_1(s) = P_1(s') \wedge P_2(s) = P_2(s'') \Rightarrow$$

(i) $(\forall \sigma \in \Sigma_{c,1} \cap \Sigma_{c,2})$

$$\begin{aligned} &(s'\sigma \in \overline{K} \wedge s \in \overline{K} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K}) \\ &\vee (s''\sigma \in \overline{K} \wedge s \in \overline{K} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K}) \end{aligned} \tag{6}$$

(ii) $(\forall \sigma \in \Sigma_{c,1} \setminus \Sigma_{c,2})$

$$s'\sigma \in \overline{K} \wedge s \in \overline{K} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K} \tag{7}$$

(iii) $(\forall \sigma \in \Sigma_{c,2} \setminus \Sigma_{c,1})$

$$s''\sigma \in \overline{K} \wedge s \in \overline{K} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K} \tag{8}$$

First observe that (ii) (resp. (iii)) above, for a controllable event σ belonging only to $\Sigma_{c,1}$, i.e. $\sigma \in \Sigma_{c,1} \setminus \Sigma_{c,2}$ (resp. $\sigma \in \Sigma_{c,2} \setminus \Sigma_{c,1}$), is simply the standard observability condition [15] with respect to P_1 (resp. P_2) that is applied. For a shared controllable event $\sigma \in \Sigma_{c,1} \cap \Sigma_{c,2}$ in (i) above, on the other hand, both observations P_1 and P_2 are involved, and the condition (6) is equivalent to

$$s'\sigma \in \overline{K} \wedge s''\sigma \in \overline{K} \wedge s \in \overline{K} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K}$$

namely the decision of enabling σ after string s will be made if it is first ratified by both supervisors working through their respective observation channels.

Coobservability, together with controllability and $L_m(\mathbf{G})$-closedness, of a language K is shown to be necessary and sufficient for the existence of two decentralized supervisors *conjunctively* synthesizing K [6]. Coobservability, however, is not closed under set union, and consequently the supremal coobservable sublanguage of K need not exist in general. This fact motivates us to propose the new concept, *relative coobservability*, which (as we will show) is algebraically better behaved.

Definition 1. *Let $C \subseteq L_m(\mathbf{G})$ be a fixed* ambient *sublanguage. A sublanguage $K \subseteq C$ is* relatively coobservable, *or simply $\overline{C}$*-coobservable, *with respect to* $\mathbf{G}$, P_1, P_2, $\Sigma_{c,1}$, $\Sigma_{c,2}$ *if*

$$(\forall s, s', s'' \in \Sigma^*)\ P_1(s) = P_1(s') \wedge P_2(s) = P_2(s'') \Rightarrow$$

(i) $(\forall \sigma \in \Sigma_{c,1} \cap \Sigma_{c,2})$

$$(s'\sigma \in \overline{K} \wedge s \in \overline{C} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K}) \wedge (s''\sigma \in \overline{K} \wedge s \in \overline{C} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K}) \quad (9)$$

(ii) $(\forall \sigma \in \Sigma_{c,1} \setminus \Sigma_{c,2})$

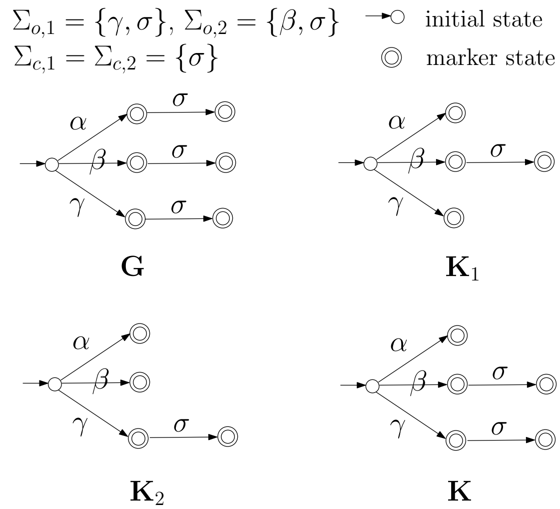
$$s'\sigma \in \overline{K} \wedge s \in \overline{C} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K} \quad (10)$$

(iii) $(\forall \sigma \in \Sigma_{c,2} \setminus \Sigma_{c,1})$

$$s''\sigma \in \overline{K} \wedge s \in \overline{C} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K} \quad (11)$$

Several remarks on the definition are in order. First, relative coobservability is a 'strengthened' version of coobservability in two respects. For one, all strings s in the ambient $\overline{C}$ are considered, instead of just strings in $\overline{K}$. For the other, the two implications in (9) are connected by "and" $\wedge$, instead of "or" $\vee$. Namely (9) requires that the 'observational consistency' hold for *both* observation channels P_1 and P_2. This requirement is crucial to provide closure under union for relative coobservability; as the example in Fig. 2 shows, using $\vee$ in (9) would fail to guarantee closure under union.[1] Hence we have identified the two defects that cause coobservability to fail to be closed under union: (1) lack of an ambient language, (2) the use of disjunctive ("or") $\vee$ logic in connecting local observational consistency.

