2015 American Control Conference
Palmer House Hilton
July 1-3, 2015. Chicago, IL, USA

On the Relationship between Codiagnosability and Coobservability under Dynamic Observations

Xiang Yin and Stéphane Lafortune

Abstract— **We investigate the relationship between the problem of decentralized fault diagnosis and the problem of decentralized control of discrete event systems under dynamic observations. The key system-theoretic properties that arise in these problems are those of codiagnosability and coobservability, respectively. It was shown by Wang *et al.* in [1] that coobservability is transformable to codiagnosability; however, the transformation for the other direction has remained an open problem. In this paper, we consider a general language-based dynamic observations setting and show how the notion of K-codiagnosability can be transformed to coobservability. Moreover, we show that, when the observation map is static, the standard notion of centralized diagnosability is transformable to observability. Our results thereby complement those in [1] and provide a better understanding of the relationship between the notions of codiagnosability and coobservability. In particular, our new results allow the leveraging of the large existing literature on decentralized control synthesis to solve problems of decentralized fault diagnosis.**

I. INTRODUCTION

Control and diagnosis are two important research areas in the study of Discrete Event Systems (DES). In complex automated systems, one is interested in designing a *supervisor* to restrict the system's behavior within a desired specification as well as designing a *diagnoser* in order to detect and isolate potential system's faults. Due to limited sensing capabilities, both problems involve dealing with partial observation of the system's behavior. Moreover, many technological systems have decentralized information structures, thereby necessitating the development of decentralized control and diagnosis architectures, where a set of supervisors or diagnosers work as a team to ensure the desired specifications.

The property of *observability* arose in the study of the control of partially observed DES [2], [3]. It is well known that observability together with controllability provide the necessary and sufficient conditions for the existence of a supervisor that achieves a given specification. In [4], this notion was extended to *coobservability* for decentralized control problems. Problems of centralized fault diagnosis of DES were initially studied in [5], [6] where the notion of *diagnosability* was introduced and characterized. Several future investigations ensued; see, e.g., the recent survey paper [7] for extensive bibliographies. Problems of decentralized fault diagnosis were considered in [8], where several communication protocols were developed. In particular, in Protocol 3 of [8], all the local agents work independently, i.e., there is no communication among them. This protocol was further investigated in several subsequent works and the associated condition of *codiagnosability* was characterized and studied; see, e.g., [9]–[11].

All of the above-mentioned works are concerned with the case of *static* observations, i.e., the set of observable events is fixed a priori. In many applications, communication among different agents (see, e.g., [12], [13]) as well as dynamic sensor activation (see, e.g., [14]–[16]) may lead to the case of *dynamic* observations. In the context of dynamic observations, the observability properties of an event are not fixed but may vary along each system trajectory. In [17], the authors studied the property of coobservability under dynamic observations. The fault diagnosis problem under dynamic observations has also been investigated in several works, such as [14], [15], [18] for the centralized case and [1] for the decentralized case.

There is a wide literature on the two properties of coobservability and codiagnosability, due to their importance in solving decentralized control and diagnosis problems, respectively. However, almost all of the existing literature deals with problems of control and problems of diagnosis *separately*. An exception of this is the work in [1], where it was shown, for the first time, how to map coobservability to codiagnosability, in the context of a language-based model for dynamic observations. This transformation from coobservability to codiagnosability makes it possible to leverage the large literature on methodologies to solve (decentralized) diagnosis problems to solve (decentralized) control problems. However, to the best of our knowledge, the reverse transformation, from codiagnosability to coobservability, has remained an open problem, as mentioned in the recent survey [19].

The contribution of this paper is to show, under a general language-based dynamic observations setting, how to transform *K-codiagnosability to coobservability*. K-codiagnosability is a strong version of codiagnosability where it is required that any failure be diagnosed within K steps after its occurrence; in codiagnosability, the detection delay has to be finite but no K is specified. The transformation that we present exploits the fact that both the problem of K-codiagnosability and the problem of coobservability can be reduced to a *state disambiguation problem*. Moreover, when the observation map is *event-based*, i.e., the observability properties of events are static, we show that the standard notion of diagnosability from [6] can be transformed to the standard notion of observability

This work was partially supported by NSF grants CCF-1138860 (Expeditions in Computing project ExCAPE: Expeditions in Computer Augmented Program Engineering), CNS-1446298, and CNS-1421122.

Xiang Yin and Stéphane Lafortune are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA. {xiangyin,stephane}@umich.edu.

from [2]. Our results thereby complement those in [1] and allow leveraging the large existing literature on problems of decentralized control to solve problems of decentralized fault diagnosis.

The remainder of this paper is organized as follows. Section II presents necessary preliminaries and in particular it reviews the notions of codiagnosability and coobservability. In Section III, the transformation from K-codiagnosability to coobservability under language-based observations is presented. The case of event-based observations is then considered in Section IV. We illustrate the application of the transformation algorithm of Section III to sensor activation problems in Section IV. Finally, we conclude the paper in Section VI. Due to space constraints, all proofs have been omitted and they are available in [20].

