Automatica 61 (2015) 241–252

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Codiagnosability and coobservability under dynamic observations: Transformation and verification[☆]

Xiang Yin [1], Stéphane Lafortune

Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA

ARTICLE INFO

Article history:
Received 9 December 2014
Received in revised form
30 April 2015
Accepted 7 August 2015

Keywords:
Discrete-event systems
Fault diagnosis
Supervisory control
Diagnosability
Observability

ABSTRACT

We investigate the relationship between decentralized fault diagnosis and 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. (2011) 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. When the observation properties are transition-based, we present a new approach for the verification of transition-based codiagnosability. An upper bound of the diagnosis delay for decentralized diagnosis under transition-based observations is derived. Moreover, we show that transition-based [co]diagnosability is transformable to transition-based [co]observability. Our results thereby complement those in Wang et al. (2011) and provide a thorough characterization of the relationship between the two notions of codiagnosability and coobservability and their verification. In particular, our results allow the leveraging of the large existing literature on decentralized control synthesis to solve corresponding problems of decentralized fault diagnosis.

1. 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 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 (Cieslak, Desclaux, Fawaz, & Varaiya, 1988; Lin & Wonham, 1988). 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. This notion was extended to *coobservability* for decentralized control problems, see, e.g., Overkamp and van Schuppen (2000), Rudie and Willems (1995), Rudie and Wonham (1992), Seatzu, Silva, and Van Schuppen (2013), Tripakis (2004) and Yoo and Lafortune (2002). Problems of centralized fault diagnosis of DES were initially studied in Lin (1994) and Sampath, Sengupta, Lafortune, Sinnamohideen, and Teneketzis (1995) where the notion of *diagnosability* was introduced and characterized. Several future investigations ensued and a large amount of literature has been published on this topic; the recent survey papers (Zaytoon & Lafortune, 2013; Zaytoon & Sayed-Mouchaweh, 2012) contain extensive bibliographies. Problems of decentralized fault diagnosis were initially considered in Debouk, Lafortune, and Teneketzis (2000), where several communication protocols were developed. In particular, in Protocol 3 of Debouk et al. (2000), 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., Moreira, Jesus, and Basilio (2011), Qiu and Kumar (2006)

[☆] This work was partially supported by the US National Science Foundation grants CCF-1138860 (Expeditions in Computing project ExCAPE: Expeditions in Computer Augmented Program Engineering), CNS-1446298, and CNS-1421122. The material in this paper was partially presented at the 2015 American Control Conference, July 1–3, 2015, Chicago, IL, USA. This paper was recommended for publication in revised form by Associate Editor Jan Komenda under the direction of Editor Ian R. Petersen.

E-mail addresses: xiangyin@umich.edu (X. Yin), stephane@umich.edu (S. Lafortune).

[1] Tel.: +1 7348343243; fax: +1 7347638041.

http://dx.doi.org/10.1016/j.automatica.2015.08.023
0005-1098/© 2015 Elsevier Ltd. All rights reserved.

and Wang, Yoo, and Lafortune (2007). State-based, distributed, and robust approaches to diagnosis have also been considered; see, e.g., Carvalho, Basilio, and Moreira (2012), Hashtrudi Zad, Kwong, and Wonham (2003), Pencolé and Cordier (2005), Seatzu et al. (2013), Su and Wonham (2005) and Zaytoon and Lafortune (2013).

All of the above-mentioned works are concerned with the case of *static* observations, where the set of observable events is fixed a priori. In many applications however, communication among different agents (see, e.g., Lin, 2014, Rudie, Lafortune, & Lin, 2003) as well as dynamic sensor activation (see, e.g., Cassez & Tripakis, 2008, Sears & Rudie, 2013a,b, Thorsley & Teneketzis, 2007, Wang, Lafortune, Girard, & Lin, 2010, Wang, Lafortune, Lin, & Girard, 2010) 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 Huang, Rudie, and Lin (2008), 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 Cassez and Tripakis (2008) and Thorsley and Teneketzis (2007) for the centralized case and Wang, Girard, Lafortune, and Lin (2011) 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 to this is the work in Wang et al. (2011), 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 existing methodologies for solving (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 (Sears & Rudie, 2015).

The contributions of this paper are two-fold. First, we show how to transform *K-codiagnosability to coobservability* under a general language-based dynamic observations setting. 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*. Second, we provide a new approach for the verification of transition-based codiagnosability. Our method is different from that in Wang et al. (2011) and adopts the standard verifier approach which is used for static diagnosis problem in the literature (Jiang, Huang, Chandra, & Kumar, 2001; Qiu & Kumar, 2006; Wang, Yoo et al., 2007; Yoo & Lafortune, 2002). Our approach ends up with the same complexity as the approach proposed in Wang et al. (2011); however it allows us to derive an upper bound for the maximal delay of diagnosis, which is not provided in Wang et al. (2011). Moreover, by using the derived upper bound for the maximal diagnosis delay, we show that transition-based [co]diagnosability is transformable to transition-based [co]observability. Therefore, the standard notion of diagnosability from Sampath et al. (1995) can be transformed to the standard notion of observability from Lin and Wonham (1988). Our results thereby complement those in Wang et al. (2011) and allow leveraging the large existing literature on problems of decentralized control to solve problems of decentralized fault diagnosis.

The remaining part of this paper is organized as follows. Section 2 presents necessary preliminaries and in particular it reviews the notions of codiagnosability and coobservability. In Section 3, the transformation from K-codiagnosability to coobservability under language-based observations is presented. In Section 4, we present a new approach for the verification of transition-based codiagnosability, with which an upper bound of the diagnosis delay for decentralized diagnosis under transition-based observations is derived. We illustrate the application of the transformation algorithm of Section 3 to sensor activation problems in Section 5. Finally, we conclude the paper in Section 6. Preliminary and partial versions of some of the results in Sections 3 and 5 are presented in Yin and Lafortune (2015).

2. Preliminaries

2.1. System model

We assume basic knowledge of DES and common notations (see, e.g., Cassandras & Lafortune, 2008). A DES is modeled as a deterministic finite-state automaton $G = (X^G, E^G, \delta^G, x_0^G)$, 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 labeled by event e from state x to state y, and $x_0^G \in X^G$ is the initial state. Function δ^G is extended to $X^G \times E^{G*}$ in the usual way. The behavior generated by G is described by $\mathcal{L}(G) = \{s \in E^{G*} : \delta^G(x_0^G, s)!\}$, where ! means "is defined". The set of transitions $TR(G)$ of G is defined by $TR(G) := \{(x, e) \in X^G \times E^G : \delta^G(x, e)!\}$. 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 $\mathcal{I} = \{1, \dots, n\}$ 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 $E_{o,i} \subset E^G$ 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 an event being observed on occurrence of one transition but not observed on occurrence of a different transition. Also, under energy, bandwidth, or security constraints, a local agent may choose 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:

$$P_{\omega_i}(\epsilon) = \epsilon, \qquad P_{\omega_i}(s\sigma) = \begin{cases} P_{\omega_i}(s)\sigma & \text{if } \sigma \in \omega_i(s) \\ P_{\omega_i}(s) & \text{if } \sigma \notin \omega_i(s). \end{cases} \tag{1}$$

The inverse of P_{ω_i}, denoted by $P_{\omega_i}^{-1}$, is defined as $P_{\omega_i}^{-1} : E_o{}^* \to 2^{E^{G*}}$ with $P_{\omega_i}^{-1}(s) := \{t \in E^{G*} : P_{\omega_i}(t) = s\}$. The projection P_{ω_i} and its inverse $P_{\omega_i}^{-1}$ are extended to languages in the usual way. Clearly, if the set of observable events is fixed in the sense that $\forall s \in \mathcal{L}(G), \omega_i(s) = E_{o,i}$, then the projection P_{ω_i} reduces to the standard natural projection.

The above definition of observation mapping is language-based; as such, it may require infinite memory to realize. In practice, one is often interested in studying a particular type of dynamic observation, namely, *transition-based* dynamic observation. Formally, for each agent $i \in \mathcal{I}$, we say that an observation mapping $\omega_i : \mathcal{L}(G) \to 2^{E_{o,i}}$ is transition-based if

$$(\forall s, t \in \mathcal{L}(G))[\delta^G(x_0^G, s) = \delta^G(x_0^G, t) \Rightarrow \omega_i(s) = \omega_i(t)]. \tag{2}$$

Thus, a transition-based observation mapping ω_i can also be described by a set of observable transitions $\Omega_i \subseteq TR(G)$ defined by $\Omega_i := \{(x, e) \in TR(G) : \exists s \in \mathcal{L}(G) \text{ s.t. } \delta^G(x_0^G, s) = x \wedge e \in \omega_i(s)\}$.

The transition-based observation mapping also induces a projection $P_{\Omega_i} : \mathcal{L}(G) \rightarrow E_{o,i}^*$, which can be computed recursively as follows:

$$P_{\Omega_i}(\epsilon) = \epsilon,$$

$$P_{\Omega_i}(s\sigma) = \begin{cases} P_{\Omega_i}(s)\sigma & \text{if } (\delta^G(x_0^G, s), \sigma) \in \Omega_i \\ P_{\Omega_i}(s) & \text{if } (\delta^G(x_0^G, s), \sigma) \notin \Omega_i. \end{cases} \quad (3)$$

2.2. 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*: $E_F = E_{F_1} \dot\cup \cdots \dot\cup E_{F_m}$; we denote by Π_F this partition and by $\mathcal{F} = \{1, \ldots, m\}$ the index set of the fault types. For any $j \in \mathcal{F}$, we define $\Psi(E_{F_j}) = \{sf \in \mathcal{L}(G) : f \in E_{F_j}\}$ to be the set of strings that end with a fault event of type F_j. We write $E_{F_j} \in s$, if $\overline{\{s\}} \cap \Psi(E_{F_j}) \neq \emptyset$. 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; this assumption is commonly made in the DES fault diagnosis literature to simplify the technical development. We denote 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 (at least) one diagnoser within a finite number of steps, i.e., event occurrences, after the fault 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 the language is K-codiagnosable. The formal definition of [K-]codiagnosability under dynamic observations is recalled from Wang et al. (2011).

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

$$(\forall j \in \mathcal{F})(\forall s \in \Psi(E_{F_j}))(\forall t \in \mathcal{L}(H)/s)[|t| \geq K \Rightarrow CD]$$

where the codiagnosability condition CD is

$$(\exists i \in \mathcal{I})(\forall w \in \mathcal{L}(H))[P_{\omega_i}(w) = P_{\omega_i}(st) \Rightarrow E_{F_j} \in w].$$

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

Since K-codiagnosability explicitly specifies a uniform detection delay bound for all fault event occurrences, it is a stronger property than codiagnosability: K-codiagnosability implies codiagnosability, but the reverse may not hold for some values of K.

Remark 1. The above definition of codiagnosability is equivalent to the one in Debouk et al. (2000) and Wang, Yoo et al. (2007) in the case of regular languages (i.e., finite state systems), as is assumed in this paper. Specifically, the definition in Debouk et al. (2000) and Wang, Yoo et al. (2007) 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 Yoo and Garcia (2009) that in the case of regular languages, the two definitions are equivalent in the centralized case for static observation mappings. The result in Yoo and Garcia (2009) can be extended to the decentralized case and to language-based observation mappings, although the proof is omitted here.