[1]This requirement is admittedly a shortcoming of our relative coobservability approach as it rules out any inconsistency in decentralized supervisors' local decisions. However, in the absence of such a requirement it does not seem possible to preserve the property of closure under union, and hence the effective computability of a useful result. Computation of a merely "maximal", as distinct from supremal, behavior (even if that could be achieved) would be, in our view, of little practical interest.



$$L_m(\mathbf{G}) = C = \{\alpha, \beta, \gamma, \alpha\sigma, \beta\sigma, \gamma\sigma\}$$
$$L_m(\mathbf{K}) = L_m(\mathbf{K}_1) \cup L_m(\mathbf{K}_2)$$

Fig. 2. Suppose that $\vee$ were used in (9). Then $L_m(\mathbf{K}_1)$ and $L_m(\mathbf{K}_2)$ would both be $\overline{C}$-coobservable, but the union $L_m(\mathbf{K}) = L_m(\mathbf{K}_1) \cup L_m(\mathbf{K}_2)$ would not be. The reason is as follows. First for $P_1 : \Sigma^* \to \Sigma_{o,1}^*$, let $s = \alpha$ and $s' = \beta$. Then $P_1(s) = P_1(s') = \epsilon$, $s'\sigma \in L(\mathbf{K})$, $s \in \overline{C}$, $s\sigma \in L(\mathbf{G})$, but $s\sigma \notin L(\mathbf{K})$. Second for $P_2 : \Sigma^* \to \Sigma_{o,2}^*$, let $s = \alpha$ and $s'' = \gamma$. Then $P_2(s) = P_2(s'') = \epsilon$, $s''\sigma \in L(\mathbf{K})$, $s \in \overline{C}$, $s\sigma \in L(\mathbf{G})$, but $s\sigma \notin L(\mathbf{K})$. (Notation: we will use the same initial and marker state notation in subsequent figures.)

The above two (strengthening) modifications lead immediately to the following.

Proposition 1. *If $K \subseteq C$ is $\overline{C}$-coobservable, then K is also coobservable.*

The reverse statement need not be true. For an example see again Fig. 2: $L_m(\mathbf{K}_1)$ (or $L_m(\mathbf{K}_2)$) is coobservable (since $\vee$ is used in (6)) but not relatively coobservable ($\wedge$ used in (9)).

Second, relative coobservability is a decentralized version of relative observability [1]. Indeed, for an unshared controllable event, namely (ii) and (iii) in the definition, individual relative observability conditions corresponding to the respective natural projections are applied; while for a shared controllable event, namely (i), both conditions must be satisfied simultaneously. This implies that the definition of relative coobservability is equivalent to the condition that for each $i \in \mathcal{I}$, K is relatively observable with respect to P_i, i.e.

$$(\forall s, s' \in \Sigma^*)(\forall \sigma \in \Sigma_{c,i})\ P_i(s) = P_i(s') \wedge s'\sigma \in \overline{K} \wedge s \in \overline{C} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K}. \quad (12)$$

Thus we see that Definition 1 is easily adapted to a general finite set $\mathcal{I}$ of decentralized supervisors. For this reason, we also refer to relative coobservability as *$\mathcal{I}$-fold relative observability*.

Third, consider a finite set $\mathcal{I}$ of decentralized supervisors. Relative coobservability ensures that if a decentralized supervisor enables (resp. disables) an event, then no other supervisor disables (resp. enables) that event. Namely, there is no conflict among decentralized supervisors' local control decisions, and each supervisor may independently decide to enable or disable an event based on its local observation.

Fourth, we note that the ambient language C is selected such that all the strings in $\overline{C}$ must be tested for the conditions of relative coobservability. In addition, if $C_1 \subseteq C_2 \subseteq L_m(\mathbf{G})$ are two ambient languages, it follows easily from Definition 1 that $\overline{C_2}$-coobservability implies $\overline{C_1}$-coobservability. Namely, the smaller the ambient language, the weaker the relative coobservability.

An alternative definition of coobservability that has appeared in the literature is disjunctive coobservability [8], defined as follows. A language $K \subseteq L_m(\mathbf{G})$ is *disjunctively coobservable* with respect to $\mathbf{G}$, P_1, P_2, $\Sigma_{c,1}$, $\Sigma_{c,2}$ if

$$
\begin{aligned}
&(\forall s, s', s'' \in \Sigma^*)\ P_1(s) = P_1(s') \wedge P_2(s) = P_2(s'') \Rightarrow \\
&\text{(i) } (\forall \sigma \in \Sigma_{c,1} \cap \Sigma_{c,2})\ s'\sigma \in L(\mathbf{G}) \setminus \overline{K} \wedge s''\sigma \in L(\mathbf{G}) \setminus \overline{K} \\
&\qquad \wedge s \in \overline{K} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in L(\mathbf{G}) \setminus \overline{K} \qquad (13)\\
&\text{(ii) } (\forall \sigma \in \Sigma_{c,1} \setminus \Sigma_{c,2})\ s'\sigma \in L(\mathbf{G}) \setminus \overline{K} \\
&\qquad \wedge s \in \overline{K} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in L(\mathbf{G}) \setminus \overline{K} \qquad (14)\\
&\text{(iii) } (\forall \sigma \in \Sigma_{c,2} \setminus \Sigma_{c,1})\ s''\sigma \in L(\mathbf{G}) \setminus \overline{K} \\
&\qquad \wedge s \in \overline{K} \wedge s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in L(\mathbf{G}) \setminus \overline{K} \qquad (15)
\end{aligned}
$$

Disjunctive coobservability requires that for a shared controllable event σ in (i) above, the decision of disabling σ after string s be ratified by both supervisors working through their respective observation channels. This implies that σ will be enabled if some supervisor decides to enable it, therefore the name "disjunctive". Disjunctive coobservability is different from conjunctive coobservability, and in general neither of the two versions implies the other [8].