II. Preliminaries

A. System Model

We assume basic knowledge of DES and common notations (see, e.g., [21]). A DES is modeled as a deterministic finite-state automaton

$$G = (X^G, E^G, \delta^G, x_0^G) \tag{1}$$

where X^G is the finite set of states, E^G is the finite set of events, $\delta^G : X^G \times E^G \to X^G$ is the partial transition function where $\delta^G(x, e) = y$ means that there is a transition labelled by event e from state x to state y, and x_0 is the initial state. δ^G is extended to $X^G \times E^{G^*}$ in the usual way. The behavior generated by G is described by $\mathcal{L}(G) = \{s \in E^{G^*} : \delta^G(x_0, s)!\}$, where ! means is defined. The prefix-closure of a language L is $\overline{L} = \{s \in E^{G^*} : (\exists t \in E^{G^*})[st \in L]\}$. We use notation $|\cdot|$ to denote the length of a string.

In both control and diagnosis problems, there are some local agents monitoring the plant based on their own observations. Here, we assume that there are n local agents and we denote by $\mathcal{I} = \{1, \ldots, n\}$ the index set of the local agents. In most of the existing literature, the observation properties of events are specified by natural projection operations, i.e., for each agent $i \in \mathcal{I}$, the set of observable events $E_{o,i} \subseteq E^G$ is fixed a priori. We denote by $E_o = \cup_{i \in \mathcal{I}} E_{o,i}$ the total set of observable events. However, in many situations, the observable events may not be fixed. For instance, communication between agents may lead to different observability properties for the same event in different transitions. Also, under energy, bandwidth, or security constraints, a local agent may chose to enable/disable sensors *dynamically* based on its observation history. This also leads to dynamic observations. Thus, in a more general setting, we specify the observations of each agent $i \in \mathcal{I}$ by the mapping $\omega_i : \mathcal{L}(G) \to 2^{E_{o,i}}$. Given an observation mapping, $\omega_i, i \in \mathcal{I}$, we define the projection $P_{\omega_i} : \mathcal{L}(G) \to E_{o,i}^*$ recursively as follows:

$$P_{\omega_i}(\epsilon) = \epsilon, \quad P_{\omega_i}(s\sigma) = \begin{cases} P_{\omega_i}(s)\sigma & \text{if } \sigma \in \omega_i(s) \\ P_{\omega_i}(s) & \text{if } \sigma \notin \omega_i(s) \end{cases} \tag{2}$$

Clearly, if the set of observable events is fixed in the sense that $\forall s \in \mathcal{L}(G), \omega_i(s) = E_o$, then the projection P_{ω_i} reduces to the standard natural projection.

B. Control and Diagnosis under Dynamic Observations

In fault diagnosis problems, $E_F \subseteq E_{uo} := E^G \setminus E_o$ is the set of fault events whose occurrences must be detected by the diagnoser. In general, the set of fault events is partitioned into m disjoint sets, or *fault types*: $E_F = E_{F_1} \dot\cup \ldots \dot\cup E_{F_m}$; we denote by Π_F this partition and by $\mathcal{F} = \{1, \ldots, m\}$ the index set of the fault types. We define $\Psi(E_{F_k}) = \{sf \in \mathcal{L}(G) : f \in E_{F_k}\}$ to be the set of strings that end with a fault event of type F_k. We write $E_{F_k} \in s$, if $\overline{s} \cap \Psi(E_{F_k}) \neq \emptyset$. We say a language L is live if, for all $s \in L$, there exists an event $\sigma \in E$, such that $s\sigma \in L$. Hereafter, we assume that $\mathcal{L}(G)$ is live when [K-][co]diagnosability is considered. We denoted by L/s the post-language of L after s, i.e., $L/s = \{t \in E^{G^*} : st \in L\}$. In decentralized problems, in order to identify the fault event after its occurrence, it is required that the type of each such fault occurrence be unambiguously detected by one diagnoser within a finite number of steps (event occurrences) after the occurrence. We say that a language is K-codiagnosable if this diagnosis delay is uniformly bounded by a given number K. We say that a language is codiagnosable if there exists an integer K such that it is K-codiagnosable. The formal definition of [K-]codiagnosability under dynamic observations is recalled from [1].

Definition 1: (Codiagnosability). A live language $\mathcal{L}(G)$ is said to be K-codiagnosable w.r.t. $\omega_i, i \in \mathcal{I}$ and Π_F on E_F if

$$(\forall k \in \mathcal{F})(\forall s \in \Psi(E_{F_k}))(\forall t \in \mathcal{L}(G)/s)[|t| \geq K \Rightarrow CD] \tag{3}$$

where the codiagnosability condition CD is

$$(\exists i \in \mathcal{I})(\forall w \in \mathcal{L}(G))[P_{\omega_i}(w) = P_{\omega_i}(st) \Rightarrow E_{F_k} \in w]. \tag{4}$$

We say that $\mathcal{L}(G)$ is codiagnosable if there exists an integer $K \in \mathbb{N}$ such that it is K-codiagnosable. ■

Remark 2.1: The above definition of codiagnosability is equivalent to the one in [8]–[10] in the case of regular languages, as assumed in this paper. Specifically, the definition in [8]–[10] states that for all faulty strings, there is a finite detection delay. The above definition reverses the two quantifiers as it states that there is a detection delay that works for all faulty strings. However, it was shown in [22] that in the case of regular languages, the two definitions are equivalent in the centralized case for static observation mappings. The result in [22] can be extended to the decentralized case and to language-based observation mappings, although the proof is omitted here.