In decentralized supervisory 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 $E_{c,i} \subseteq E^G$ its set of controllable events and denote by $E_c = \cup_{i \in \mathcal{I}} E_{c,i}$ the total set of controllable events. A local supervisor is a mapping $S_i : E_{o,i}^* \rightarrow 2^{E_{c,i}}$ 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 Huang et al. (2008) and Wang et al. (2011). Let $I^c(\sigma) := \{i \in \mathcal{I} : \sigma \in E_{c,i}\}$.

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

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

Note that in both Definitions 1 and 2, codiagnosability and coobservability 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 will 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.

3. 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.

3.1. Case of one fault type

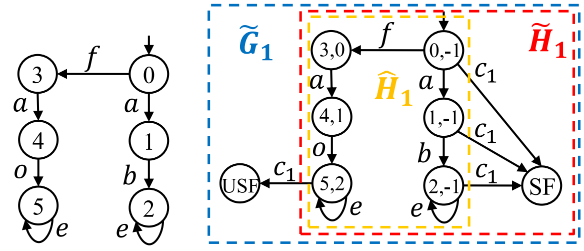
First, we show that the notion of K-codiagnosability can be transformed to coobservability when there is only one type of fault events. We shall need the notation $A \sqsubseteq B$ to denote that automaton A is a *sub-automaton* of automaton B, as defined in Cassandras and Lafortune (2008, p. 86). Let $\mathcal{L}(H)$ be the language to be diagnosed, where $H = (X^H, E^H, \delta^H, x_0^H)$, and E_{F_j} be the unique set of fault events under consideration (i.e., only type j faults are to be diagnosed). We construct two automata $\tilde{H}_j = (X^{\tilde{H}_j}, E^{\tilde{H}_j}, \delta^{\tilde{H}_j}, x_0^{\tilde{H}_j})$ and $\tilde{G}_j = (X^{\tilde{G}_j}, E^{\tilde{G}_j}, \delta^{\tilde{G}_j}, x_0^{\tilde{G}_j})$ with $\tilde{H}_j \sqsubseteq \tilde{G}_j$, as follows.

Algorithm KCOD–COOB-I

Input: $H = (X^H, E^H, \delta^H, x_0^H)$, E_{F_j} and K.

Output: $\tilde{H}_j = (X^{\tilde{H}_j}, E^{\tilde{H}_j}, \delta^{\tilde{H}_j}, x_0^{\tilde{H}_j})$, $\tilde{G}_j = (X^{\tilde{G}_j}, E^{\tilde{G}_j}, \delta^{\tilde{G}_j}, x_0^{\tilde{G}_j})$ and $E_{c,i} = \{c_j\}, \forall i \in \mathcal{I}$.

Step 1: Build a new automaton $\hat{H}_j = (X^{\hat{H}_j}, E^{\hat{H}_j}, \delta^{\hat{H}_j}, x_0^{\hat{H}_j})$, where $X^{\hat{H}_j} \subseteq X^H \times \{-1, 0, 1, \ldots, K\}$ is the set of states; $E^{\hat{H}_j} = E^H$ is the set of events; $\delta^{\hat{H}_j} : X^{\hat{H}_j} \times E^{\hat{H}_j} \rightarrow X^{\hat{H}_j}$ is the partial



(a) H.

(b) $K = 2$.

Fig. 1. $\omega_1(s) = \{o, e\}$, $\forall s \in \mathcal{L}(H)$.

transition function where for any $\hat{x} = (x, n) \in X^{\hat{H}_j}$, $\delta^{\hat{H}_j}$ is defined by

$$\delta^{\hat{H}_j}(\hat{x},\sigma)=\begin{cases}(\delta^H(x,\sigma),-1), & \text{if } \begin{pmatrix} n=-1 \text{ and}\\ \sigma\in E^H\setminus E_{F_j}\end{pmatrix}\\ (\delta^H(x,\sigma),n+1), & \\ \quad\text{if } \begin{pmatrix}0\le n<K \text{ or}\\ n=-1\wedge\sigma\in E_{F_j}\end{pmatrix} & \\ (\delta^H(x,\sigma),K), & \text{if } n=K\end{cases} \tag{4}$$

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

Step 2: Set $\tilde{H}_j \leftarrow \hat{H}_j$. Set $X^{\tilde{H}_j} \leftarrow X^{\hat{H}_j} \cup \{SF\}$ and set $E^{\tilde{H}_j} \leftarrow E^{\hat{H}_j} \cup \{c_j\}$.

Step 3: For all $\hat{x} = (x, n) \in X^{\hat{H}_j}$ in $\tilde{H}_j$, if $n = -1$, then add new transition $\delta^{\tilde{H}_j}(\hat{x}, c_j) = SF$.

Step 4: Set $\tilde{G}_j \leftarrow \tilde{H}_j$. Set $X^{\tilde{G}_j} \leftarrow X^{\tilde{G}_j} \cup \{USF\}$.

Step 5: For all $\hat{x} = (x, n) \in X^{\hat{H}_j}$ in $\tilde{G}_j$, if $n = K$, then add new transition $\delta^{\tilde{G}_j}(\tilde{x}, c_j) = USF$.

Step 6: For all $i \in \mathcal{I}$, $E_{c,i} \leftarrow \{c_j\}$.

For the above automaton $\tilde{G}_j$, the observation mapping $\omega_{i,\tilde{G}_j}$ of each agent $i \in \mathcal{I}$ is specified as follows. For all $s \in \mathcal{L}(\hat{H}_j)$, $\omega_{i,\tilde{G}_j}(s) \leftarrow \omega_i(s)$. For all $s = tc_j \in \mathcal{L}(\tilde{G}_j) \setminus \mathcal{L}(\hat{H}_j)$, $\omega_{i,\tilde{G}_j}(s) \leftarrow \omega_i(t)$. In fact, we do not need to compute $\omega_{i,\tilde{G}_j}$ explicitly, since we can use H and ω_i to simulate $\omega_{i,\tilde{G}_j}$. Specifically, for any string $s \in \mathcal{L}(\tilde{G}_j)$, we can use H to determine if $s \in \mathcal{L}(H) = \mathcal{L}(\hat{H})$. If so, we return $\omega_i(s)$. Otherwise, we know that $s = tc_j \in \mathcal{L}(\tilde{G}_j) \setminus \mathcal{L}(\hat{H}_j)$ and return $\omega_i(t)$.

Example 1. Consider the centralized static diagnosis problem instance shown in Fig. 1(a). $\mathcal{L}(H)$ is the language to be diagnosed when $E_F = \{f\}$ is the set of fault events and $E_o = \{o, e\}$ is the set of observable events. The observation mapping ω_1 is given by $\forall s \in \mathcal{L}(G), \omega_1(s) = \{o, e\}$. When the desired diagnosis delay is set to $K = 2$, by applying Algorithm KCOD–COOB-I, the corresponding $\tilde{G}_1$ and $\tilde{H}_1$ are constructed; they are shown in Fig. 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 observable w.r.t. $\mathcal{L}(\tilde{G}_1)$, $\omega_{\tilde{G}_1}$ and $\{c_1\}$, since we know for sure that we need to disable c_1 after the occurrence of string *fao*. The original system H is 2-diagnosable. However, it can be verified that H is not 1-diagnosable. The relationship between H, $\tilde{H}_j$ and $\tilde{G}_j$ will be formally described in Theorem 1. □

Before we present the correctness of Algorithm KCOD–COOB-I, we first discuss the intuition behind the construction procedure in it. 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., Wang, Lafortune, and Lin (2007). Clearly, we see that $\hat{H}_j$ is a finite unfolding of H and they are language equivalent, i.e., $\mathcal{L}(H) = \mathcal{L}(\hat{H}_j)$. Let us define the set of *conflicting* states pairs

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

where $[u]_n$ denotes the integer component of u. Let $\mathcal{E}_{\omega_i}(s)$ be the state estimator of Agent i upon the occurrence $s \in \mathcal{L}(\hat{H}_j)$, i.e., $\mathcal{E}_{\omega_i}(s) := \{x \in X^{\hat{H}_j} : \exists t \in P_{\omega_i}^{-1}(P_{\omega_i}(s)) \text{ s.t. } \delta^{\hat{H}_j}(x_0^{\hat{H}_j}, t) = x\}$. If $\mathcal{L}(\hat{H}_j)$ is K-codiagnosable, then for any string $s \in \mathcal{L}(\hat{H}_j)$, there exists at least one agent $i \in \mathcal{I}$ such that it is not confused with any state pair in T_{conf} after s, i.e., $T_{conf} \cap (\mathcal{E}_{\omega_i}(s) \times \mathcal{E}_{\omega_i}(s)) = \emptyset$. In the context of supervisory control, by construction, to achieve specification $\mathcal{L}(\tilde{H}_j)$ for plant $\mathcal{L}(\tilde{G}_j)$, we always need to enable c_j at states labeled with integer -1 and disable c_j at states labeled with integer K. Thus, for any execution of the system, we also need that at least one agent can 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_j. 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. ω_i, $i \in \mathcal{I}$ and fault event set E_{F_j}, if and only if, $\mathcal{L}(\tilde{H}_j)$ is coobservable w.r.t. $\mathcal{L}(\tilde{G}_j)$, $\omega_{i,\tilde{G}_j}$ and $E_{c,i} = \{c_j\}$, $i \in \mathcal{I}$.*

Proof. ($\Rightarrow$) By contrapositive. Suppose that $\mathcal{L}(\tilde{H}_j)$ is not coobservable. This implies that there exists a string $t \in \mathcal{L}(\tilde{H}_j)$ and an event $c_j \in E_c = \{c_j\}$, such that $tc_j \in \mathcal{L}(\tilde{G}_j) \setminus \mathcal{L}(\tilde{H}_j)$ and

$$(\forall i \in \mathcal{I})(\exists s \in \mathcal{L}(\tilde{H}_j))[P_{\omega_{i,\tilde{G}_j}}(s) = P_{\omega_{i,\tilde{G}_j}}(t) \wedge sc_j \in \mathcal{L}(\tilde{H}_j)]. \tag{5}$$

Consider the above s and t. By the transformation algorithm, since $sc_j \in \mathcal{L}(\tilde{H}_j)$, we know that $[\delta^{\tilde{H}_j}(x_0^{\tilde{H}_j}, s)]_n = -1$, which means that $E_{F_j} \notin s$. Similarly, $tc_j \in \mathcal{L}(\tilde{G}_j) \setminus \mathcal{L}(\tilde{H}_j)$ implies that $[\delta^{\tilde{H}_j}(x_0^{\tilde{H}_j}, t)]_n = K$. Then we can write $t = t_1t_2$ such that $t_1 \in \Psi(F_j)$ and $|t_2| \geq K$. Note that strings s and t also exist in $\mathcal{L}(H)$ and, by construction, we have that $P_{\omega_i}(s) = P_{\omega_{i,\tilde{G}_j}}(s) = P_{\omega_{i,\tilde{G}_j}}(t) = P_{\omega_i}(t)$. Thus, we know that

$$\begin{aligned}&(\exists t_1 \in \Psi(E_{F_j}))(\exists t_2 \in \mathcal{L}(H)/t_1 : |t_2| \geq K)\\ &\quad(\forall i \in \mathcal{I})(\exists s \in \mathcal{L}(H))[P_{\omega_i}(s) = P_{\omega_i}(t_1t_2) \wedge E_{F_j} \notin s]\end{aligned} \tag{6}$$

which is a violation of K-codiagnosability.

