2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Robust codiagnosability of discrete event systems

João Carlos Basilio and Stéphane Lafortune

Abstract— **We consider robust decentralized diagnosis of discrete event systems, where the goal is to detect the occurrence of unobservable fault events using a set of local diagnosers that are themselves subject to failures. We introduce a formal notion of robust decentralized diagnosability, called robust codiagnosability, and study its properties. Two different tests of robust codiagnosability are presented; one uses diagnoser automata and the other uses verifier automata. We also revisit the problem of centralized diagnosability and study the problem of diagnosability under partial observation, where the set of observable events is reduced; in this regard, we introduce the notions of partial diagnosers and indeterminate hidden cycles, which are subsequently used in the study of robust codiagnosability.**

I. INTRODUCTION

We are interested in the problem of fault diagnosis in decentralized discrete event systems (DES), in the presence of unreliable communications or unreliable diagnostic engines at local sites. Our primary focus is the decentralized architecture for diagnosis considered in [1]. Several extensions of this work, involving refined local diagnostic engines that issue conditional decisions or perform multiple levels of inferencing, have been proposed lately; see, e.g., [2], [3], [4]. In this work, we pursue another direction. Namely, we are interested in the robustness properties of the architecture in [1] when local diagnostic engines go down due to faults in communication or at sites. Such faulty behavior is common in application areas where DES techniques for fault diagnosis have been employed: automated transportation systems [5], nuclear systems [6], or software systems [7].

Robustness is an important property of diagnostic systems. It has been studied extensively in the context of continuous-variable time-driven systems (see [8], [9] and the references therein). Despite the large body of work on diagnosis of DES in centralized and decentralized architectures, few works, if any, have explicitly addressed robustness of the diagnostic system. References [10], [6], [11] consider the problem of event diagnosis under unreliable sensors and use stochastic models for this purpose. The problem of fault-tolerant control of DES, where diagnostic methodologies are embedded in control architectures, has been considered in [12], [13]. Reliability of local controllers in decentralized control architectures (without diagnosis) has been considered in [14].

In this paper, we consider a decentralized architecture where the local sites do not communicate with one another and no explicit coordination among sites is necessary. This is the situation corresponding to Protocol 3 in [1] and is often referred to as *codiagnosability* in recent work [2], [4]. Note that several works consider architectures with communicating diagnostic engines, e.g., [15], [16], [17], [18], [19], where a variety of modeling formalisms are adopted: automata, communicating automata, Petri nets, and related models. Codiagnosability refers to the situation where it is required that each fault be diagnosed by at least one local site, when all local sites operate autonomously by processing their local observations. We propose in this paper a formal definition for a notion of *robust codiganosability* and study its properties. A decentralized diagnosis strategy is robust when it can withstand the permanent failure of one or more sites. We do not specify why a site becomes unavailable; it may be due to a hardware or software fault at the site, or to a fault in the communication channel with the site. Two methods of testing for codiagnosability are presented, one based on diagnoser automata and one based on verifier automata. We also discuss how to use diagnoser automata online for robust codiagnosis. These results are presented in Section IV.

For the purpose of establishing our results on robust codiagnosability, we develop new results on centralized diagnosability when the set of observable events is reduced. We call this problem "centralized diagnosis under partial observation" and refer to the associated diagnosers as "partial diagnosers." This problem is treated in Section III, where we introduce the new notion of *hidden cycle* and use it to characterize diagnosability under partial observation. Hidden cycles and partial diagnosers play a role in the study of robust codiagnosability in Section IV.

The next section reviews necessary results and establishes the notation used in the sequel.

II. PRELIMINARIES

A. Fault diagnosis of discrete event systems

Let

$$G = (X, \Sigma, f, \Gamma, x_0, X_m) \tag{1}$$

be a deterministic automaton, where X is the state space, Σ is the set of events, f is the partial transition function, Γ is the active event function, x_0 is the initial state of the system, and $X_m \subseteq X$ is the set of marked states. In addition, assume

This work was carried out when the first author was on a sabbatical leave at the University of Michigan. The research of J. C. Basilio was partially supported by the Brazilian Research Council (CNPq) grant 200820/2006-0. The research of S. Lafortune was partially supported by NSF grants CCR-0325571 and EECS-0624821

J. C. Basilio is with COPPE - Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, 21949-900, Rio de Janeiro, RJ, Brazil. basilio@dee.ufrj.br

S. Lafortune is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA. stephane@eecs.umich.edu

that the set of events Σ is partitioned into two subsets: Σ_o, the set of observable events, *i.e.* the set of events whose occurrence can be observed, and Σ_{uo}, the set of unobservable events. The unobservable events of the system are those whose occurrence cannot be recorded by sensors, together with the fault events. Therefore, model G accounts for the normal and failed behavior of the system. Let $\Sigma_f \subseteq \Sigma_{uo}$ denote the set of fault events.