In decentralized supervisor control problems, each local agent not only monitors the plant, but it can also dynamically disable/enable events to actively control the plant based on its observations. Formally, for each agent $i \in \mathcal{I}$, we denote by $E_{c,i} \subseteq E$ its set of controllable events. A local supervisor is a mapping $S_i : E_{o,i}^* \to \Gamma_i$, where $\Gamma_i := \{\gamma \in 2^E : E \setminus E_{c,i} \subseteq \gamma\}$ and $\wedge_{i \in \mathcal{I}} S_i/G$ denotes the controlled system under the conjunctive fusion rule for enabled events. The legal behavior to be achieved under control is specified by a prefix-closed (regular) language $\mathcal{L}(H) \subseteq \mathcal{L}(G)$, where $H = (X^H, E^H, \delta^H, x_0^H)$ is the automaton that generates the specification language. It is well known that coobservability together with controllability provide the necessary and sufficient conditions for the existence of a set of decentralized supervisors that together achieve a given language. Formally, we recall the definition of coobservability under dynamic observations from [1], [17].

Definition 2: (Coobservability). A language $\mathcal{L}(H) \subseteq \mathcal{L}(G)$ is said to be coobservable w.r.t. $\mathcal{L}(G), \omega_i$ and $E_{c,i}, i \in \mathcal{I}$ if for all $s \in \mathcal{L}(H)$ and for all $\sigma \in E_c := \cup_{i\in\mathcal{I}} E_{c,i}$,

$$(s\sigma \in \mathcal{L}(G) \setminus \mathcal{L}(H)) \Rightarrow (\exists i \in I^c(\sigma))[P_{\omega_i}^{-1}(P_{\omega_i}(s))\sigma \cap \mathcal{L}(H) = \emptyset] \quad (5)$$

where, $I^c(\sigma) := \{i \in \mathcal{I} : \sigma \in E_{c,i}\}$. ■

Note that in both Definitions 1 and 2, codiagnosability and coobsevability are defined in the most general manner, i.e., we consider the case where there are multiple agents under language-based dynamic observations. For the sake of brevity, we also use the following terminologies hereafter. We refer to [K-]codiagnosability as [K-]diagnosability in the centralized case, i.e., when $|\mathcal{I}| = 1$; similarly for observability. Moreover, we say the system is *static* [K-][co]diagnosable or [K-][co]observable if the observation mappings are specified by natural projections.

III. From K-Codiagnosability to Coobservability

In this section, we present an algorithm to transform the problem of K-codiagnosability to the problem of coobservability under general language-based dynamic observations.

The definition of codiagnosability requires that every occurrence of the fault events be diagnosed within a finite delay, without specifying a bound for that delay. In contrast, K-codiagnosability explicitly specifies a uniform detection delay bound for all fault event occurrences. Hence, K-codiagnosability is a stronger property than codiagnosability in the sense that K-codiagnosability implies codiagnosability, but the reverse may not hold for some values of K.

First, we show that the notion of K-codiagnosability can be transformed to coobservability when there is only one type of fault events. We shall need the notation $A \sqsubseteq B$ to denote that automaton A is a *sub-automaton* of automaton B, as defined in [21] (p. 86). Let $H = (X^H, E^H, \delta^H, x_0^H)$ be the automaton to be diagnosed with fault events and $E_{F_k}, k \in \mathcal{F}$ be the set of fault events under consideration (i.e., only type k faults are to be diagnosed). We construct two automata $\tilde{H}_k = (X^{\tilde{H}_k}, E^{\tilde{H}_k}, \delta^{\tilde{H}_k}, x_0^{\tilde{H}_k})$ and $\tilde{G}_k = (X^{\tilde{G}_k}, E^{\tilde{G}_k}, \delta^{\tilde{G}_k}, x_0^{\tilde{G}_k})$ with $\tilde{H}_k \sqsubseteq \tilde{G}_k$, as follows.

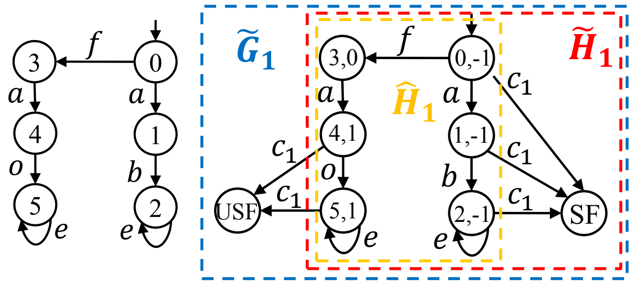
Algorithm KCOD-COOB-I
Input: $H = (X^H, E^H, \delta^H, x_0^H), E_{F_k}$ and K.
Output: $\tilde{H}_k = (X^{\tilde{H}_k}, E^{\tilde{H}_k}, \delta^{\tilde{H}_k}, x_0^{\tilde{H}_k})$ and $\tilde{G}_k = (X^{\tilde{G}_k}, E^{\tilde{G}_k}, \delta^{\tilde{G}_k}, x_0^{\tilde{G}_k})$.