($\Leftarrow$) By contrapositive. Suppose that $\mathcal{L}(H)$ is not K-codiagnosable w.r.t. P_{ω_i}, $i \in \mathcal{I}$ and E_F. This implies that

$$\begin{aligned}&(\exists s \in \Psi(E_{F_j}))(\exists t \in \mathcal{L}(H)/s : |t| \geq K)\\ &\quad(\forall i \in \mathcal{I})(\exists w \in \mathcal{L}(H))[P_{\omega_i}(w) = P_{\omega_i}(st) \wedge E_{F_j} \notin w].\end{aligned} \tag{7}$$

By the transformation algorithm, we know that $[\delta^{\tilde{G}_j}(x_0^{\tilde{G}_j}, st)]_n = K$ and $[\delta^{\tilde{G}_j}(x_0^{\tilde{G}_j}, w)]_n = -1$ in $\tilde{G}_j$, which means that $stc_j \in \mathcal{L}(\tilde{G}_j) \setminus$

$\mathcal{L}(\tilde{H}_j)$ and $wc_j \in \mathcal{L}(\tilde{H}_j)$. Thus, there exists $st \in \mathcal{L}(\tilde{H}_j)$ and there exists $c_j \in E_c = \{c_j\}$, such that $stc_j \in \mathcal{L}(\tilde{G}_j) \setminus \mathcal{L}(\tilde{H}_j)$ and for all $i \in \mathcal{I}$ we have

$$(\exists w \in \mathcal{L}(\tilde{H}_j))[P_{\omega_{i,\tilde{G}_j}}(st) = P_{\omega_{i,\tilde{G}_j}}(w) \wedge wc_j \in \mathcal{L}(\tilde{H}_j)]$$

which is a violation of coobservability. □

Remark 2. Let $\mathcal{L}(H)$ be the language to be diagnosed. In Algorithm KCOD–COOB-I, the construction of automaton $\hat{H}_j$ in Step 1 can be done in time $O(K|X^H||E^H|)$. The computational effort for Steps 2, 4 and 6 is a constant. Steps 3 and 5 require the search of the state space of $\hat{H}$, which can be done in time $O(K|X^H|)$. Consequently, the worst-case time complexity of Algorithm KCOD–COOB-I is $O(K|X^H||E^H|)$.

3.2. *Case of multiple fault types*

Building on the results in the preceding section, 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., Cassandras & Lafortune, 2008).

Algorithm KCOD–COOB-II

```
Input: $H = (X^H, E^H, \delta^H, x_0^H)$, $E_F$, $\Pi_F$ and $K$.
Output: $\tilde{H} = (X^{\tilde{H}}, E^{\tilde{H}}, \delta^{\tilde{H}}, x_0^{\tilde{H}})$, $\tilde{G} = (X^{\tilde{G}}, E^{\tilde{G}}, \delta^{\tilde{G}}, x_0^{\tilde{G}})$ and $E_{c,i} = \{c_j : j \in \mathcal{F}\}, \forall i \in \mathcal{I}$.
Step 1: For each type of fault $j \in \mathcal{F}$, build an automaton $\hat{H}_j$, 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 $\tilde{H} \leftarrow \hat{H}$. Set $X^{\tilde{H}} \leftarrow X^{\tilde{H}} \cup \{SF\}$ and set $E^{\tilde{H}} \leftarrow E^{\tilde{H}} \cup \{c_j : j \in \mathcal{F}\}$.
Step 4: For all $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_{|\mathcal{F}|}) \in X^{\hat{H}}$ in $\tilde{H}$, for all $j \in \mathcal{F}$, if
        $[\hat{x}_j]_n = -1$, then add new transition $\delta^{\tilde{H}}(\hat{x}, c_j) = SF$.
Step 5: Set $\tilde{G} \leftarrow \tilde{H}$. Set $X^{\tilde{G}} \leftarrow X^{\tilde{G}} \cup \{USF\}$.
Step 6: For all $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_{|\mathcal{F}|}) \in X^{\hat{H}}$ in $\tilde{G}$, for all $j \in \mathcal{F}$, if
        $[\hat{x}_j]_n = K$, then add new transition $\delta^{\tilde{G}}(\tilde{x}, c_j) = USF$.
Step 7: For all $i \in \mathcal{I}, E_{c,i} \leftarrow \{c_j : j \in \mathcal{F}\}$.
```

For the above automaton $\tilde{G}$, the observation mapping $\omega_{i,\tilde{G}}$ of each agent $i \in \mathcal{I}$ 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 $c \in \{c_j : j \in \mathcal{F}\}$, $\omega_{i,\tilde{G}}(s) \leftarrow \omega_i(t)$. In fact, we still do not need to compute $\omega_{i,\tilde{G}}$, since we can still use H and ω_i to simulate $\omega_{i,\tilde{G}}$ as we discussed earlier for the case of one fault type.

Remark 3. Algorithm KCOD–COOB-II essentially *merges* all automata $\hat{H}_j, j \in \mathcal{F}$, constructed by Algorithm KCOD–COOB-I into a single automaton $\hat{H}$. Then single copies of the new states SF and USF are added. Note that, in Step 2 of Algorithm KCOD–COOB-II, the parallel composition between $\hat{H}_j, j \in \mathcal{F}$, could have resulted in an automaton with $\prod_{j \in \mathcal{F}} |X^{\hat{H}_j}|$ 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., $|X^{\hat{H}}| \leq K^m |X^H|$.

The following result shows the correctness of the transformation in Algorithm KCOD–COOB-II.

Theorem 2. *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 $E_{c,i} = \{c_j : j \in \mathcal{F}\}, i \in \mathcal{I}$.*

Proof. In order to prove the result, it suffices to show that the following 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 $j \in \mathcal{F}$, $\mathcal{L}(H)$ is K-coodiagnosable w.r.t. $\omega_i, i \in \mathcal{I}$ and the fault event set E_{F_j}.

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

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

S1 $\Leftrightarrow$ S2 follows from Definition 1 and S2 $\Leftrightarrow$ S3 follows from Theorem 1. Thus, it remains to show that S3 and S4 are equivalent. For S3 and S4, we have

S3 is false

$$\Leftrightarrow (\exists j \in \mathcal{F})(\exists t \in \mathcal{L}(\tilde{H}_j) : tc_j \in \mathcal{L}(\tilde{G}_j) \setminus \mathcal{L}(\tilde{H}_j))(\forall i \in \mathcal{I})$$
$$(\exists s \in \mathcal{L}(\tilde{H}_j))[P_{\omega_{i,\tilde{G}_j}}(s) = P_{\omega_{i,\tilde{G}_j}}(t) \wedge sc_j \in \mathcal{L}(\tilde{H}_j)] \quad (8)$$
$$\Leftrightarrow (\exists c_j \in E_c)(\exists t \in \mathcal{L}(\tilde{H}) : tc_j \in \mathcal{L}(\tilde{G}) \setminus \mathcal{L}(\tilde{H}))(\forall i \in \mathcal{I})$$
$$(\exists s \in \mathcal{L}(\tilde{H}))[P_{\omega_{i,\tilde{G}}}(s) = P_{\omega_{i,\tilde{G}}}(t) \wedge sc_j \in \mathcal{L}(\tilde{H})] \quad (9)$$

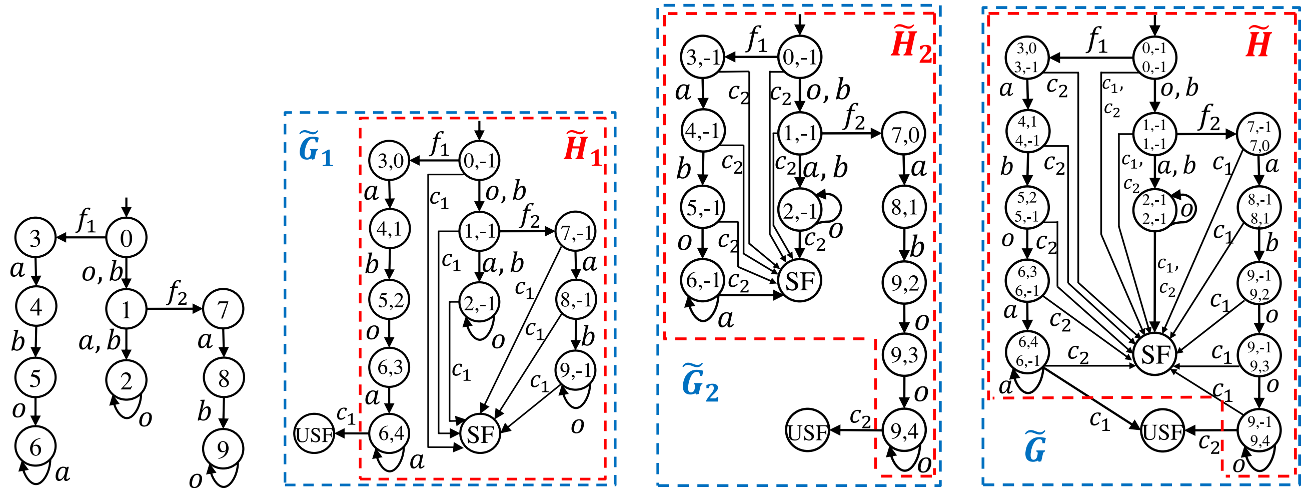
$\Leftrightarrow$ S4 is false.

To see why the second equivalence holds, let us first suppose that Eq. (8) holds and consider the same t, s and c_j in Eq. (8). Since $tc_j \in \mathcal{L}(\tilde{G}_j) \setminus \mathcal{L}(\tilde{H}_j)$ and $sc_j \in \mathcal{L}(\tilde{H}_j)$, by the definitions of $\tilde{G}_j$ and $\tilde{H}_j$, we know that $s, t \in \mathcal{L}(\hat{H}_j) = \mathcal{L}(\hat{H}) \subseteq \mathcal{L}(\tilde{H})$, $E_{F_j} \notin s$ and $E_{F_j} \in t$. By the definitions of $\tilde{G}$ and $\tilde{H}$, we have that $tc_j \in \mathcal{L}(\tilde{G}) \setminus \mathcal{L}(\tilde{H})$ and $sc_j \in \mathcal{L}(\tilde{H})$. Moreover, since $\forall s \in \mathcal{L}(\hat{H}) = \mathcal{L}(H) : \omega_i(s) = \omega_{i,\tilde{G}}(s) = \omega_{i,\tilde{G}_j}(s)$, $P_{\omega_{i,\tilde{G}_j}}(s) = P_{\omega_{i,\tilde{G}_j}}(t)$ implies that $P_{\omega_{i,\tilde{G}}}(s) = P_{\omega_{i,\tilde{G}}}(t)$. Therefore, we know that Eq. (9) holds for the same t, s and c_j.