Disjunctive coobservability, together with controllability and $L_m(\mathbf{G})$-closedness, of a language K is proved to be necessary and sufficient for the existence of two decentralized supervisors *disjunctively* synthesizing K [8]. Again, however, it is not closed under set union, and consequently the supremal element need not exist in general. We show next that our relative coobservability is stronger than disjunctive coobservability.

Proposition 2. *If $K \subseteq C$ is $\overline{C}$-coobservable, then K is also disjunctively coobservable.*

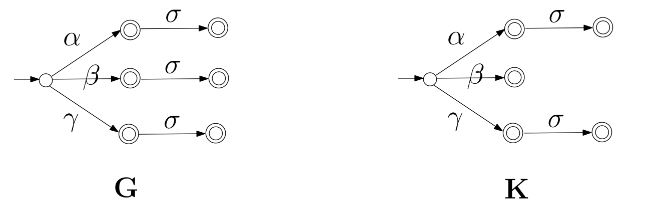
Proof. Let $s, s', s'' \in \overline{K} \subseteq \overline{C}$, $P_1(s) = P_1(s')$, and $P_2(s) = P_2(s'')$. We show that condition (i), namely (13), of disjunctive coobservability holds. Let $\sigma \in \Sigma_{c,1} \cap \Sigma_{c,2}$, $s'\sigma \in L(\mathbf{G}) \setminus \overline{K}$, $s''\sigma \in L(\mathbf{G}) \setminus \overline{K}$, and $s\sigma \in L(\mathbf{G})$. We will show that $s\sigma \in L(\mathbf{G}) \setminus \overline{K}$. From (9) we know that

$$
\begin{aligned}
&(s'\sigma \notin \overline{K} \Rightarrow s'\sigma \notin L(\mathbf{G}) \vee s' \notin \overline{C} \vee s\sigma \notin \overline{K}) \\
\wedge\ &(s''\sigma \notin \overline{K} \Rightarrow s''\sigma \notin L(\mathbf{G}) \vee s'' \notin \overline{C} \vee s\sigma \notin \overline{K}).
\end{aligned}
$$

We have $s'\sigma \notin \overline{K}$, $s'\sigma \in L(\mathbf{G})$, $s' \in \overline{C}$; and $s''\sigma \notin \overline{K}$, $s''\sigma \in L(\mathbf{G})$, $s'' \in \overline{C}$. It follows that $s\sigma \notin \overline{K}$. Since $s\sigma \in L(\mathbf{G})$, we conclude that $s\sigma \in L(\mathbf{G}) \setminus \overline{K}$.

The same reasoning proves conditions (ii) and (iii), namely (14) and (15), of disjunctive coobservability.[2] □

$\gamma, \sigma \in \Sigma_{o,1}$, $\beta, \sigma \in \Sigma_{o,2}$
$\sigma \in \Sigma_{c,1} \cap \Sigma_{c,2}$



$L_m(\mathbf{G}) = C = \{\alpha, \beta, \gamma, \alpha\sigma, \beta\sigma, \gamma\sigma\}$ $\qquad L_m(\mathbf{K}) = \{\alpha, \beta, \gamma, \alpha\sigma, \gamma\sigma\}$
$L(\mathbf{G}) = \overline{C} = \{\epsilon, \alpha, \beta, \gamma, \alpha\sigma, \beta\sigma, \gamma\sigma\}$ $\qquad L(\mathbf{K}) = \{\epsilon, \alpha, \beta, \gamma, \alpha\sigma, \gamma\sigma\}$

Fig. 3. $L_m(\mathbf{K})$ is disjunctively coobservable but not relatively coobservable. For P_1, let $s = \beta$ and $s' = \alpha$. Then $P_1(s) = P_1(s') = \epsilon$, $s'\sigma \in L(\mathbf{K})$, $s \in \overline{C}$, $s\sigma \in L(\mathbf{G})$, but $s\sigma \notin L(\mathbf{K})$. This violates (9), and therefore relative coobservability fails. For P_2, on the other hand, let $s'' = \gamma$ so that $P_2(s') = P_2(s'') = \epsilon$. The fact that $s''\sigma \in L(\mathbf{K})$ makes (13) true. One may check that disjunctive coobservability of $L_m(\mathbf{K})$ indeed holds.

The reverse statement of Proposition 2 need not be true. An example is displayed in Fig. 3, of a disjunctively coobservable language that is not relatively coobservable.