Step 1: Build a new automaton $\hat{H}_k = (X^{\hat{H}_k}, E^{\hat{H}_k}, \delta^{\hat{H}_k}, x_0^{\hat{H}_k})$, where
$X^{\hat{H}_k} \subseteq X^H \times \{-1, 0, 1, \ldots, K\}$ is the set of states;
$E^{\hat{H}_k} = E^H$ is the set of events;
$\delta^{\hat{H}_k} : X^{\hat{H}_k} \times E^{\hat{H}_k} \to X^{\hat{H}_k}$ is the partial transition function where for any $\hat{x} = (x, n) \in X^{\hat{H}_k}$, $\delta^{\hat{H}_k}$ is defined by

$$\delta^{\hat{H}_k}(\hat{x}, \sigma) = \begin{cases} (\delta^H(x,\sigma), -1), & \\ \quad \text{if} \begin{pmatrix} n = -1 \text{ and} \\ \sigma \in E \setminus E_{F_k} \end{pmatrix} & \\ (\delta^H(x,\sigma), n+1), & \\ \quad \text{if} \begin{pmatrix} 0 \le n < K \text{ or} \\ n = -1 \wedge \sigma \in E_{F_k} \end{pmatrix} & \\ (\delta^H(x,\sigma), K), & \text{if } n = K \end{cases} \quad (6)$$

$x_0^{\hat{H}_k} = (x_0^H, -1)$ is the initial state.

Step 2: Set $\tilde{H}_k \leftarrow \hat{H}_k$. Add state $X^{\tilde{H}_k} \leftarrow X^{\tilde{H}_k} \cup \{SF\}$ and add event $E^{\tilde{H}_k} \leftarrow E^{\hat{H}_k} \cup \{c_k\}$.
Step 3: For all $\hat{x} = (x, n) \in X^{\hat{H}_k}$ in $\tilde{H}_k$, if $n = -1$, then add new transition $\delta^{\tilde{H}_k}(\hat{x}, c_k) = SF$.
Step 4: Set $\tilde{G}_k \leftarrow \tilde{H}_k$. Add state $X^{\tilde{G}_k} \leftarrow X^{\tilde{G}_k} \cup \{USF\}$.
Step 5: For all $\hat{x} = (x, n) \in X^{\hat{H}_k}$ in $\tilde{G}_k$, if $n = K$, then add new transition $\delta^{\tilde{G}_k}(\tilde{x}, c_k) = USF$.
Step 6: For all $i \in \mathcal{I}$, the observation mapping $\omega_{i,\tilde{G}_k}$ for $\tilde{G}_k$ is specified as follows. For all $s \in \mathcal{L}(\hat{H}_k)$, $\omega_{i,\tilde{G}_k}(s) \leftarrow \omega_i(s)$. For all $s = tc_k \in \mathcal{L}(\tilde{G}_k) \setminus \mathcal{L}(\hat{H}_k)$, $\omega_{i,\tilde{G}_k}(s) \leftarrow \omega_i(t)$.
Step 7: For all $i \in \mathcal{I}$, $E_{c,i} \leftarrow \{c_k\}$.



(a) H (b) $K = 1$

Fig. 1. $\omega(s) = \{o, e\}, \forall s \in \mathcal{L}(H)$

Example 3.1: Consider the centralized static diagnosis problem instance shown in Figure 1(a). H is the automaton to be diagnosed with fault events, where $E_{F_1} = \{f\}$ is the set of fault events and $E_o = \{o, e\}$ is the set of observable events. The observation mapping ω is given by $\forall s \in \mathcal{L}(G), \omega(s) = \{o, e\}$. When the desired diagnosis delay is set to $K = 1$, by applying Algorithm KCOD-COOB-I, the corresponding $\tilde{G}_1$ and $\tilde{H}_1$ can be constructed, as shown in Figure 1(b). The observation mapping is also given by $\forall s \in \mathcal{L}(\tilde{G}_1), \omega_{\tilde{G}_1}(s) = \{o, e\}$. It is clear that $\mathcal{L}(\tilde{H}_1)$ is not observable w.r.t. $\mathcal{L}(\tilde{G}_1)$, $\omega_{\tilde{G}_1}$ and $\{c_k\}$, since for strings fa and a with $P(fa) = P(a)$, we have $fac_1 \in \mathcal{L}(\tilde{G}_1) \setminus \mathcal{L}(\tilde{H}_1)$ but $ac_1 \in \mathcal{L}(\tilde{H}_1)$. Also, the original system H is not 1-diagnosable. However, it can be verified that H is 2-diagnosable. The relationship between H, $\tilde{H}_k$ and $\tilde{G}_k$ will be formally described in Theorem 1 below. ■

The following theorem establishes that the above construction procedure transforms the problem of K-codiagnosability to the problem of coobservability for each type of fault events.

Theorem 1: Language $\mathcal{L}(H)$ is K-coodiagnosable w.r.t. $\omega_i, i \in \mathcal{I}$ and fault event set E_{F_k}, if and only if, $\mathcal{L}(\tilde{H}_k)$ is coobservable w.r.t. $\mathcal{L}(\tilde{G}_k)$, $\omega_{i,\tilde{G}_k}$ and $E_{c,i}, i \in \mathcal{I}$.