Similarly, suppose that Eq. (9) holds and consider the same t, s and c_j in Eq. (9). Since $tc_j \in \mathcal{L}(\tilde{G}) \setminus \mathcal{L}(\tilde{H})$ and $sc_j \in \mathcal{L}(\tilde{H})$, by the definitions of $\tilde{G}$ and $\tilde{H}$, we know that $s, t \in \mathcal{L}(\hat{H}) = \mathcal{L}(\hat{H}_j) \subseteq \mathcal{L}(\tilde{H}_j)$, $E_{F_j} \notin s$ and $E_{F_j} \in t$. By the definitions of $\tilde{G}_j$ and $\tilde{H}_j$, we have that $tc_j \in \mathcal{L}(\tilde{G}_j) \setminus \mathcal{L}(\tilde{H}_j)$ and $sc_j \in \mathcal{L}(\tilde{H}_j)$. Moreover, since $\forall s \in \mathcal{L}(\hat{H}) = \mathcal{L}(H) : \omega_i(s) = \omega_{i,\tilde{G}}(s) = \omega_{i,\tilde{G}_j}(s)$, $P_{\omega_{i,\tilde{G}}}(s) = P_{\omega_{i,\tilde{G}}}(t)$ implies that $P_{\omega_{i,\tilde{G}_j}}(s) = P_{\omega_{i,\tilde{G}_j}}(t)$. Therefore, we know that Eq. (8) holds for the same t, s and j. □

Remark 4. Let $\mathcal{L}(H)$ be the language to be diagnosed. In Algorithm KCOD–COOB-II, the construction of all the automata $\hat{H}_j, j \in \mathcal{F}$, in Step 1 can be done in time $O(mK|X^H||E^H|)$. The parallel composition in Step 2 can also be done in time $O(K^m|X^H||E^H|)$, since $\hat{H}$ has at most $K^m|X^H|$ states as discussed in Remark 3. The computational effort for Steps 3, 5 and 7 is a constant. Steps 4 and 6 require the search of the state space of $X^{\hat{H}}$, which can be done in time $O(K^m|X^H|)$. Consequently, the worst-case time complexity of Algorithm KCOD–COOB-II is $O(K^m|X^H||E^H|)$.

Example 2. Let the automaton H shown in Fig. 2 be the system to be diagnosed. Suppose that $E_{F_1} = \{f_1\}$ and $E_{F_2} = \{f_2\}$ are the 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 Fig. 2(b)–(d).



(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 $E_{o,1} = \{a, o\}, E_{o,2} = \{b, o\}$.

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 $s_2 = oboo$ such that $P_2(s) = P_2(s_2)$ 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 by Theorem 2. □

3.3. Case of event-based observations

In transformation Algorithms KCOD–COOB-I and KCOD–COOB-II in Section 3, the desired diagnosis delay K is specified a priori and the observations are language-based. Let us eliminate that extra level of generality and assume that there is only one agent and the set of observable events $E_{o,1} \subseteq E^H$ is fixed a priori. In this case, it is possible to relax the pre-information on K and extend Algorithms KCOD–COOB-I and KCOD–COOB-II from K-diagnosability to diagnosability. We now explain how to proceed.

In Yoo and Lafortune (2002), 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 (Proposition 1 in Yoo & Lafortune, 2002). Such an upper bound of the diagnosis delay is derived from the size of *verifier*, a special type of automaton used for verifying diagnosability. Therefore, we obtain the following result.

Proposition 3. *When the observations are event-based, diagnosability can be transformed to observability in* $O(|X^H|^{2m+1}|E^H|)$.

Proof. Since H is diagnosable if and only if it is $|X^H|^2$-diagnosable (Proposition 1 in Yoo & Lafortune, 2002), we can simply use the upper bound $|X^H|^2$ to replace the integer K in $O(K^m|X^H||E^H|)$, which is the complexity of Algorithm KCOD–COOB-II. □

3.4. Case of language-based observations

In view of the results in the preceding section, a natural question to ask is: In the *dynamic* decentralized diagnosis problem, can we also find an upper bound to replace K, where this upper bound would work for any language-based mapping? The answer is that, in general, such an upper bound does not exist, since when the observation policy is language-based, K could be arbitrarily large. This phenomenon is illustrated by the following example.

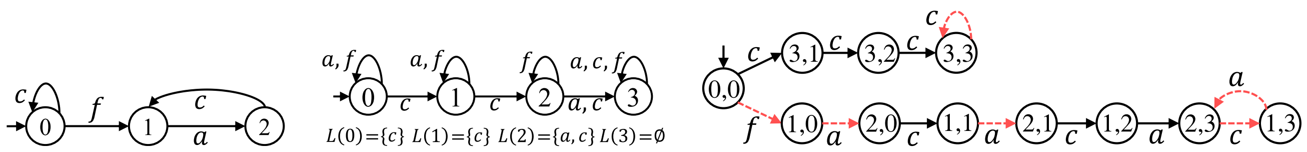
Example 3. Consider the automaton H in Fig. 3(a), where f is the unique fault event. Consider the observation mapping $\omega : \mathcal{L}(G) \to 2^{E_o}$ defined by:

$$\omega(s) = \begin{cases} \{c\}, & \text{if } s \in \overline{\{f(ac)^n, c^n\}} \\ \{a, c\}, & \text{if } s \in \mathcal{L}(G) \setminus \overline{\{f(ac)^n, c^n\}} \end{cases} \tag{10}$$

for some finite non-negative integer n. 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 arbitrarily large, there is no general upper bound for the diagnosis delay under language-based observations. □

The reason why such an upper bound does not exist is that the observation policy may require arbitrarily large memory to realize. In practice, for language-based observations, the observation policy needs to be realized by a finite structure, e.g., a finite state transducer; see, e.g., Cassez and Tripakis (2008). Formally, a finite state transducer is a deterministic labeled automaton $T = (A, L)$, where $A = (X^A, E^A, f^A, x_0^A)$ is a finite state automaton such that $E^A = E^H$ and $L : X^A \to 2^{E^A}$ is a labeling function that specifies the set of events that are observable at state x. Therefore, given a system H with dynamic observations specified by transducer $T = (A, L)$, we can construct a new system $H_T := H \parallel A$ with transition-based observation Ω_T, where Ω_T is specified by: for any $(x_G, x_T) \in X^{H_T}$, for any $\sigma \in E^{H_T}$, $((x_G, x_T), \sigma) \in \Omega_T$ if $\sigma \in L(x_T)$ and $((x_G, x_T), \sigma)$ is a transition in H_T. It follows from Cassez and Tripakis (2008) that H_T and Ω_T capture the information mapping of H and T. We use the following example to show how language-based observations can be reduced to transition-based observations by refining the system's original state space with the given transducer.

Example 4. Consider again the automaton H in Fig. 3(a), where f is the unique fault event. Consider the transducer T in Fig. 3(b), where the system decides to observe event a only after observing event c twice. The refined system H_T with transition-based observation



(a) Automaton H. (b) Transducer $T = (A, L)$ for H. (c) $H_T = H \parallel A$; Ω_T is the set of (black) solid-line transitions.

Fig. 3. Example of language-based observation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

map Ω_T is shown in Fig. 3(c), where the (red) dashed lines represent unobservable transitions. It can be verified that H_T is diagnosable under Ω_T. Therefore, the original system H is diagnosable under T. □

As discussed above, since any language-based observation map with finite memory can be reduced to a transition-based observation map, to show that language-based [co]diagnosability is transformable to language-based [co]observability, it suffices to find a general upper bound of the diagnosis delay *for the transition-based [co]diagnosis problem*. Such an upper bound does exist and we will derive it in the next section.

4. Verification of transition-based codiagnosability: a verifier approach

In this section, we first provide a new approach for the verification of transition-based diagnosability, by constructing a new automaton that we call the *T-Verifier*. Then, by using the T-Verifier, we derive an upper bound for the diagnosis delay with transition-based observations. For the sake of simplicity, the verification algorithm will be presented by assuming that there are two agents, i.e., $\mathcal{I} = \{1, 2\}$, and that there is only one type of faults, i.e., $E_F = E_{F_1}$. The latter assumption is without loss of generality, since the diagnosis of each fault can be analyzed individually; Section 4.3 contains further discussion on the case of multiple fault types. The case of multiple agents is discussed in Section 4.2.

4.1. T-verifier

First, we denote by G_N the accessible part of G after removing all fault transitions in G, i.e., G_N models the "non-faulty" behavior of G. Then, given the automaton G, fault event set E_F, and local agents with transition-based observations $\Omega_i, i = 1, 2$, the T-verifier V is defined as the *deterministic* automaton

$$V = (X^V, E^V, \delta^V, x_0^V)$$

where:

(i) $X^V = X^G \times X^{G_N} \times X^{G_N} \times L$, where $L = \{N, F\}$ is the label set;
(ii) $E^V = (E^G \cup \{\epsilon\}) \times (E^G \cup \{\epsilon\}) \times (E^G \cup \{\epsilon\})$, is the event set;
(iii) $x_0^V = (x_0^G, x_0^{G_N}, x_0^{G_N}, N)$ is the initial state;
(iv) $\delta^V : X^V \times E^V \to X^V$ is the partial (deterministic) transition function defined according to the even cases (a)–(g) below:

(a) For $\sigma \in E_F$,
$$\delta^V((x_0, x_1, x_2, l), (\sigma, \epsilon, \epsilon)) = (\delta^G(x_0, \sigma), x_1, x_2, F). \tag{11}$$
(b) For $\sigma \in E^G \setminus E_F$ such that A.1 in Table 1 holds
$$\delta^V((x_0, x_1, x_2, l), (\sigma, \sigma, \sigma)) = (\delta^G(x_0, \sigma), \delta^{G_N}(x_1, \sigma), \delta^{G_N}(x_2, \sigma), l). \tag{12}$$
(c) For $\sigma \in E^G \setminus E_F$ such that C.3, C.4, D.3 or D.4 in Table 1 holds
$$\delta^V((x_0, x_1, x_2, l), (\sigma, \epsilon, \epsilon)) = (\delta^G(x_0, \sigma), x_1, x_2, l). \tag{13}$$
(d) For $\sigma \in E^G \setminus E_F$ such that B.1, B.2, B.3, B.4, D.1, D.2, D.3 or D.4 in Table 1 holds
$$\delta^V((x_0, x_1, x_2, l), (\epsilon, \sigma, \epsilon)) = (x_0, \delta^{G_N}(x_1, \sigma), x_2, l). \tag{14}$$
(e) For $\sigma \in E^G \setminus E_F$ such that A.2, A.4, B.2, B.4, C.2, C.4, D.2 or D.4 in Table 1 holds
$$\delta^V((x_0, x_1, x_2, l), (\epsilon, \epsilon, \sigma)) = (x_0, x_1, \delta^{G_N}(x_2, \sigma), l). \tag{15}$$
(f) For $\sigma \in E^G \setminus E_F$ such that A.3 or A.4 in Table 1 holds
$$\delta^V((x_0, x_1, x_2, l), (\sigma, \sigma, \epsilon)) = (\delta^G(x_0, \sigma), \delta^{G_N}(x_1, \sigma), x_2, l). \tag{16}$$
(g) For $\sigma \in E^G \setminus E_F$ such that C.1 or D.1 in Table 1 holds
$$\delta^V((x_0, x_1, x_2, l), (\sigma, \epsilon, \sigma)) = (\delta^G(x_0, \sigma), x_1, \delta^{G_N}(x_2, \sigma), l). \tag{17}$$

This completes the definition of the T-verifier.