Remark 1. *We note that in [12], "strong conjunctive" and "strong disjunctive" coobservability are studied, the essence being to choose strings from the ambient language $L_m(\mathbf{G})$ instead of K. For that reason they are stronger than their respective type of coobservability. Strong disjunctive coobservability is shown to be closed under set union (while strong conjunctive coobservability is not), but no finitely convergent algorithm is given to compute the supremal element. Our relative coobservability may be shown to be stronger than these strong versions of coobservability; nevertheless we shall present an algorithm that computes the supremal relatively coobservable sublanguage of a given language.*

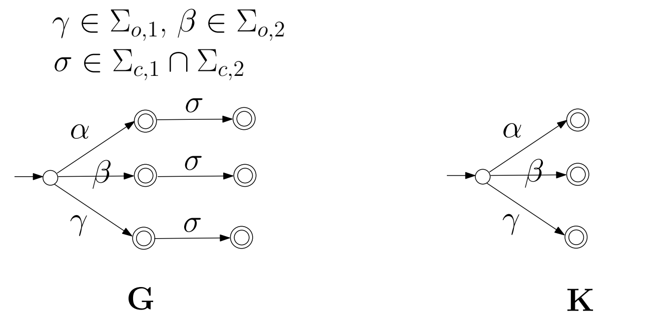
We also note in passing that since either conjunctive or disjunctive coobservability is stronger than the mixed coobservability [8], which is furthermore stronger than the conditional coobservability [9], our coobservability is stronger than all versions of coobservability reported in the literature.

We turn now to prove that relative coobservability is weaker than conormality (or strong decomposibility in [6]). A language $K \subseteq L_m(\mathbf{G})$ is *conormal* with respect to $\mathbf{G}$, P_1, P_2, $\Sigma_{c,1}$, $\Sigma_{c,2}$ if

$$\left(P_1^{-1}P_1(\overline{K}) \cup P_2^{-1}P_2(\overline{K})\right) \cap L(\mathbf{G}) = \overline{K}. \qquad (16)$$

Conormality may be overly restrictive because it requires that for each decentralized supervisor $i \in \mathcal{I}$, only observable (under P_i), controllable events may be disabled. Relative coobservability, by contrast, does not impose this restriction,

[2]That relative coobservability (or $\mathcal{I}$-fold relative observability) is stronger than disjunctive coobservability (Proposition 2) or conjunctive coobservability (Proposition 1) can also be proved by noting that it is stronger than a property called local observability [12]: local observability requires that for each $i \in \mathcal{I}$, K be observable with respect to P_i, i.e. $\mathcal{I}$-fold observability, and is proved to be stronger than disjunctive and conjunctive coobservability.



$L_m(\mathbf{G}) = C = \{\alpha, \beta, \gamma, \alpha\sigma, \beta\sigma, \gamma\sigma\}$ $\quad L_m(\mathbf{K}) = \{\alpha, \beta, \gamma\}$

$L(\mathbf{G}) = \overline{C} = \{\epsilon, \alpha, \beta, \gamma, \alpha\sigma, \beta\sigma, \gamma\sigma\}$ $\quad L(\mathbf{K}) = \{\epsilon, \alpha, \beta, \gamma\}$

Fig. 4. $L_m(\mathbf{K})$ is relatively coobservable but not conormal. A straightforward calculation shows that $\left(P_1^{-1}P_1(\overline{K}) \cup P_2^{-1}P_2(\overline{K})\right) \cap L(\mathbf{G}) = L(\mathbf{G}) \supsetneqq \overline{K}$; hence $L_m(\mathbf{K})$ is not conormal. On the other hand, by noting that the controllable event σ is removed after strings α, β, and γ, it is easily checked that $L_m(\mathbf{K})$ is relatively observable with respect to both P_1 and P_2, and therefore is relatively coobservable.

i.e. control may be exercised by each decentralized supervisor over its unobservable controllable events.

Proposition 3. *If $K \subseteq C$ is conormal with respect to $\mathbf{G}$, P_1, P_2, $\Sigma_{c,1}$, $\Sigma_{c,2}$, then K is $\overline{C}$-coobservable.*

Proof. Let $s, s', s'' \in \Sigma^*$, $P_1(s) = P_1(s')$, and $P_2(s) = P_2(s'')$. We show that (9)-(11) all hold. First for (9), let $\sigma \in \Sigma_{c,1} \cap \Sigma_{c,2}$, $s'\sigma \in \overline{K}$, $s \in \overline{C}$, and $s\sigma \in L(\mathbf{G})$; it will be shown that $s\sigma \in \overline{K}$. From $s'\sigma \in \overline{K}$ we have

$$\begin{aligned} P_1(s'\sigma) \in P_1\overline{K} &\Rightarrow P_1(s)P_1(\sigma) \in P_1\overline{K} \\ &\Rightarrow s\sigma \in P_1^{-1}P_1\overline{K} \\ &\Rightarrow s\sigma \in P_1^{-1}P_1(\overline{K}) \cup P_2^{-1}P_2(\overline{K}) \end{aligned}$$

Hence $s\sigma \in \left(P_1^{-1}P_1(\overline{K}) \cup P_2^{-1}P_2(\overline{K})\right) \cap L(\mathbf{G}) = \overline{K}$ by conormality of $\overline{K}$. Similarly, let $s''\sigma \in \overline{K}$; through P_2 we derive $s\sigma \in \overline{K}$.