The intuition behind the construction procedure in Algorithm KCOD-COOB-I is as follows. The idea of the transformation is based on the fact that both the problem of K-codiagnosability and the problem of coobservability can be reduced to the problem of *state disambiguation*; see, e.g., [23]. Clearly, we see that $\hat{H}_k$ is a finite unfolding of H and they are language equivalent, i.e., $\mathcal{L}(H) = \mathcal{L}(\hat{H}_k)$. Let us define the set of *conflicting* states pairs

$$T_{conf} := \{(u,v) \in X^{\hat{H}_k} \times X^{\hat{H}_k} : [u]_n = -1 \text{ and } [v]_n = K\}$$

where $[u]_n$ denotes the integer component of u. If $\mathcal{L}(\hat{H}_k)$ is K-codiagnosable, then for any state pair in the set T_{conf}, at least one agent should be able to distinguish the states in it. In the context of supervisory control, by construction, to achieve specification $\mathcal{L}(\tilde{H}_k)$ for plant $\mathcal{L}(\tilde{G}_k)$, we always need to enable c_k at states labeled with integer -1 and disable c_k at states labeled with integer K. Thus we also need to distinguish the states of any state pair in T_{conf}; otherwise, we will not be able to know whether or not we need to disable c_k. The correctness proof of the transformation algorithm follows immediately from these results.

The next result gives the worst-case complexity of Algorithm KCOD-COOB-I.

Theorem 2: Let H be the automaton to be diagnosed with fault events. Then the worst-case time complexity of Algorithm KCOD-COOB-I is $O(K|X^H||E^H|)$.

So far, we have shown that for each individual type of fault, the problem of K-codiagnosability can be transformed to the problem of coobservability. However, our objective is to show that the problem of K-codiagnosability with *multiple* fault types is transformable to the problem of coobservability. For this purpose, we need to transform the problem of K-codiagnosability to the problem of coobservability in a *single* automaton. This is achieved by Algorithm KCOD-COOB-II presented next. The notation $A \parallel B$ denotes the usual *parallel composition* operation of automata A and B (see, e.g., [21]).

Algorithm KCOD-COOB-II
Input: $H = (X^H, E^H, \delta^H, x_0^H), E_F, \Pi_F$ and K.
Output: $\tilde{H} = (X^{\tilde{H}}, E^{\tilde{H}}, \delta^{\tilde{H}}, x_0^{\tilde{H}})$
and $\tilde{G} = (X^{\tilde{G}}, E^{\tilde{G}}, \delta^{\tilde{G}}, x_0^{\tilde{G}})$.

Step 1: For each type of fault $k \in \mathcal{F}$, build an automaton $\hat{H}_k$, as described in Step 1 of Algorithm KCOD-COOB-I.

Step 2: Set $\hat{H} \leftarrow \hat{H}_1 \parallel \hat{H}_2 \parallel \cdots \parallel \hat{H}_{|\mathcal{F}|}$.

Step 3: Set $\tilde{H} \leftarrow \hat{H}$. Add state $X^{\tilde{H}} \leftarrow X^{\tilde{H}} \cup \{SF\}$ and add event $E^{\tilde{H}} \leftarrow E^{\tilde{H}} \cup \{c_k : k \in \mathcal{F}\}$.

Step 4: For all $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_{|\mathcal{F}|}) \in X^{\hat{H}}$ in $\tilde{H}$, for all $k \in \mathcal{F}$, if $[\hat{x}_k]_n = -1$, then add new transition $\delta^{\tilde{H}}(\hat{x}, c_k) = SF$.

Step 5: Set $\tilde{G} \leftarrow \tilde{H}$. Add state $X^{\tilde{G}} \leftarrow X^{\tilde{G}} \cup \{USF\}$.

Step 6: For all $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_{|\mathcal{F}|}) \in X^{\hat{H}}$ in $\tilde{G}$, for all $k \in \mathcal{F}$, if $[\hat{x}_k]_n = K$, then add new transition $\delta^{\hat{G}}(\tilde{x}, c_k) = USF$.

Step 7: For all $i \in \mathcal{I}$, the observation mapping $\omega_{i,\tilde{G}}$ for $\tilde{G}$ is specified as follows. For all $s \in \mathcal{L}(\hat{H})$, $\omega_{i,\tilde{G}}(s) \leftarrow \omega_i(s)$. For all $tc \in \mathcal{L}(\tilde{G}) \setminus \mathcal{L}(\hat{H})$, where $c \in \{c_k : k \in \mathcal{F}\}$, $\omega_{i,\tilde{G}}(s) \leftarrow \omega_i(t)$.

Step 8: For all $i \in \mathcal{I}$, $E_{c,i} \leftarrow \{c_k : k \in \mathcal{F}\}$.

Remark 3.1: Algorithm KCOD-COOB-II essentially *merges* all automata $\hat{H}_k$, $k \in \mathcal{F}$, constructed by Algorithm KCOD-COOB-I into a single automaton $\hat{H}$. Then single copies of the new states s and us are added. Note that, in Step 2 of Algorithm KCOD-COOB-II, the parallel composition between $\hat{H}_k, k \in \mathcal{F}$, could have resulted in an automaton with $\prod_{k\in\mathcal{F}} |X^{\hat{H}_k}|$ number of states in general. However, since for any $i \in \mathcal{F}$, $\hat{H}_i$ is a finite unfolding of H, then for any state $((x_1,n_1),(x_2,n_2),\ldots,(x_m,n_m)) \in X^{\hat{H}}$, we have that $x_1 = x_2 = \cdots = x_m$. The number of states in the composed system is only exponential in K, i.e., $|X^{\hat{H}}| \le K^m |X^H|$.

The following results show the properties and the correctness of the transformation in Algorithm KCOD-COOB.

Lemma 3.1: The following four statements are equivalent.