Table 1 captures the 16 different combinations of transition-based observations from a state triple (x_0, x_1, x_2). Intuitively, the T-Verifier tracks one string in $\mathcal{L}(G)$ and two strings in $\mathcal{L}(G_N)$ that look identical for agents 1 and 2 under their own observations. To formalize this assertion, we introduce the following notation: for any string $t = \sigma_1 \ldots \sigma_n \in \mathcal{L}(V), \sigma_i = (\sigma_i^0, \sigma_i^1, \sigma_i^2)$, we define $\theta_k(t), k = 0, 1, 2$ to be the restriction of t to each of its components; namely, $\theta_k(t) = \sigma_1^k \ldots \sigma_n^k \in \mathcal{L}(G), k = 0, 1, 2$. Then we have the following results. The proofs of Lemmas 4 and 5 are given in the Appendix.

Lemma 4. *For any $t \in \mathcal{L}(V)$ and for any $i \in \{1, 2\}$, we have*

$$P_{\Omega_i}(\theta_0(t)) = P_{\Omega_i}(\theta_i(t)). \tag{18}$$

Lemma 5. *For any $s_0 \in \mathcal{L}(G), s_1, s_2 \in \mathcal{L}(G_N)$ such that $P_{\Omega_i}(s_0) = P_{\Omega_i}(s_i), \forall i = 1, 2$, there exists a string $t \in \mathcal{L}(V)$ such that $\theta_i(t) = s_i, \forall i = 0, 1, 2$.*

Definition 3 (*Path and Cycle*)**.** We call a sequence of states and events in the form of $t = v_0\sigma_0 \ldots v_{p-1}\sigma_{p-1}v_p,\ v_i \in X^V, \sigma_i \in E$ a *path* in V, if $v_{i+1} = \delta^V(v_i, \sigma_i), \forall i \in \{0, \ldots, p-1\}$. We define the length of a path as the number of events in it, e.g., $|t| = p$. We say that a path forms a *cycle* if $v_0 = v_p$.

A cycle $v_0\sigma_0 \ldots v_{p-1}\sigma_{p-1}v_p$ in V is said to be *F-real* if:

(1) $(v_i)_l = F, \forall i \in \{0, 1, \ldots, p-1\}$, where $(\cdot)_l$ means the label component of v_i and;
(2) $\exists j \in \{0, 1, \ldots, p-1\}$ such that $\theta_0(\sigma_j) \neq \epsilon$.

The following set of results establish how the T-Verifier can be applied to the verification of transition-based codiagnosability. The relationship between these results is depicted in Fig. 4.

Lemma 6. *$\mathcal{L}(G)$ is not codiagnosable w.r.t. Ω_1, Ω_2 and E_F if there exists an F-real cycle in V.*

Proof. Suppose that there exists an F-real cycle $v_p\sigma_pv_{p+1} \ldots v_{p+m}\sigma_{p+m}v_p$ in V. Since $(v_i)_l = F, \forall i \in \{p, p+1, \ldots, p+m\}$, we know that there exists a path $v_0\sigma_0 \ldots v_{p-1}\sigma_{p-1}v_p$, where $v_0 = x_0^V$, such that

$$[\exists i \in \{0, \ldots, p-1\} \text{ s.t. } \theta_0(\sigma_i) \in E_F]$$
$$\wedge [st^n = \sigma_0 \ldots \sigma_{p-1}(\sigma_p \ldots \sigma_{p+m})^n \in \mathcal{L}(V),\ \forall n \in \mathbb{N}].$$

Considering the above s and t, we have that: (i) by the construction of V, $E_F \in \theta_0(s)$; and (ii) by Definition 3, $\theta_0(t) \neq \epsilon$.

Table 1
Shorthand symbol for different cases.

Case	Agent 1	Agent 2	Case	Agent 1	Agent 2
A.1	$(x_0,\sigma)\in\Omega_1, (x_1,\sigma)\in\Omega_1$	$(x_0,\sigma)\in\Omega_2, (x_2,\sigma)\in\Omega_2$	C.1	$(x_0,\sigma)\notin\Omega_1, (x_1,\sigma)\in\Omega_1$	$(x_0,\sigma)\in\Omega_2, (x_2,\sigma)\in\Omega_2$
A.2	Above	$(x_0,\sigma)\in\Omega_2, (x_2,\sigma)\notin\Omega_2$	C.2	Above	$(x_0,\sigma)\in\Omega_2, (x_2,\sigma)\notin\Omega_2$
A.3	Above	$(x_0,\sigma)\notin\Omega_2, (x_2,\sigma)\in\Omega_2$	C.3	Above	$(x_0,\sigma)\notin\Omega_2, (x_2,\sigma)\in\Omega_2$
A.4	Above	$(x_0,\sigma)\notin\Omega_2, (x_2,\sigma)\notin\Omega_2$	C.4	Above	$(x_0,\sigma)\notin\Omega_2, (x_2,\sigma)\notin\Omega_2$
B.1	$(x_0,\sigma)\in\Omega_1, (x_1,\sigma)\notin\Omega_1$	$(x_0,\sigma)\in\Omega_2, (x_2,\sigma)\in\Omega_2$	D.1	$(x_0,\sigma)\notin\Omega_1, (x_1,\sigma)\notin\Omega_1$	$(x_0,\sigma)\in\Omega_2, (x_2,\sigma)\in\Omega_2$
B.2	Above	$(x_0,\sigma)\in\Omega_2, (x_2,\sigma)\notin\Omega_2$	D.2	Above	$(x_0,\sigma)\in\Omega_2, (x_2,\sigma)\notin\Omega_2$
B.3	Above	$(x_0,\sigma)\notin\Omega_2, (x_2,\sigma)\in\Omega_2$	D.3	Above	$(x_0,\sigma)\notin\Omega_2, (x_2,\sigma)\in\Omega_2$
B.4	Above	$(x_0,\sigma)\notin\Omega_2, (x_2,\sigma)\notin\Omega_2$	D.4	Above	$(x_0,\sigma)\notin\Omega_2, (x_2,\sigma)\notin\Omega_2$

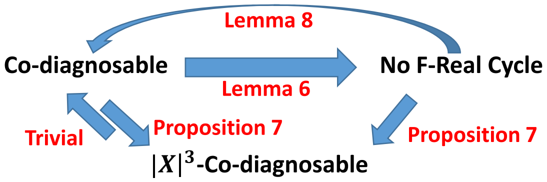


Fig. 4. Codiagnosability, $|X^G|^3$-codiagnosability and the non-existence of F-real cycle are equivalent.

Then, we define strings $w_0(n)\in\mathcal{L}(G)$ and $w_1(n), w_2(n)\in\mathcal{L}(G_N)$ by $w_i(n)=\theta_i(st^n), i=0,1,2, n\geq 0$. As discussed above, $|w_0(n)|\geq 1+n$. Moreover, by Lemma 4, for any $n\in\mathbb{N}$, we have $P_{\Omega_1}(w_0(n))=P_{\Omega_1}(w_1(n))$ and $P_{\Omega_2}(w_0(n))=P_{\Omega_2}(w_2(n))$. Since n can be arbitrarily large, $w_0(n)$ can be made arbitrarily long; this leads to a violation of codiagnosability. □

Moreover, the T-Verifier can also be used to estimate the upper bound for the diagnosis delay, as stated in the following proposition.

Proposition 7. *When $\mathcal{I}=\{1,2\}$, for transition-based observations, $\mathcal{L}(G)$ is codiagnosable w.r.t. Ω_1, Ω_2 and E_F if and only if it is $|X^G|^3$-codiagnosable.*

Proof. The "if" part is trivial. We show the "only if" part by contrapositive. Suppose that $\mathcal{L}(G)$ is not $|X^G|^3$-codiagnosable. Then there exists a string $st\in\mathcal{L}(G)$ such that $s\in\Psi(E_F)\wedge|t|=|X^G|^3$ and there exist strings $w_i:=s_it_i\in\mathcal{L}(G_N), i=1,2$, such that $P_{\Omega_i}(s)=P_{\Omega_i}(s_i), \forall i=1,2$ and $P_{\Omega_i}(st)=P_{\Omega_i}(s_it_i), \forall i=1,2$. Define the following notation: $x_s:=\delta^G(x_0^G,s)$, $x_{st}:=\delta^G(x_0^G,st)$ and $x_{s_i}:=\delta^{G_N}(x_0^{G_N},s_i)$, $x_{w_i}:=\delta^{G_N}(x_0^{G_N},w_i)$, $\forall i=1,2$. By Lemma 5, we know that $v_0:=(x_s,x_{s_1},x_{s_2},F)$ and $v_r:=(x_{st},x_{w_1},x_{w_2},F)$ are all reachable states in V with a path $g=v_0\sigma_0\ldots\sigma_{r-1}v_r$ such that $\sigma_0^0\ldots\sigma_{r-1}^0=t$ and $\sigma_0^i\ldots\sigma_{r-1}^i=t_i, i=1,2$, where $\sigma_l=(\sigma_l^0,\sigma_l^1,\sigma_l^2), \forall l=1,\ldots,r-1$. Therefore, there exist some $|t|$ integers $0\leq i_1<i_2<\cdots<i_{|t|}\leq r-1$ such that $\theta_0(\sigma_{i_l})=\sigma_{i_l}^0\neq\epsilon, \forall l=1,\ldots,|t|$. Since $i_{|t|}\geq|t|=|X^G|^3$ and there are at most $|X^G|^3$ states in V labeled by F, we know that there exist two integers $1\leq p<q\leq|t|$ such that $v_{i_p}\sigma_{i_p}v_{i_p+1}\ldots\sigma_{i_q-1}v_{i_q}$ forms a cycle and $\theta_0(\sigma_{i_p})=\sigma_{i_p}^0\neq\epsilon$. Thus, there exists an F-real cycle in V, which means that $\mathcal{L}(G)$ is not codiagnosable by Lemma 6. □

Lemma 8. *There exists an F-real cycle in V if $\mathcal{L}(G)$ is not codiagnosable w.r.t. Ω_1, Ω_2 and E_F.*

Proof. As depicted in Fig. 4, the proof of this lemma follows immediately from the proof of Proposition 7, in which we have shown that codiagnosability and $|X^G|^3$-codiagnosability are equivalent and non-$|X^G|^3$-codiagnosable implies the existence of an F-real cycle in V. □

Theorem 9. *$\mathcal{L}(G)$ is not codiagnosable w.r.t. Ω_1, Ω_2 and E_F if and only if there exists an F-real cycle in V.*

Proof. Follows from Lemmas 6 and 8. □

By Theorem 9, to verify transition-based codiagnosability, it suffices to verify the presence or not of an F-real cycle in V, which is similar to the static case (Qiu & Kumar, 2006; Wang, Yoo et al., 2007). We illustrate the verification procedure by the following example.