For (10), let $\sigma \in \Sigma_{c,1} \setminus \Sigma_{c,2}$, $s'\sigma \in \overline{K}$, $s \in \overline{C}$, and $s\sigma \in L(\mathbf{G})$. By the same derivation as above, we get $s\sigma \in \overline{K}$. Finally for (11), let $\sigma \in \Sigma_{c,2} \setminus \Sigma_{c,1}$, $s''\sigma \in \overline{K}$, $s \in \overline{C}$, and $s\sigma \in L(\mathbf{G})$. Again by the same derivation as above but through P_2, we get $s\sigma \in \overline{K}$. □

The reverse statement of Proposition 3 need not be true; an example is displayed in Fig. 4.

Remark 2. *A weak conormality concept was studied in [11], which is proved to be weaker than conormality and also preserved under set union. However no algorithm is given to compute the supremal element. Then in [12], weak conormality is shown to be stronger than the "strong local observability". The latter is the special case of our relative coobservability with the largest possible ambient language $C = L_m(\mathbf{G})$, hence the strongest. Therefore we conclude that relative coobservability is generally weaker than weak conormality.*

III. Supremal Relatively Coobservable Sublanguage and Algorithms

First, we show that an arbitrary union of relatively coobservable languages is again relatively coobservable. Let $\mathcal{I}$ denote the set of decentralized supervisors, and P_i the natural projection for each $i \in \mathcal{I}$.

Proposition 4. *Let $K_\alpha \subseteq C \subseteq L_m(\mathbf{G})$, $\alpha \in \mathcal{A}$ (some index set), be $\overline{C}$-coobservable. Then $K = \bigcup\{K_\alpha \mid \alpha \in \mathcal{A}\}$ is also $\overline{C}$-coobservable.*

Proof. To prove that K is $\overline{C}$-coobservable, we show that K is $\overline{C}$-observable with respect to P_i for each $i \in \mathcal{I}$. Let $i \in \mathcal{I}$, $s, s' \in \Sigma^*$, $P_i(s) = P_i(s')$, $\sigma \in \Sigma_{c,i}$, $s\sigma \in \overline{K}$, $s' \in \overline{C}$, and $s'\sigma \in L(\mathbf{G})$; it will be shown that $s'\sigma \in \overline{K}$. Since $\overline{K} = \overline{\bigcup_{\alpha \in \mathcal{A}} K_\alpha} = \bigcup_{\alpha \in \mathcal{A}} \overline{K_\alpha}$, there exists $\alpha \in \mathcal{A}$ such that $s\sigma \in \overline{K_\alpha}$. Since K_α is $\overline{C}$-coobservable, it is $\overline{C}$-observable with respect to P_j for all $j \in \mathcal{I}$. In particular, K_α is $\overline{C}$-observable with respect to P_i, and thereby we derive that $s'\sigma \in \overline{K_\alpha}$. Finally $s'\sigma \in \bigcup_{\alpha \in \mathcal{A}} \overline{K_\alpha} = \overline{K}$. □

In the proof to establish closure under union for relative coobservability, it was essential that K_α ($\alpha \in \mathcal{A}$) being $\overline{C}$-coobservable means that K_α is $\overline{C}$-observable with respect to *all* channels P_j, $j \in \mathcal{I}$. This confirms the importance of using $\wedge$ in (9) in the definition of relative coobservability.

Now let $K \subseteq C \subseteq L_m(\mathbf{G})$. Whether or not K is $\overline{C}$-coobservable, write

$$\mathcal{O}(K, C) := \{K' \subseteq K \mid K' \text{ is } \overline{C}\text{-coobservable}\} \tag{17}$$

for the family of $\overline{C}$-coobservable sublanguages of K. Note that the empty language $\emptyset$ is trivially $\overline{C}$-coobservable, thus a member of $\mathcal{O}(K, C)$. By Proposition 4 we obtain that $\mathcal{O}(K, C)$ has a unique supremal element sup$\mathcal{O}(K, C)$ given by

$$\text{sup}\mathcal{O}(K, C) := \bigcup\{K' \mid K' \in \mathcal{O}(K, C)\}. \tag{18}$$

This is the supremal $\overline{C}$-coobservable sublanguage of K. We state these important facts about $\mathcal{O}(K, C)$ in the following

Theorem 1. *Let $K \subseteq C \subseteq L_m(\mathbf{G})$. The set $\mathcal{O}(K, C)$ is nonempty, and contains the supremal element sup$\mathcal{O}(K, C)$ in (18).*

Next we present an algorithm to compute sup$\mathcal{O}(K, C)$. The idea is to apply the algorithm in [1], iteratively for each P_i ($i \in \mathcal{I}$), to compute the respective supremal relatively observable sublanguage. Let $\mathbf{G}$, $\mathbf{C}$, and $\mathbf{K}$ be finite-state generators (as in (1)) with marked languages $L_m(\mathbf{G})$, C, and K, respectively.