S1 $\mathcal{L}(H)$ is K-coodiagnosable w.r.t. $\omega_i, i \in \mathcal{I}$ and the fault event set E_F with partition Π_F.

S2 For any $k \in \mathcal{F}$, $\mathcal{L}(H)$ is K-coodiagnosable w.r.t. $\omega_i, i \in \mathcal{I}$ and the fault event set E_{F_k}.

S3 For any $k \in \mathcal{F}$, $\mathcal{L}(\tilde{H}_k)$ is coobservable w.r.t. $\mathcal{L}(\tilde{G}_k)$, $\omega_{i,\tilde{G}_k}$ and $E_{c,i} = \{c_k\}, \forall i \in \mathcal{I}$.

S4 $\mathcal{L}(\tilde{H})$ is coobservable w.r.t. $\mathcal{L}(\tilde{G})$, $\omega_{i,\tilde{G}}$ and $E_{c,i} = \{c_k : k \in \mathcal{F}\}, i \in \mathcal{I}$.

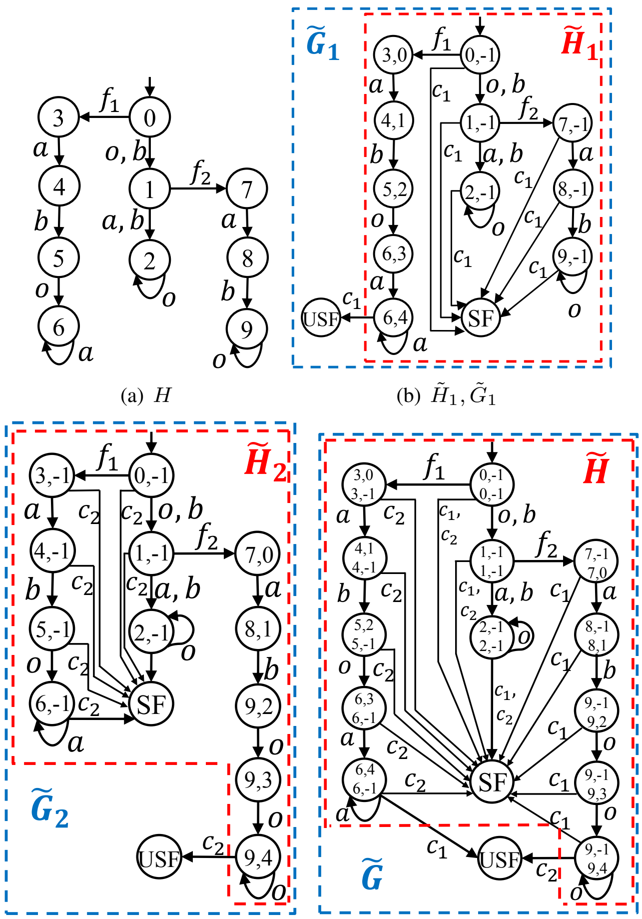
Follows from Lemma 3.1, we have the following theorems.

Theorem 3: Language $\mathcal{L}(H)$ is K-coodiagnosable w.r.t. $\omega_i, i \in \mathcal{I}$ and Π_F on E_F, if and only if, $\mathcal{L}(\tilde{H})$ is coobservable w.r.t. $\mathcal{L}(\tilde{G})$, $\omega_{i,\tilde{G}}$ and $E_{c,i}, i \in \mathcal{I}$.

Theorem 4: Let H the automaton to be diagnosed with fault events. Then the worst-case time complexity of Algorithm KCOD-COOB-II is $O(K^m|X^H||E^H|)$.

Example 3.2: Let the automaton H shown in Figure 2(a) be the system to be diagnosed with fault events, where $E_{F_1} = \{f_1\}$ and $E_{F_2} = \{f_2\}$ are two types of fault events. When $K = 4$, by applying Algorithm KCOD-COOB-I and Algorithm KCOD-COOB-II, the corresponding $\tilde{H}_1, \tilde{G}_1, \tilde{H}_2, \tilde{G}_2, \tilde{H}$ and $\tilde{G}$ that are obtained are shown in Figures 2(b)-2(d).

Suppose that the observation mapping for H is static, i.e., the sets of observable for agents 1 and 2 are constant and given by $E_{o,1} = \{a, o\}$ and $E_{o,2} = \{b, o\}$, respectively. The transformed observation mappings for $\tilde{G}$ are also specified by natural projections $P_1 : \mathcal{L}(\tilde{G}) \to E_{o,1}^*$ and $P_2 : \mathcal{L}(\tilde{G}) \to E_{o,2}^*$. Consider strings $s = of_2aboo \in \mathcal{L}(\tilde{G})$ and controllable event $c_2 \in E^{\tilde{G}}$ such that $sc_2 \in \mathcal{L}(\tilde{G}) \setminus \mathcal{L}(\tilde{H})$. For agent 1, there exists string $s_1 = oaoo$ such that $P_1(s) = P_1(s_1)$ and $s_1c_2 \in \mathcal{L}(\tilde{H})$; and for agent two, there exists string



(a) H (b) $\tilde{H}_1, \tilde{G}_1$

(c) $\tilde{H}_2, \tilde{G}_2$ (d) $\tilde{H}, \tilde{G}$

Fig. 2. $K = 4$ and $E_{o,1} = \{a, o\}$, $E_{o,2} = \{b, o\}$

$s_2 = oboo$ such that $P_2(s) = P_2(s_1)$ and $s_2c_2 \in \mathcal{L}(\tilde{H})$. By Definition 2, we conclude that $\mathcal{L}(\tilde{H})$ is not coobservable w.r.t. $\mathcal{L}(\tilde{G})$, P_1, P_2 and $E_{c,1} = E_{c,2} = \{c_1, c_2\}$. Consequently, the original system H is not 4-codiagnosable. ■

IV. Case of Event-Based Observation

In this section, we show that in the case of event-based observations, i.e., static observability properties of events, the transformation results in Section III can be extended from the notion of K-[co]diagnosability to the stronger notion of [co]diagnosability.