Example 5. Consider the system G in Fig. 5(a), where f is the single fault event. Two agents monitor the plant with transition-based observations Ω_1 and Ω_2, respectively, as shown in Fig. 5(b) and (c), where (red) dashed lines represent unobservable transitions. Part of the corresponding T-Verifier V is shown in Fig. 5(d). For example, at state $(0,0,0,N)$, we have that $(0,o),(0,o)\in\Omega_1$ and $(0,o),(0,o)\notin\Omega_2$. Therefore, events (ϵ,ϵ,o) and (o,o,ϵ) are defined at this state and according to the transition function, the successor states are $(0,0,1,N)$ and $(1,1,0,N)$, respectively. By definition, we see that the cycle $(3,3,3,F)(a,a,a)(3,3,3,F)$ is an F-real cycle. By Theorem 9, the system G is not codiagnosable w.r.t. Ω_1, Ω_2 and $\{f\}$. □

4.2. Case of multiple agents

In this section, we highlight how to generalize the results of the previous section to the case of n local agents. In this context, $X^V=X^G\times\underbrace{X^{G_N}\times\cdots\times X^{G_N}}_{n\text{ times}}\times L$, $E^V=\underbrace{(E^G\cup\{\epsilon\})\times\cdots\times(E^G\cup\{\epsilon\})}_{(n+1)\text{ times}}$, and the transition function $\delta^V:X^V\times E^V\times X^V$ is defined as follows:

(a) For $\sigma\in E_F$, then

$$\delta^V((x_0,x_1,\ldots,x_n,l),(\sigma,\epsilon,\ldots,\epsilon))$$
$$=(\delta^G(x_0,\sigma),x_1,\ldots,x_n,F). \tag{19}$$

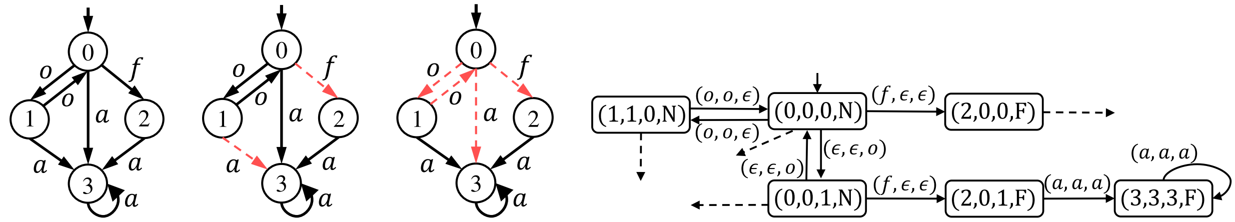
(b) For $\sigma\in E^G\setminus E_F$, we partition $\mathcal{I}$ into four disjoint sets $\mathcal{I}=\mathcal{I}_1\dot\cup\mathcal{I}_2\dot\cup\mathcal{I}_3\dot\cup\mathcal{I}_4$, where: (i) $\mathcal{I}_1=\{i\in\mathcal{I}:(x_0,\sigma)\in\Omega_i\wedge(x_i,\sigma)\in\Omega_i\}$; (ii) $\mathcal{I}_2=\{i\in\mathcal{I}:(x_0,\sigma)\in\Omega_i\wedge(x_i,\sigma)\notin\Omega_i\}$; (iii) $\mathcal{I}_3=\{i\in\mathcal{I}:(x_0,\sigma)\notin\Omega_i\wedge(x_i,\sigma)\in\Omega_i\}$; and (iv) $\mathcal{I}_4=\{i\in\mathcal{I}:(x_0,\sigma)\notin\Omega_i\wedge(x_i,\sigma)\notin\Omega_i\}$. Then, we have the following two types of transitions:

(b1) If $\mathcal{I}_2=\emptyset$, then the single transition

$$\delta^V((x_0,x_1,\ldots,x_n,l),(\sigma,e_1,\ldots,e_n))$$
$$=(\delta^G(x_0,\sigma),\delta^{G_N}(x_1,e_1),\ldots,\delta^{G_N}(x_n,e_n),l),$$
$$\text{where for any } i\in\mathcal{I}, e_i=\begin{cases}\sigma, & \text{if } i\in\mathcal{I}_1\\ \epsilon, & \text{if } i\in\mathcal{I}_3\cup\mathcal{I}_4.\end{cases} \tag{20}$$

(b2) The following transition for each $i\in\mathcal{I}_2\cup\mathcal{I}_4$,

$$\delta^V((x_0,x_1,\ldots,x_n,l),(\epsilon,\epsilon,\ldots,\epsilon,\underset{(i+1)^{\text{th}}}{\sigma},\epsilon,\ldots,\epsilon))$$
$$=(x_0,x_1,\ldots,x_{i-1},\delta^{G_N}(x_i,\sigma),x_{i+1},\ldots,x_n,l). \tag{21}$$



(a) System: G. (b) Agent 1: Ω_1. (c) Agent 2: Ω_2. (d) Part of the T-Verifier V for G, Ω_1 and Ω_2.

Fig. 5. Example of the verification of transition-based codiagnosability. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The above construction generalizes Eqs. (11)–(17) in the two agents' case. Specifically, Eq. (11) is generalized by Eq. (19), Eqs. (12), (13), (16), and (17) are special cases of Eq. (20), and Eqs. (14) and (15) are special cases of Eq. (21). In words, case (b1) covers all cases in Table 1 except A.2, C.2, D.2, and all cases B. In this case, the first component x_0 moves on observable transition σ and the agents that observe that transition move as well (first case of Eq. (20)) or do not move if they do not see the transition (second case in Eq. (20)). Case (b2) covers the remaining cases in Table 1, as well as cases A.4, C.4, D.3, and D.4. Here, the first component x_0 does not move and we list all individual moves of the agents that project to the empty string. Note that if $\mathcal{I}_2 \neq \emptyset$, then the first component x_0 cannot move as this would violate the conditions that projected strings are to remain the same.

Similar to the case of two agents, the diagnosis delay in the case of n agents is bounded by the size of the first $n+1$ components of the T-Verifier, i.e., $|X^G|^{n+1}$. Consequently, Proposition 7 can be extended to the n-agents' case as follows.

Proposition 10. *When $|\mathcal{I}| = n$, for transition-based observations, $\mathcal{L}(G)$ is codiagnosable w.r.t. E_F and Ω_i, $i \in \mathcal{I}$, if and only if it is $|X^G|^{n+1}$-codiagnosable.*

The proof of this result follows the same strategy as the proof of Proposition 7, but is more tedious.

Remark 5. When there are n local agents, the T-Verifier has at most $2|X^G|^{n+1}$ states. For each state, there are at most $|E^G| + n(|E^G| - |E_F|)$ transitions that originate from it ($|E^G|$ choices when the first component is involved and $n(|E^G| - |E_F|)$ choices when the remaining n components are involved). Thus, the total number of transitions in V is bounded by $2|X^G|^{n+1}(|E^G| + n(|E^G| - |E_F|))$. The verification of the existence of an F-real cycle is a strongly connected graph detection problem, which follows the same technique as described in Moreira et al. (2011), Qiu and Kumar (2006) and Wang, Yoo et al. (2007) and can be done in linear complexity w.r.t. the number of states and transitions. Thus, the total worst-case time complexity of the verification procedure is $O(|X^G|^{n+1}(|E^G| + n(|E^G| - |E_F|)))$. This is in fact the same computational complexity as reported in Moreira et al. (2011) and Qiu and Kumar (2006). However, we are considering transition-based codiagnosability here, which is more general than the static codiagnosability case considered in Moreira et al. (2011) and Qiu and Kumar (2006). In other words, when using the verifier approach, the verification of codiagnosability is not more computationally difficult for transition-based dynamic observations as it is with static observations.

4.3. Discussion and summary of results

Let $\mathcal{L}(G)$ be the language to be diagnosed. The preceding derivation of the upper bound for the maximal diagnosis delay is based on the assumption that there is one only type of fault, which was made for the sake of simplicity at the beginning of

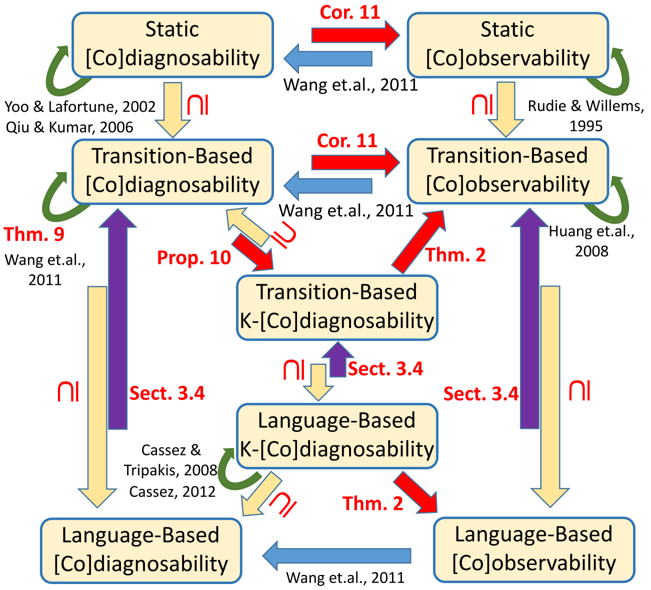


Fig. 6. The relationship between [K-][Co]diagnosability and [Co]observability.

Section 4. In the case of multiple fault types, a system is not codiagnosable if and only if it is not codiagnosable w.r.t. at least *one* type of faults. Moreover, given a certain type of faults, the system is not codiagnosable if and only if it is not $|X^G|^{n+1}$-codiagnosable. Therefore, this upper bound can also be applied to the case of multiple fault types. By using the upper bound $|X^G|^{n+1}$ to replace the integer K in $O(K^m|X^G||E^G|)$, which is the complexity of Algorithm KCOD–COOB-II, we consequently have the following result.

Corollary 11. *Let $\mathcal{L}(G)$ be the language to be diagnosed with $|\mathcal{I}| = n$ and $|\mathcal{F}| = m$. Transition-based codiagnosability can be transformed to transition-based coobservability in $O(|X^G|^{mn+m+1}|E^G|)$.*

We summarize the various transformations between the different notions of [K-][co]diagnosability and [co]observability that have been discussed so far in this paper in the diagram in Fig. 6. In this diagram, the arrows between blocks represent the transformations and the self-loop at each block represents the corresponding verification algorithm. As can be seen in the diagram, we cannot transform language-based codiagnosability directly to language-based coobservability. However, as discussed in Section 3.4, we can first transform language-based codiagnosability to transition-based codiagnosability, and then transform transition-based codiagnosability to transition-based coobservability by using Corollary 11, which is a special case of language-based coobservability.

Fig. 6 also highlights different verification algorithms for different notions. For a comprehensive survey on the computational complexity of these verification algorithms, the reader is referred to Sears and Rudie (2015). Since all the notions in the diagram can be related from one to the other, an interesting question that arises is: How to choose the "right" approach to verify a certain property? For instance, Ref. Huang et al. (2008) presents a verification algorithm of complexity of $O(|X|^6|E|)$ for transition-based coobservability with two agents ($|X|$ is the number of states and $|E|$ is the number of events of the system); however, the extension of this algorithm to the case of n agents is not available in the literature (Sears & Rudie, 2015). However, by applying the transformation algorithm in Wang et al. (2011) and the verification algorithm in Section 4, the verification of transition-based coobservability can be done in $O(|X|^3|E|^2)$, which may improve the computational complexity over the algorithm in Huang et al. (2008), since $|E|$ is usually much smaller than $|X|$. Moreover, this approach is applicable to the case of n agents.