Algorithm 1: Input $\mathbf{G}$, $\mathbf{C}$, $\mathbf{K}$, and $P_i : \Sigma^* \to \Sigma_{o,i}^*$, $i \in \mathcal{I} := \{1, ..., N\}$.
1. Set $\mathbf{K}_0 := \mathbf{K}$.
2. For $j \geq 0$, set $\mathbf{K}_{j,1} := \mathbf{K}_j$.
3. For $i \geq 1$, apply the algorithm in [1] with inputs $\mathbf{G}$, $\mathbf{K}_{j,i}$, and P_i to obtain $\mathbf{K}_{j,i+1}$ such that $L_m(\mathbf{K}_{j,i+1})$ is the supremal $\overline{C}$-observable sublanguage of $L_m(\mathbf{K}_{j,i})$ with respect to P_i. Proceed until $\mathbf{K}_{j,N}$ is computed, and set it to be $\mathbf{K}_{j+1}$. If $\mathbf{K}_{j+1} = \mathbf{K}_j$,[3] then output $\mathbf{K}^{\uparrow} := \mathbf{K}_{j+1}$. Otherwise, advance j to $j + 1$ and go to Step 2.

[3]Here $=$ means that the two generators are isomorphic [5, Chapter 3].

Algorithm 1 terminates in finite steps, because the algorithm in [1] does so and removes states and/or transitions from the finite-state generator $\mathbf{K}$. The complexity of Algorithm 1 is exponential in the state size of $\mathbf{K}$, inasmuch as the algorithm in [1] is of this complexity.

Theorem 2. *The output $\mathbf{K}^{\uparrow}$ of Algorithm 1 satisfies $L_m(\mathbf{K}^{\uparrow}) = sup\mathcal{O}(K, C)$, the supremal $\overline{C}$-coobservable sublanguage of K.*

Proof. First, it is guaranteed by Step 3 of Algorithm 1 that $L_m(\mathbf{K}^{\uparrow})$ is $\overline{C}$-observable with respect to P_i for each $i \in \mathcal{I}$. Thus $L_m(\mathbf{K}^{\uparrow}) \in \mathcal{O}(K, C)$. It remains to prove that if $K' \in \mathcal{O}(K, C)$, then $K' \subseteq L_m(\mathbf{K}^{\uparrow})$. To see this, consider induction on the iterations $j = 0, 1, 2, ...$ (Step 2) of Algorithm 1. Since $K' \subseteq K = L_m(\mathbf{K})$, we have $K' \subseteq L_m(\mathbf{K}_0)$. Suppose now $K' \subseteq L_m(\mathbf{K}_j)$. Since K' is $\overline{C}$-observable for all P_i, no change will be made in the subsequent Step 3 by applying the algorithm in [1]. Therefore $K' \subseteq L_m(\mathbf{K}_{j+1})$, and eventually $K' \subseteq L_m(\mathbf{K}^{\uparrow})$. □

In practice we shall use Algorithm 1 as follows. Given a (specification) language $K \subseteq L_m(\mathbf{G})$, check if K is coobservable (polynomial algorithm available [16]). If so, we stop. Otherwise apply Algorithm 1 to obtain the supremal $\overline{K}$-coobservable sublanguage of K. Since relative coobservability implies coobservability, the obtained supremal sublanguage is also coobservable.

Now let us bring in control. Let $K \subseteq L_m(\mathbf{G})$ be a nonempty specification language. Since $\overline{C}$-coobservability, controllability, and $L_m(\mathbf{G})$-closedness are all closed under set union, there exists the supremal sublanguage of K that satisfies these three properties. Denote this supremal sublanguage by $K^{\uparrow}$; by Proposition 1 (or Proposition 2), $K^{\uparrow}$ is conjunctively (or disjunctively) coobservable, controllable, and $L_m(\mathbf{G})$-closed. Therefore, by [6] (resp. [8]) there exist decentralized supervisors conjunctively (resp. disjunctively) synthesizing $K^{\uparrow}$.

We present an algorithm to compute $K^{\uparrow}$. Let $\mathbf{G}$ and $\mathbf{K}$ be finite-state generators (as in (1)) with marked languages $L_m(\mathbf{G})$ and K, respectively.

Algorithm 2: Input $\mathbf{G}$, $\mathbf{K}$, and $P_i : \Sigma^* \to \Sigma_{o,i}^*$, $i \in \mathcal{I}$.
1. Set $\mathbf{K}_0 = \mathbf{K}$.
2. For $j \geq 0$, apply the algorithm in [17] with inputs $\mathbf{G}$ and $\mathbf{K}_j$ to obtain $\mathbf{H}_j$ such that $L_m(\mathbf{H}_j)$ is the supremal controllable and $L_m(\mathbf{G})$-closed sublanguage of $L_m(\mathbf{K}_j)$.
3. Apply Algorithm 1 with inputs $\mathbf{G}$, $\mathbf{H}_j$, $\mathbf{H}_j$, and $P_i : \Sigma^* \to \Sigma_o^*$ ($i \in \mathcal{I}$) to obtain $\mathbf{K}_{j+1}$ such that $L_m(\mathbf{K}_{j+1})$ is the supremal $L(\mathbf{H}_j)$-coobservable sublanguage of $L_m(\mathbf{H}_j)$. If $\mathbf{K}_{j+1} = \mathbf{K}_j$, then output $\mathbf{K}^{\uparrow} = \mathbf{K}_{j+1}$. Otherwise, advance j to $j + 1$ and go to Step 2.