Recall that, in the transformation algorithm in Section III, the desired diagnosis delay K is specified a priori and the observation is language-based. Let us eliminate that extra level of generality and assume that, for each agent $i \in \mathcal{I}$, the set of observable events $E_{o,i} \subseteq E$ is fixed a priori. In this case, it is possible to relax the pre-information on K and extend the transformation algorithm of Section III from K-diagnosability to diagnosability. We now explain how to proceed.

In [24], the authors show that for the centralized static diagnosis problem, if H is diagnosable, then any fault occurrence will be detected within $|X^H|^2$ transitions after the fault event occurs. Such an upper bound of the diagnosis delay is derived from the size of *verifier*, a special type of automaton used for verifying diagnosibility. Therefore, we can simply use the upper bound $|X^H|^2$ to replace the integer K in Theorem 4 and we obtain the following result.

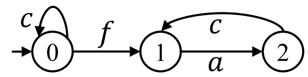


Fig. 3. Automaton H for Example 4.1

Proposition 4.1: When the observations are event-based, diagnosability can be transformed to observability in $O(|X^H|^{2m+1}|E^H|)$.

In [9] and [10], the verifier technique was extended to the decentralized case; the decentralized verifier has size proportional to $|X|^{n+1}$. By using the same argument as above, we have the following result.

Proposition 4.2: When the observations are event-based, co-diagnosability can be transformed to coobservability in $O(|X^H|^{mn+m+1}|E^H|^2)$.

The question that arises is the following: In the dynamic decentralized diagnosis problem, can we also find such an upper bound to replace K, where this upper bound would work for any language-based mapping? In general, such an upper bound does not exist, since the observation policy is language-based and K could be arbitrary large. This phenomenon is illustrated by the following example.

Example 4.1: Consider the automaton H in Figure 3, where f is the unique fault event. Consider the information mapping $\omega : \mathcal{L}(G) \to 2^{E_o}$ defined by:

$$\omega(s) = \begin{cases} \{c\}, & \text{if } s \in \overline{\{f(ac)^n, c^n\}} \\ \{a, c\}, & \text{if } s \in \mathcal{L}(G) \setminus \overline{\{f(ac)^n, c^n\}} \end{cases} \quad (7)$$

where n is an arbitrary non-negative integer. Since we are unable to distinguish strings c^m and $f(ac)^m$ until the first time we observe event a, which does not occur until $m = n+1$, we see that under information mapping ω, the system is $(2n+3)$-diagnosable but not $(2n+2)$-diagnosable. Since n can be arbitrary large, there is no general upper bound for the diagnosis delay under language-based observations. ■

V. Application to Optimization of Sensor Activation

In this section, we show how the transformation algorithm from Section III can leverage the research on observability and coobservability to solve problems related to diagnosability and codiagnosability.

In sensor activation problems, the sensors can be turned on/off on-line by the agents based on their observation histories. In this scenario, one is interested in synthesizing a *sensor activation policy* that achieves certain observation properties; see, e.g., [25], [26]. Roughly speaking, sensor activation policies are a particular class of information mappings satisfying the property that the sensor activations for any two indistinguishable strings must be the same. This property is called the *feasibility* condition in [25], [26]. It is formally defined as follows.

Definition 3: Given a system G, a set of observation mappings $\omega_i : \mathcal{L}(G) \to 2^{E_{o,i}}$ is said to be a feasible sensor activation policy if

$$(\forall s, t \in \mathcal{L}(G))[P_{\omega_i}(s) = P_{\omega_i}(t) \Rightarrow \omega_i(s) = \omega_i(t)]$$

■

The following theorem reveals that feasibility is preserved under the transformation algorithm of Section III. In other words, any sensor activation policy synthesized for the transformed system can be applied back to the original system.

Theorem 5: Let H be the original system and $\tilde{G}$ be the transformed system. Then, ω_i is a feasible sensor activation policy for H if and only if $\omega_{i,\tilde{G}}$ is a feasible sensor activation policy for $\tilde{G}$.

Unlike the direct approach investigated in [14], [15] for K-diagnosability, the above theorem provides an alternative approach for the synthesis of optimal sensor activation policies for K-codiagnosability. Suppose H is the system to be diagnosed with fault events; $\tilde{H}$ and $\tilde{G}$ are the transformed systems. We can then apply the algorithm in [25] to obtain the optimal sensor activation policy $\omega_{i,\tilde{G}}$ ensuring coobservability for the transformed systems $\tilde{H}$ and $\tilde{G}$. Then an optimal sensor activation policy ω_i for K-codiagnosability can be calculated by setting $\omega_i(s) = \omega_{i,\tilde{G}}(s)$ for all $s \in \mathcal{L}(H)$.