Another implication of the relationships in the diagram in Fig. 6 is that we can solve a control (respectively, diagnosis) problem by applying the methodology for its corresponding diagnosis (respectively, control) problem. We discuss this issue in more detail through a specific problem in the next section.

5. Application to optimization of sensor activation

In this section, we show how the results in Section 3 can be used to leverage the research on observability and coobservability to solve problems pertaining to diagnosability and codiagnosability.

In sensor activation problems, under energy, bandwidth, or security constraints, 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 observational properties; see, e.g., Wang, Lafortune, Girard et al. (2010), Wang, Lafortune, Lin et al. (2010) and Sears and Rudie (2013a,b). Roughly speaking, sensor activation policies are a particular class of observation mappings satisfying the property that the sensor activation decisions for any two indistinguishable strings must be the same. This property is called the *feasibility* condition in Wang, Lafortune, Girard et al. (2010), Wang, Lafortune, Lin et al. (2010). It is formally defined as follows.

Definition 4. Given a system G, an observation mappings $\omega_i : \mathcal{L}(G) \to 2^{E_{o,i}}, i \in \mathcal{I}$, is said to be a feasible sensor activation policy if

$$(\forall \sigma \in E^G)(\forall s\sigma, t\sigma \in \mathcal{L}(G))$$
$$P_{\omega_i}(s) = P_{\omega_i}(t) \Rightarrow [\sigma \in \omega_i(s) \Leftrightarrow \sigma \in \omega_i(t)]. \tag{22}$$

The following theorem reveals that feasibility is preserved under transformation Algorithm KCOD–COOB-II of Section 3. In other words, any sensor activation policy synthesized for the transformed system can be applied back to the original system.

Theorem 12. *Let H be the input to Algorithm KCOD–COOB-II and consider output $\tilde{G}$. Then, for each agent $i \in \mathcal{I}$, 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 $\tilde{G}$.*

Proof. ($\Leftarrow$) By contrapositive. Since $\mathcal{L}(H) = \mathcal{L}(\hat{H}) \subseteq \mathcal{L}(\tilde{G})$ and $\forall s \in \mathcal{L}(H) : \omega_i(s) = \omega_{i,\tilde{G}}(s)$, ω_i is not feasible for H implies immediately that $\omega_{i,\tilde{G}}$ is not feasible for $\tilde{G}$ either.

($\Rightarrow$) Suppose ω_i is feasible for H. For any $\sigma \in E^{\tilde{G}}$, we know that either (i) $\sigma \in E^H$; or (ii) $\sigma \in \{c_j : j \in \mathcal{F}\}$. If $\sigma \in E^H$, then for any $s\sigma, t\sigma \in \mathcal{L}(\tilde{G})$, we know that $s, t \in \mathcal{L}(\hat{H}) = \mathcal{L}(H)$. Moreover, by the definition of $\omega_{i,\tilde{G}}$, we know that $\forall s \in \mathcal{L}(\hat{H}) = \mathcal{L}(H) : \omega_i(s) = \omega_{i,\tilde{G}}(s)$. Therefore, $P_{\omega_i}(s) = P_{\omega_i}(t) \Rightarrow [\sigma \in \omega_i(s) \Leftrightarrow \sigma \in \omega_i(t)]$ also implies that $P_{\omega_{i,\tilde{G}}}(s) = P_{\omega_{i,\tilde{G}}}(t) \Rightarrow [\sigma \in \omega_{i,\tilde{G}}(s) \Leftrightarrow \sigma \in \omega_{i,\tilde{G}}(t)]$. If $\sigma \in \{c_j : j \in \mathcal{F}\}$, then by the definition of $\omega_{i,\tilde{G}}$, we know that $\sigma \notin \omega_{i,\tilde{G}}(s), \forall s \in \mathcal{L}(\tilde{G})$. Therefore, it is always true that $P_{\omega_{i,\tilde{G}}}(s) = P_{\omega_{i,\tilde{G}}}(t) \Rightarrow [\sigma \in \omega_{i,\tilde{G}}(s) \Leftrightarrow \sigma \in \omega_{i,\tilde{G}}(t)]$. Thus, $\omega_{i,\tilde{G}}$ is also feasible for $\tilde{G}$. □

The above theorem provides an approach for the synthesis of optimal sensor activation policies for K-codiagnosability. The results in Cassez and Tripakis (2008) and Thorsley and Teneketzis (2007) only pertain to the centralized sensor activation problem while the algorithm developed in Wang, Lafortune, Girard et al. (2010) 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. (The verification of K-codiagnosability has been studied in Cassez, 2012.) This problem can now be solved by applying transformation Algorithm KCOD–COOB-II of Section 3 together with the algorithm in Wang, Lafortune, Lin et al. (2010). Suppose that H is the system to be diagnosed; $\tilde{H}$ and $\tilde{G}$ are the transformed systems produced by Algorithm KCOD–COOB-II. We can then apply Algorithm Min-Sen-Co in Wang, Lafortune, Lin et al. (2010) 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 $s \in \mathcal{L}(H)$.

6. Conclusion

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. Moreover, we have presented a new approach for the verification of transition-based codiagnosability. A new upper bound of the diagnosis delay for decentralized diagnosis under transition-based observations has been derived. We have shown that, when the observation properties are transition-based, [co]diagnosability is transformable to [co]observability. These new results complement those in Wang et al. (2011) that pertain to the reverse transformation, from coobservability to codiagnosability, thereby resulting in a thorough characterization of the relationship between the two notions of codiagnosability and coobservability and their verification, summarized in the diagram in Fig. 6. The new results in this paper also allow the leveraging of the large existing literature on solution methodologies for problems of decentralized control to solve corresponding problems of decentralized fault diagnosis. One such instance is the problem of optimal sensor activation for K-codiagnosability, for which we present the first solution procedure.

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful comments on improving this paper. They are also very grateful to David Sears, who pointed out a mistake in the T-Verifier in an earlier draft of this paper.

Appendix. Proofs not contained in main body

Proof of Lemma 4. We prove the case $i = 1$ by induction on the length of string t. If $|t| = 0$, i.e., $t = (\epsilon, \epsilon, \epsilon)$, then (18) holds trivially. Assume that (18) is true for $|t| = p$ and consider the string $t\sigma \in \mathcal{L}(V)$, $\sigma \in E^V$ with length $p + 1$. We have that

$$\begin{aligned} P_{\Omega_1}(\theta_0(t\sigma)) &= P_{\Omega_1}(\theta_0(t)\theta_0(\sigma)) \\ &= P_{\Omega_1}(\theta_0(t))I_1(\delta^G(\theta_0(t)), \theta_0(\sigma)) \end{aligned} \tag{A.1}$$

$$\begin{aligned} P_{\Omega_1}(\theta_1(t\sigma)) &= P_{\Omega_1}(\theta_1(t)\theta_1(\sigma)) \\ &= P_{\Omega_1}(\theta_1(t))I_1(\delta^G(\theta_1(t)), \theta_1(\sigma)) \end{aligned} \tag{A.2}$$

where, $I_1 : X^G \times E^G \to E^G \cup \{\epsilon\}$ is defined by $I_1(x, \sigma) = \sigma$ if $(x, \sigma) \in \Omega_1$ and $I_1(x, \sigma) = \epsilon$ if $(x, \sigma) \notin \Omega_1$.

If $\theta_0(\sigma) \in E_F$, then $\theta_1(\sigma) = \epsilon$ and we know that $I_1(\delta^G(\theta_0(t)), \theta_0(\sigma)) = I_1(\delta^G(\theta_1(t)), \theta_1(\sigma)) = \epsilon$. Therefore, we assume that $\theta_0(\sigma) \in E^G \setminus E_F$. By the definition of V, we know that one of the following cases is true: (i) $\theta_0(\sigma) = \theta_1(\sigma) \neq \epsilon$; or (ii) $\theta_0(\sigma) \neq \epsilon, \theta_1(\sigma) = \epsilon$; or (iii)$\theta_0(\sigma) = \epsilon, \theta_1(\sigma) \neq \epsilon$; or (iv) $\theta_0(\sigma) = \theta_1(\sigma) = \epsilon$. If (i) is true, then we know that A.1, A.3 or A.4 holds at $\delta^V(x_0^V, t)$ for $\theta_0(\sigma)$, which means that $I_1(\delta^G(\theta_0(t)), \theta_0(\sigma)) = I_1(\delta^G(\theta_1(t)), \theta_1(\sigma)) = \theta_0(\sigma)$. If (ii) is true, then we know that C.1, C.3, C.4, D.1, D.3 or D.4 holds at $\delta^V(x_0^V, t)$ for $\theta_0(\sigma)$. Under any of these cases, we have that $I_1(\delta^G(\theta_0(t)), \theta_0(\sigma)) = I_1(\delta^G(\theta_1(t)), \theta_1(\sigma)) = \epsilon$. Similarly, we can also verify that $I_1(\delta^G(\theta_0(t)), \theta_0(\sigma)) = I_1(\delta^G(\theta_1(t)), \theta_1(\sigma))$ still holds if (iii) or (iv) is true. Moreover, by the induction hypothesis, we know that $P_{\Omega_1}(\theta_0(t)) = P_{\Omega_1}(\theta_1(t))$. Thus, $P_{\Omega_1}(\theta_0(t\sigma)) = P_{\Omega_1}(\theta_1(t\sigma))$, which completes the induction. □

Proof of Lemma 5. We prove by induction on the length of s_0, s_1 and s_2. Let $s_i^p, i = 0, 1, 2$ denote the string that consists of the first p events in s_i and σ_i^p denote the $(p+1)^{th}$ event in s_i, so that $s_i^0 = \epsilon, s_i^1 = \sigma_i^0$, etc. Without loss of generality, we assume that $|s_0| = |s_1| = |s_2|$ and $P_{\Omega_i}(s_0^p) = P_{\Omega_i}(s_i^p), \forall p = 1, \ldots, |s_0|, i = 1, 2$, since these assumptions always hold if we consider ϵ as a single event. Initially, when $s_0^0 = s_1^0 = s_2^0 = \epsilon$, we can take $t = (\epsilon, \epsilon, \epsilon)$ such that $\theta_0(t) = s_i^0, \forall i = 0, 1, 2$. Assume that for $|s_0| = p$, there exists $t^p \in \mathcal{L}(V)$ such that $\theta_i(t^p) = s_i, \forall i = 0, 1, 2$. In order to show that the above assumption is also true for $|s_0| = p + 1$, i.e., $s_i = s_i^p \sigma_i^p, i = 0, 1, 2$, we need to consider the following cases:

(i) $\sigma_0^p \neq \epsilon, \sigma_1^p = \epsilon$ and $\sigma_2^p = \epsilon$
(ii) $\sigma_0^p = \epsilon, \sigma_1^p \neq \epsilon$ and $\sigma_2^p = \epsilon$
(iii) $\sigma_0^p = \epsilon, \sigma_1^p = \epsilon$ and $\sigma_2^p \neq \epsilon$
(iv) $\sigma_0^p \neq \epsilon, \sigma_1^p \neq \epsilon$ and $\sigma_2^p = \epsilon$
(v) $\sigma_0^p \neq \epsilon, \sigma_1^p = \epsilon$ and $\sigma_2^p \neq \epsilon$
(vi) $\sigma_0^p = \epsilon, \sigma_1^p \neq \epsilon$ and $\sigma_2^p \neq \epsilon$
(vii) $\sigma_0^p \neq \epsilon, \sigma_1^p \neq \epsilon$ and $\sigma_2^p \neq \epsilon$.