Algorithm 2 terminates in finite steps, inasmuch as both algorithms used in Steps 2 and 3 do so and both remove states and/or transitions from the finite-state generator $\mathbf{K}$. The complexity of Algorithm 2 is exponential in the state size of $\mathbf{K}$, because Algorithm 1 is of this complexity.

Note that in applying Algorithm 1 in Step 3 above, the ambient language successively shrinks to the supremal controllable sublanguage $L(\mathbf{H}_j)$ computed at the immediately previous Step 2. Using successively smaller ambient languages helps generate less restrictive controlled behavior by discarding any strings outside $L(\mathbf{H}_j)$ that may be effectively prohibited by means of control.

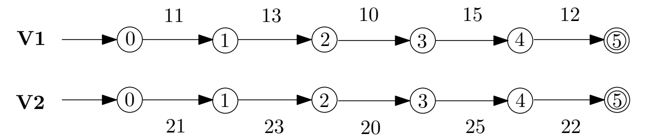


Fig. 5. Vehicle generator models

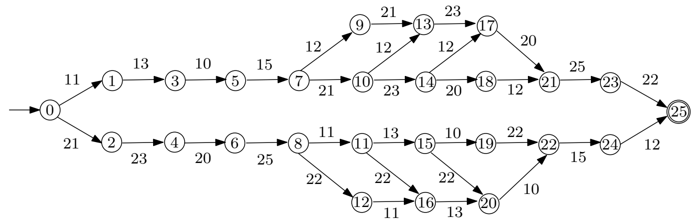


Fig. 6. Supremal conormal, controllable, and $L_m(\mathbf{G})$-closed sublanguage

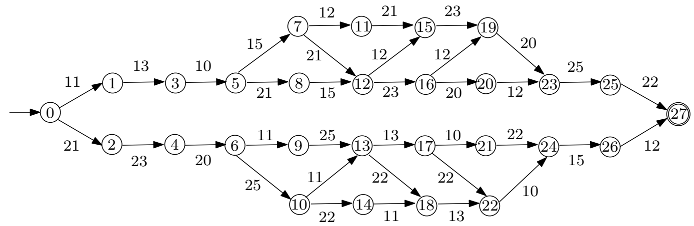


Fig. 7. Supremal relatively coobservable, controllable, and $L_m(\mathbf{G})$-closed sublanguage

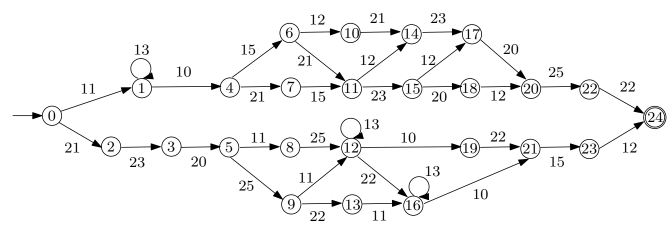


Fig. 8. Decentralized supervisor $\mathbf{SUP}_1$. The unobservable controllable event 13 is selflooped at those states where it is enabled.

IV. Guideway

We demonstrate relative coobservability and Algorithm 2 with a Guideway example, adapted from [5, Section 6.6]. As displayed in Fig. 5, two vehicles, $\mathbf{V}_1$ and $\mathbf{V}_2$, use the Guideway simultaneously and travel from station A (state 0) to B (state 5). The track between the two stations consists of 4 sections (states 1, 2, 3, 4). The plant $\mathbf{G}$ to be controlled is the synchronous product (e.g. [5]) $\mathbf{G} = \mathbf{V}_1 || \mathbf{V}_2$, and the control specification is to ensure that $\mathbf{V}_1$ and $\mathbf{V}_2$ never travel on the same section of track simultaneously, i.e. ensure *mutual exclusion* of the state pairs $(j, j), j = 1, ..., 4$. Let $\mathbf{K}$ be a generator representing this specification.

We consider the following decentralized supervisory control problem. Suppose that there are two supervisors, with unobservable event subsets $\Sigma_{uo,1} = \{13\}$, $\Sigma_{uo,2} = \{23\}$, and controllable event subsets $\Sigma_{c,1} = \{11, 13, 23, 15\}$, $\Sigma_{c,2} = \{21, 13, 23, 25\}$. The unobservable subsets $\Sigma_{uo,i}$ define the corresponding natural projections P_i, $i = 1, 2$, and the shared controllable events are 13, 23.