Note that the works in [14], [15] only consider the centralized sensor activation problem and that the algorithm developed in [26] is for codiagnosability, not for K-codiagnosability. To the best of our knowledge, the problem of synthesizing an optimal sensor activation policy for K-codiagnosability had remained an open problem. It can now be solved by applying the transformation algorithm of Section III together with the algorithm in [25].

VI. Conclusion

In this paper, we have presented a new transformation algorithm that shows that the property of language-based K-codiagnosability can be transformed to the property of language-based coobservability, where the integer K is given a priori. Language-based properties are those where the observability properties of an event are dynamic, i.e., are history-dependent. These results complement those in [1] that pertain to the reverse transformation, from coobservability to codiagnosability. Moreover, we have shown that, when the observation properties are static, referred to as the event-based case, (static) [co]diagnosability is transformable to (static) [co]observability. These new results allow the leveraging of the large existing literature on solution methodologies for problems of decentralized control to be applied to solve problems of decentralized fault diagnosis. When the desired diagnosis delay K is not given, the transformation from language-based codiagnosability to language-based coobervability still remains an open problem.

References

[1] W. Wang, A. R. Girard, S. Lafortune, and F. Lin, "On codiagnosability and coobservability with dynamic observations," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1551–1566, 2011.

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

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

[4] K. Rudie and W. M. Wonham, "Think globally, act locally: Decentralized supervisory control," *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1692–1708, 1992.

[5] F. Lin, "Diagnosability of discrete event systems and its applications," *Discrete Event Dynamic Systems: Theory & Appl.*, vol. 4, no. 2, pp. 197–212, 1994.

[6] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, 1995.

[7] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for discrete event systems," *Annual Reviews in Control*, vol. 37, no. 2, pp. 308–320, 2013.

[8] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discrete Event Dyn. Sys.: Theory & Appl.*, vol. 10, no. 1-2, pp. 33–86, 2000.

[9] W. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," *IEEE Trans. Systems, Man and Cybernetics, Part A*, vol. 36, no. 2, pp. 384–395, 2006.

[10] Y. Wang, T.-S. Yoo, and S. Lafortune, "Diagnosis of discrete event systems using decentralized architectures," *Discrete Event Dynamic Systems: Theory & Appl.*, vol. 17, no. 2, pp. 233–263, 2007.

[11] M. V. Moreira, T. C. Jesus, and J. C. Basilio, "Polynomial time verification of decentralized diagnosability of discrete event systems," *IEEE Trans. Automatic Control*, vol. 56, no. 7, pp. 1679–1684, 2011.

[12] K. Rudie, S. Lafortune, and F. Lin, "Minimal communication in a distributed discrete-event system," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 957–975, 2003.

[13] F. Lin, "Control of networked discrete event systems: dealing with communication delays and losses," *SIAM Journal on Control and Optimization*, vol. 52, no. 2, pp. 1276–1298, 2014.

[14] D. Thorsley and D. Teneketzis, "Active acquisition of information for diagnosis and supervisory control of discrete event systems," *Discrete Event Dyn. Sys.: Theory & Appl.*, vol. 17, no. 4, pp. 531–583, 2007.

[15] F. Cassez and S. Tripakis, "Fault diagnosis with static and dynamic observers," *Fund. Informaticae*, vol. 88, no. 4, pp. 497–540, 2008.

[16] D. Sears and K. Rudie, "Efficient computation of sensor activation decisions in discrete-event systems," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 6966–6971.

[17] Y. Huang, K. Rudie, and F. Lin, "Decentralized control of discrete-event systems when supervisors observe particular event occurrences," *IEEE Trans. Automatic Control*, vol. 53, no. 1, pp. 384–388, 2008.

[18] E. Dallal and S. Lafortune, "On most permissive observers in dynamic sensor activation problems," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 966–981, 2014.

[19] D. Sears and K. Rudie, "Sensor activation and communication problems in discrete-event systems," *Tech. Rep., Queens University*, 2013.

[20] X. Yin and S. Lafortune, "Codiagnosability and coobservability under dynamic observations: Transformation and verification," *University of Michigan, Tech. Rep.*, Dec, 2014.

[21] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. Springer, 2008.

[22] T.-S. Yoo and H. E. Garcia, "Event counting of partially-observed discrete-event systems with uniformly and nonuniformly bounded diagnosis delays," *Discrete Event Dynamic Systems: Theory & Appl.*, vol. 19, no. 2, pp. 167–187, 2009.

[23] W. Wang, S. Lafortune, and F. Lin, "An algorithm for calculating indistinguishable states and clusters in finite-state automata with partially observable transitions," *Systems & Control Letters*, vol. 56, no. 9, pp. 656–661, 2007.

[24] T.-S. Yoo and S. Lafortune, "Polynomial-time verification of diagnosability of partially observed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 9, pp. 1491–1495, 2002.

[25] W. Wang, S. Lafortune, F. Lin, and A. R. Girard, "Minimization of dynamic sensor activation in discrete event systems for the purpose of control," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2447–2461, 2010.

[26] W. Wang, S. Lafortune, A. R. Girard, and F. Lin, "Optimal sensor activation for diagnosing discrete event systems," *Automatica*, vol. 46, no. 7, pp. 1165–1175, 2010.