The fault diagnosis problem can be stated as follows: Given that a fault event has occurred, identify its occurrence, assuming that, in the traces generated by G, only the events in Σ_o are observed. The set of fault events Σ_f is usually partitioned into different subsets Σ_{f_i}, $i = 1, 2, \ldots, m$, not necessarily singleton sets, so that each set Σ_{f_i} accounts for specific fault types; the reader is referred to [20], [21], [22] for further details. Let $\Pi_f = \{\Sigma_{f_1}, \Sigma_{f_2}, \ldots, \Sigma_{f_m}\}$ denote this partition. Then, every time it is stated that a fault of type F_i has occurred, it should be understood that some event from the set Σ_{f_i} has occurred.

Let L denote the language generated by G. Three assumptions are made in this paper:

A1. The language generated by G is live, *i.e.* $\Gamma(x_i) \neq \emptyset$ for all $x_i \in X$;

A2. Automaton G has no cycle of unobservable events, *i.e.* $\exists n_0 \in \mathbb{N} : \forall ust \in L,\ s \in \Sigma_{uo}^*,\ u, t \in \Sigma_o \Rightarrow \|s\| \leq n_0$, where $\|s\|$ denotes the length of trace s.

A3. There is only one fault type *i.e.* $\Sigma_f = \{\sigma_f\}$.

Assumptions A1 and A2 are standard in the literature. Assumption A1 is made for the sake of simplicity and can be relaxed at the cost of additional technical details. Assumption A2 is a necessary condition for diagnosability of faults, since it precludes the existence of arbitrarily long traces of unobservable events. Assumption A3 is made in this paper to keep the notation and treatment simpler. It is without loss of generality as one can always diagnose each fault type separately.

B. Notation

The notation used throughout the paper is the usual one [23]. The post-language of L after s is denoted by L/s, and is defined as

$$L/s = \{t \in \Sigma^* : st \in L\}. \tag{2}$$

The language projection operation P_o is defined in the usual manner [24], as

$$\begin{array}{rcccl} P_o & : & \Sigma^* & \rightarrow & \Sigma_o^* \\ & & s & \mapsto & P_o(s), \end{array} \tag{3}$$

with the following properties:

$$\begin{array}{rcl} P_o(\epsilon) & = & \epsilon, \\ P_o(\sigma) & = & \left\{ \begin{array}{l} \sigma, \text{ if } \sigma \in \Sigma_o, \\ \epsilon, \text{ if } \sigma \in \Sigma_{uo}, \end{array} \right. \\ P_o(s\sigma) & = & P(s)P(\sigma), s \in \Sigma^*, \sigma \in \Sigma, \end{array} \tag{4}$$

where ϵ denotes the empty trace. The inverse projection operator over language L, denoted by $P_{o_L}^{-1}$, is defined as

$$P_{o_L}^{-1} = \{s \in L : P(s) = y\}. \tag{5}$$

The set of all traces that end with the fault event σ_f is denoted as $\Psi(\Sigma_f)$. Formally,

$$\Psi(\Sigma_f) = \{s \in L : s_f \in \Sigma_f\}, \tag{6}$$

where s_f denotes the last event of s. With slight abuse of notation, given a trace s, the membership relation $\Sigma_f \in s$ can be used to denote that $\overline{s} \cap \Psi(\Sigma_f) \neq \emptyset$, where $\overline{s}$ denote the prefix-closure of s. Finally, a trace $s \in L$ is a faulty trace if $\Sigma_f \in s$.

C. Centralized diagnosis of DES

In words, the language generated by an automaton is diagnosable with respect to a set of observable events and a failure set Σ_f if the occurrence of any fault in Σ_f can be detected, within a finite delay, using only traces of observable events. In a formal way, diagnosability is defined as follows [20].

Definition 1: A prefix-closed and live language L is diagnosable with respect to projection P_o and $\Sigma_f = \{\sigma_f\}$ if the following holds true:

$$(\exists n \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall t \in L/s)(\|t\| \geq n \Rightarrow D),$$

where the diagnosability condition D is

$$(\forall \omega \in P_{o_L}^{-1}(P(st)))[\Sigma_f \in \omega].$$

□

Diagnosability analysis is carried out using either a deterministic automaton called diagnoser or a nondeterministic automaton called verifier [25]. The main advantage of verifiers over diagnosers is that the diagnosability test using verifiers requires polynomial time in the cardinality of the state space of the system model; testing diagnosability using diagnosers can be performed in polynomial time in the cardinality of the state space of the diagnoser; however, this state space is, in the worst case, exponential in the cardinality of the state space of the system model. On the other hand, diagnosers can also be used to perform online diagnosis through the observation of the occurrences of observable events of the system, since they provide information not only on the possible states where the system can be after the occurrence of an observable event, but also on fault occurrence. The diagnoser for G, hereafter denoted as G_d, is a deterministic automaton with labels Y and N attached to the states of G, in the states of G_d, to indicate whether event σ_f has occurred or not; specifically, they are of the form (x, Y) or (x, N), depending on whether or not σ_f is present in the traces that take x_0 to x. Formally, G_d is defined as