For comparison, we first compute the conormal, controllable, and $L_m(\mathbf{G})$-closed sublanguage, represented by the generator in Fig. 6. Then applying Algorithm 2, we obtain the generator in Fig. 7, which represents the supremal relatively coobservable, controllable, and $L_m(\mathbf{G})$-closed sublanguage. Observe that the relatively coobservable controlled behavior is strictly more permissive than the conormal counterpart. We next construct as in [6] the corresponding two decentralized supervisors $\mathbf{SUP}_i$, with $\Sigma_{uo,i}$ and $\Sigma_{c,i}$ $(i = 1, 2)$; $\mathbf{SUP}_1$ is displayed in Fig. 8 and $\mathbf{SUP}_2$ is similar.

We explain a representative case of the control logic of $\mathbf{SUP}_1$. If $\mathbf{SUP}_1$ observes that $\mathbf{V2}$ arrives at track section 3 (i.e. after string 21.23.20), either it allows $\mathbf{V1}$ to enter section 1 (i.e. $\mathbf{SUP}_1$ enables its private event 11), or $\mathbf{V2}$ is allowed by $\mathbf{SUP}_2$ to move onto section 4 (i.e. $\mathbf{SUP}_2$ enables its private event 25). When the former occurs, $\mathbf{SUP}_1$ must prevent $\mathbf{V1}$ from entering section 2 (i.e. $\mathbf{SUP}_1$ must disable the unobservable event 13 at its state 8) because otherwise $\mathbf{V1}$ can thereafter uncontrollably enter section 3 (event 10) and violate mutual exclusion at section 3. Note that since event 13 is shared, in the above case $\mathbf{SUP}_2$ must also disable 13. The above control action is not possible for conormality, since disabling unobservable events is not allowed. This is why relative coobservability achieves strictly more permissive than conormality does.

V. Conclusions

We have studied the new concept of relative coobservability in decentralized supervisory control of DES. We have proved that relative coobservability is stronger than (any variations of) coobservability, weaker than conormality, and closed under set union. Moreover, we have presented an algorithm for computing the supremal relatively coobservable (and controllable, $L_m(\mathbf{G})$-closed) sublanguage of a given language, and demonstrated the result with a Guideway example. In future work, we aim to apply relative coobservability in decentralized control of large systems and follow the architectural approach in [18].

References

[1] K. Cai, R. Zhang, and W. M. Wonham, "Relative observability of discrete-event systems and its supremal sublanguages," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 659–670, 2015.

[2] ——, "On relative observability of discrete-event systems," in *Proc. 52nd IEEE Conf. Decision and Control*, Florence, Italy, 2013, pp. 7285–7290.

[3] ——, "On relative observability of timed discrete-event systems," in *Proc. Workshop on Discrete-Event Systems*, Cachan, France, 2014, pp. 208–213.

[4] ——, "On relative coobservability of discrete-event systems," in *Proc. American Control Conference*, Chicago, IL, 2015, pp. 371–376.

[5] W. M. Wonham, "Supervisory Control of Discrete-Event Systems," Systems Control Group, ECE Dept, University of Toronto, updated July 1, 2015. Available online at http://www.control.toronto.edu/DES.

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

[7] R. Cieslak, C. Desclaux, A. S. Fawaz, and P. Varaiya, "Supervisory control of discrete-event processes with partial observations," *IEEE Trans. Autom. Control*, vol. 33, no. 3, pp. 249–260, 1988.

[8] T. S. Yoo and S. Lafortune, "A general architecture for decentralized supervisory control of discrete-event systems," *Discrete Event Dynamic Systems*, vol. 12, no. 3, pp. 335–377, 2002.

[9] ——, "Decentralized supervisory control with conditional decisions: supervisor existence," *IEEE Trans. Autom. Control*, vol. 49, no. 11, pp. 1886–1904, 2004.

[10] S. Tripakis, "Undecidable problems of decentralized observation and control on regular languages," *Information Processing Letters*, vol. 90, no. 1, pp. 21–28, 2004.

[11] S. Takai and T. Ushio, "A modified normality condition for decentralized supervisory control of discrete event systems," *Automatica*, vol. 38, no. 1, pp. 185–189, 2002.

[12] S. Takai, R. Kumar, and T. Ushio, "Characterization of co-observable languages and formulas for their super/sublanguages," *IEEE Trans. Autom. Control*, vol. 50, no. 4, pp. 434–447, 2005.

[13] P. Kozak and W. M. Wonham, "Fully decentralized solutions of supervisory control problems," *IEEE Trans. Autom. Control*, vol. 40, no. 12, pp. 2094–2097, 1995.

[14] K. Rohloff and S. Lafortune, "On the synthesis of safe control policies in decentralized control of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 1064–1068, 2003.

[15] F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Inform. Sci.*, vol. 44, no. 3, pp. 173–198, 1988.

[16] K. Rudie and J. C. Willems, "The computational complexity of decentralized discrete-event control problems," *IEEE Trans. Autom. Control*, vol. 40, no. 7, pp. 1313–1319, 1995.

[17] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM J. of Control and Optimization*, vol. 25, no. 3, pp. 637–659, 1987.

[18] L. Feng and W. M. Wonham, "Supervisory control architecture for discrete-event systems," *IEEE Trans. Autom. Control*, vol. 53, no. 6, pp. 1449–1461, 2008.