First, let us suppose that (i) is true. By $P_{\Omega_1}(s_0^p\sigma_0^p) = P_{\Omega_1}(s_1^p\epsilon)$ and $P_{\Omega_2}(s_0^p\sigma_0^p) = P_{\Omega_2}(s_2^p\epsilon)$, we know that $(\delta^G(x_0^G, s_0^p), \sigma_0^p) \notin \Omega_i, i = 1, 2$. Therefore, event $(\sigma_0^p, \epsilon, \epsilon)$ is defined at $\delta^V(x_0^V, t^p)$. By the induction hypothesis, we know that there exists $t^{p+1} = t^p(\sigma_0^p, \epsilon, \epsilon) \in \mathcal{L}(V)$ such that $\theta_i(t^{p+1}) = s_i^p\sigma_i^p, \forall i = 0, 1, 2$. Similarly, we can show that the induction step also holds for (ii) and (iii).

Suppose that (iv) is true. First, by $P_{\Omega_2}(s_0^p\sigma_0^p) = P_{\Omega_2}(s_2^p\epsilon)$, we know that $(\delta^G(x_0^G, s_0^p), \sigma_0^p) \notin \Omega_2$. Also, since $P_{\Omega_1}(s_0^p\sigma_0^p) = P_{\Omega_1}(s_1^p\sigma_1^p)$, we know that either (a) $(\delta^G(x_0^G, s_0^p), \sigma_0^p) \in \Omega_1$, $(\delta^{G_N}(x_0^{G_N}, s_1^p), \sigma_1^p) \in \Omega_1$ and $\sigma_0^p = \sigma_1^p$; or (b) $(\delta^G(x_0^G, s_0^p), \sigma_0^p) \notin \Omega_1$ and $(\delta^{G_N}(x_0^{G_N}, s_1^p), \sigma_1^p) \notin \Omega_1$. If (a) is true, we know that A.3 or A.4 holds, which implies that there exists $t^{p+1} = t^p(\sigma_0^p, \sigma_0^p, \epsilon) \in \mathcal{L}(V)$ such that $\theta_i(t^{p+1}) = s_i^p\sigma_i^p, \forall i = 0, 1, 2$. If (b) is true, we know that D.3 or D.4 holds. Therefore, there exists $t^{p+1} = t^p(\sigma_0^p, \epsilon, \epsilon)(\epsilon, \sigma_1^p, \epsilon) \in \mathcal{L}(V)$ or $t^{p+1} = t^p(\epsilon, \sigma_1^p, \epsilon)(\sigma_0^p, \epsilon, \epsilon) \in \mathcal{L}(V)$ such that $\theta_i(t^{p+1}) = s_i^p\sigma_i^p, \forall i = 0, 1, 2$. Similarly, we can show that the induction step holds for (v), (vi) and (vii).

References

Carvalho, L. K., Basilio, J. C., & Moreira, M. V. (2012). Robust diagnosis of discrete event systems against intermittent loss of observations. *Automatica*, *48*(9), 2068–2078.

Cassandras, C. G., & Lafortune, S. (2008). *Introduction to discrete event systems* (2nd ed.). Springer.

Cassez, F. (2012). The complexity of codiagnosability for discrete event and timed systems. *IEEE Transactions on Automatic Control*, *57*(7), 1752–1764.

Cassez, F., & Tripakis, S. (2008). Fault diagnosis with static and dynamic observers. *Fundamenta Informaticae*, *88*(4), 497–540.

Cieslak, R., Desclaux, C., Fawaz, A. S., & Varaiya, P. (1988). Supervisory control of discrete-event processes with partial observations. *IEEE Transactions on Automatic Control*, *33*(3), 249–260.

Debouk, R., Lafortune, S., & Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications*, *10*(1–2), 33–86.

Hashtrudi Zad, S., Kwong, R. H., & Wonham, W. M. (2003). Fault diagnosis in discrete-event systems: framework and model reduction. *IEEE Transactions on Automatic Control*, *48*(7), 1199–1212.

Huang, Y., Rudie, K., & Lin, F. (2008). Decentralized control of discrete-event systems when supervisors observe particular event occurrences. *IEEE Transactions on Automatic Control*, *53*(1), 384–388.

Jiang, S., Huang, Z., Chandra, V., & Kumar, R. (2001). A polynomial algorithm for testing diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *46*(8), 1318–1321.

Lin, F. (1994). Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems: Theory and Applications*, *4*(2), 197–212.

Lin, F. (2014). Control of networked discrete event systems: dealing with communication delays and losses. *SIAM Journal on Control and Optimization*, *52*(2), 1276–1298.

Lin, F., & Wonham, W. M. (1988). On observability of discrete-event systems. *Information Sciences*, *44*(3), 173–198.

Moreira, M. V., Jesus, T. C., & Basilio, J. C. (2011). Polynomial time verification of decentralized diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, *56*(7), 1679–1684.

Overkamp, A., & van Schuppen, J. H. (2000). Maximal solutions in decentralized supervisory control. *SIAM Journal on Control and Optimization*, *39*(2), 492–511.

Pencolé, Y., & Cordier, M. (2005). A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artificial Intelligence*, *164*(1), 121–170.

Qiu, W., & Kumar, R. (2006). Decentralized failure diagnosis of discrete event systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, *36*(2), 384–395.

Rudie, K., Lafortune, S., & Lin, F. (2003). Minimal communication in a distributed discrete-event system. *IEEE Transactions on Automatic Control*, *48*(6), 957–975.

Rudie, K., & Willems, J. C. (1995). The computational complexity of decentralized discrete-event control problems. *IEEE Transactions on Automatic Control*, *40*(7), 1313–1319.

Rudie, K., & Wonham, W. M. (1992). Think globally, act locally: Decentralized supervisory control. *IEEE Transactions on Automatic Control*, *37*(11), 1692–1708.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *40*(9), 1555–1575.

Sears, D., & Rudie, K. (2013a). Computing sensor activation decisions from state equivalence classes in discrete-event systems. In *52nd IEEE conference on decision and control* (pp. 6972–6977).

Sears, D., & Rudie, K. (2013b). Efficient computation of sensor activation decisions in discrete-event systems. In *52nd IEEE conference on decision and control* (pp. 6966–6971).

Sears, D., & Rudie, K. (2015). Minimal sensor activation and minimal communication in discrete-event systems. *Discrete Event Dynamic Systems: Theory and Applications*,.

Seatzu, C., Silva, M., & Van Schuppen, J. H. (2013). *Control of discrete-event systems. automata and petri net perspectives*. London: Springer.

Su, R., & Wonham, W. M. (2005). Global and local consistencies in distributed fault diagnosis for discrete-event systems. *IEEE Transactions on Automatic Control*, *50*(12), 1923–1935.

Thorsley, D., & Teneketzis, D. (2007). Active acquisition of information for diagnosis and supervisory control of discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications*, *17*(4), 531–583.

Tripakis, S. (2004). Undecidable problems of decentralized observation and control on regular languages. *Information Processing Letters*, *90*(1), 21–28.

Wang, W., Girard, A. R., Lafortune, S., & Lin, F. (2011). On codiagnosability and coobservability with dynamic observations. *IEEE Transactions on Automatic Control*, *56*(7), 1551–1566.

Wang, W., Lafortune, S., Girard, A. R., & Lin, F. (2010). Optimal sensor activation for diagnosing discrete event systems. *Automatica*, *46*(7), 1165–1175.

Wang, W., Lafortune, S., & Lin, F. (2007). An algorithm for calculating indistinguishable states and clusters in finite-state automata with partially observable transitions. *Systems & Control Letters*, *56*(9), 656–661.

Wang, W., Lafortune, S., Lin, F., & Girard, A. R. (2010). Minimization of dynamic sensor activation in discrete event systems for the purpose of control. *IEEE Transactions on Automatic Control*, *55*(11), 2447–2461.

Wang, Y., Yoo, T.-S., & Lafortune, S. (2007). Diagnosis of discrete event systems using decentralized architectures. *Discrete Event Dynamic Systems: Theory and Applications*, *17*(2), 233–263.

Yin, X., & Lafortune, S. (2015). On the relationship between codiagnosability and coobservability under dynamic observations. In *American control conference* (pp. 390–395).

Yoo, T.-S., & Garcia, H. E. (2009). Event counting of partially-observed discrete-event systems with uniformly and nonuniformly bounded diagnosis delays. *Discrete Event Dynamic Systems: Theory and Applications*, *19*(2), 167–187.

Yoo, T.-S., & Lafortune, S. (2002). Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Transactions on Automatic Control*, *47*(9), 1491–1495.

Zaytoon, J., & Lafortune, S. (2013). Overview of fault diagnosis methods for discrete event systems. *Annual Reviews in Control*, *37*(2), 308–320.

Zaytoon, J., & Sayed-Mouchaweh, M. (2012). Discussion on fault diagnosis methods of discrete event systems. In *Proceedings of the 11th international workshop on discrete event systems* (pp. 9–12).

Xiang Yin was born in Anhui, China, in 1991. He received the B.Eng. degree from Zhejiang University in 2012 and the M.S. degree from the University of Michigan, Ann Arbor, in 2013, both in Electrical Engineering. He is currently a Ph.D. candidate in the Electrical Engineering: System program at the University of Michigan, Ann Arbor. His research interests include supervisory control of discrete-event systems, model-based fault diagnosis, formal methods, game theory and their applications to cyber and cyber–physical systems.

Stéphane Lafortune received the B.Eng. degree from Ecole Polytechnique de Montréal in 1980, the M.Eng. degree from McGill University in 1982, and the Ph.D. degree from the University of California at Berkeley in 1986, all in Electrical Engineering. Since September 1986, he has been with the University of Michigan, Ann Arbor, where he is a Professor of Electrical Engineering and Computer Science. He is a Fellow of the IEEE (1999). He received the Presidential Young Investigator Award from the National Science Foundation in 1990 and the George S. Axelby Outstanding Paper Award from the Control Systems Society of the IEEE in 1994 (for a paper co-authored with S.-L. Chung and F. Lin) and in 2001 (for a paper co-authored with G. Barrett). His research interests are in discrete event systems and include multiple problem domains: modeling, diagnosis, control, optimization, and applications to computer and software systems. He is the lead developer of the software package UMDES and co-developer of DESUMA with L. Ricker. He co-authored, with C. Cassandras, the textbook Introduction to Discrete Event Systems—Second Edition (Springer, 2008). He is Editor-in-Chief of the Journal of Discrete Event Dynamic Systems: Theory and Applications.