$$G_d = (X_d, \Sigma_o, f_d, \Gamma_d, x_{0_d}). \tag{7}$$

The construction of G_d was initially presented in [20]. An equivalent approach (see [23]) is to proceed in two steps, as follows: *(i)* perform the parallel computation $G \| A_l$, where $A_l = (X_l, \Sigma_l, f_l, x_{0_l})$ is a two-state label automaton, with $X_l = \{N, F\}$, $\Sigma = \{\sigma_f\}$, $x_{0_l} = N$, $f_l(x_{0_l}, \sigma_f) = x_1 = F$ and $f_l(x_1, \sigma_f) = x_1$ and; *(ii)* compute *Obs*$(G \| A_l)$,

where *Obs* is the deterministic observer automaton, whose computation is presented in [23, p. 103](see also [26]).

It is not difficult to see that the language generated by G_d is equal to $P_o(L)$ and that $X_d \subset 2^{X\times\{N,Y\}}$. As far as the presence of Y and N labels in the states of G_d are concerned, the states $x_d \in X_d$ are defined as certain (or faulty), normal or uncertain [20]. A state x_d is called certain (or faulty), if $\ell = Y$ for all $(x,\ell) \in x_d$, and normal (or non-faulty) if $\ell = N$ for all $(x,\ell) \in x_d$. If there exist $(x,\ell),(y,\tilde{\ell}) \in x_d$, x not necessarily distinct from y such that $\ell = Y$ and $\tilde{\ell} = N$, then x_d is called an uncertain state of G_d. It can be seen from the diagnoser construction [20] that once the diagnoser becomes certain about fault occurrence, i.e., reaches a certain state, it is not possible for it to become uncertain again.

Let $L(G,x)$ denote the set of all traces that originate in state x of G. Then

1) A set of states $x_1, x_2, \ldots, x_n \in X$ forms a cycle in G if there exists a trace $s = \sigma_1\sigma_2\ldots\sigma_n \in L(G,x_1)$ such that $f(x_l,\sigma_l) = x_{l+1}$, $l = 1,\ldots,n-1$, and $f(x_n,\sigma_n) = x_1$.

2) A set of uncertain states $x_{d_1}, x_{d_2}, \ldots, x_{d_n} \in X_d$ forms an indeterminate cycle if the following conditions hold true:

2.1) $x_{d_1}, x_{d_2}, \ldots, x_{d_n}$ form a cycle in G_d, *i.e.* there exist $\sigma_l \in \Sigma_o$, $l = 1,2,\ldots,n$, such that $f_d(x_{d_l},\sigma_l) = x_{d_{l+1}}$, $l = 1,2,\ldots,n-1$, and $f_d(x_{d_n},\sigma_n) = x_{d_1}$;

2.2) $\exists (x_l^{k_l},\ell_l^{k_l}),(\tilde{x}_l^{r_l},\tilde{\ell}_l^{r_l}) \in x_{d_l}$, $x_l^{k_l}$ not necessarily distinct from $\tilde{x}_l^{r_l}$, $l = 1,2,\ldots,n$, $k_l = 1,2,\ldots,m_l$, and $r_l = 1,2,\ldots,\tilde{m}_l$ such that

a) $Y \in \ell_l^{k_l}$, $Y \notin \tilde{\ell}_l^{r_l}$, for all l, k and r;

b) The sequences of states $\{x_l^{k_l}\}$, $l = 1,2,\ldots,n$, $k_l = 1,2,\ldots,m_l$ and $\{\tilde{x}_l^{r_l}\}$, $l = 1,2,\ldots,n$, $r_l = 1,2,\ldots,\tilde{m}_l$ form cycles in G, such that the corresponding traces s and $\tilde{s}$, formed with the events that define the evolution of the cycles, have as projection $\sigma_1\sigma_2\ldots\sigma_n$, where σ_1, σ_2, and σ_n are defined in 1).

In addition, the following relationship between the traces of the language generated by G and the states of G_d has been established [20].

Lemma 1:

(i) Let $x_d = f_d(x_{0_d}, s)$. If x_d is a certain state, then for all $\omega \in P_{o_L}^{-1}(s)$, $\Sigma_f \in \omega$.

(ii) If x_d is an uncertain state, then there exist $s_1, s_2 \in L$ such that $\Sigma_f \in s_1$ and $\Sigma_f \notin s_2$, but $P_o(s_1) = P_o(s_2)$ and $f_d[x_{0_d}, P(s_1)] = f_d[x_{0_d}, P(s_2)] = x_d$. □

The following necessary and sufficient condition for language diagnosis can be stated [20].

Theorem 1: A language L generated by an automaton G is diagnosable with respect to projection P_o and $\Sigma_f = \{\sigma_f\}$ if, and only if, its diagnoser G_d has no indeterminate cycles. □

III. CENTRALIZED DIAGNOSABILITY UNDER PARTIAL OBSERVATION

The dependence of language diagnosability on the set of observable events suggests that it may be possible that the language generated by an automaton can also be diagnosable with respect to another projection $P_o' : \Sigma^* \to \Sigma_o'^*$, where $\Sigma_o' \subset \Sigma_o$. This problem is known as centralized diagnosability under partial observation, and in order to address it, besides Assumptions A1–A3, the following assumption is also made:

A4. L is diagnosable with respect to projection $P_o : \Sigma^* \to \Sigma_o^*$ and Σ_f (centralized diagnosable).

Let $G_d' = (X_d', \Sigma_o', f_d', \Gamma_d', x_{0_d}')$ denote the diagnoser for L assuming partial observation, *i.e.* G_d' is capable of observing only events in a set $\Sigma_o' \subset \Sigma_o$. For this reason, G_d' will be referred to as a centralized diagnoser with partial observation or simply partial diagnoser. The following result can be stated.

Theorem 2: Let $G_d = (X_d, \Sigma_o, f_d, \Gamma_d, x_{0_d})$ and $G_d' = (X_d', \Sigma_o', f_d', \Gamma_d', x_{0_d}')$ denote centralized diagnosers, assuming, respectively, full and partial observation, *i.e.* $\Sigma_o' \subset \Sigma_o$ and $\Sigma_o' \neq \emptyset$. Then, $Obs(G_d, \Sigma_o') = (\hat{X}_d, \Sigma_o', \hat{f}_d, \hat{\Gamma}_d, \hat{x}_{0_d})$ (the observer of G_d with respect to projection $P_o' : \Sigma_o^* \to \Sigma_o'^*$) and G_d' are equal up to the following renaming of states:

$$\hat{x}_d = \{x_{d_1}, x_{d_2}, \ldots, x_{d_n}\} \in \hat{X}_d,\ x_{d_i} \in X_d \Leftrightarrow x_d' = \bigcup_{i=1}^{n} x_{d_i} \in X_d'.$$

Proof: The proof is omitted here; it can be found in [26]. □

According to Theorem 2, the partial diagnoser G_d' that observes the events in a subset Σ_o' of the set of observable events Σ_o can be built from G_d in a straightforward way as follows:

1) The states of G_d' are obtained by merging all the states of G_d that are connected by the events in $\Sigma_o \setminus \Sigma_o'$ into a single state formed by the union of the sets of the states merged;

2) The transition function for each state x_d' defined in the previous step is defined as

$$f_d'(x_d', e) = \bigcup_{x_d \text{ merged and } e\in\Sigma_o'\cap\Gamma_d(x_d)} f_d(x_d, e),$$

where x_d *merged* denotes all states that have been merged to form the new state x_d'.

Another contribution of Theorem 2 is that, although the language generated by the centralized diagnoser with full observation is always live (due to Assumption A2), the languages generated by partial diagnosers are not necessarily live. This happens whenever the events that form a cycle in G_d become unobservable in the partial diagnoser. It is not difficult to see that when this happens, this cycle reduces to a single state in G_d'. When unobservable events occur after the system reaches such a particular state, even though there is no change of state in the partial diagnoser, the actual states of the automaton change cyclically. In this case, it is said that this particular partial diagnoser has a hidden cycle, whose formal definition is as follows.

Definition 2: (Hidden cycle and indeterminate hidden cycle) Let $x_d' \in X_d'$ be obtained by merging states $x_{d_1}, x_{d_2}, \ldots, x_{d_n} \in X_d$. Then, there exists a hidden cycle in x_d' in G_d' if, for some $\{i_1, i_2, \ldots, i_k\} \subset \{1,2,\ldots,n\}$, $x_{d_{i_1}}, x_{d_{i_2}}, \ldots, x_{d_{i_k}}$ form a cycle in G_d. Moreover, if x_d' is uncertain and all states $x_{d_{i_1}}, x_{d_{i_2}}, \ldots, x_{d_{i_k}}$ are certain, then the hidden cycle is indeterminate. □

Due to Assumption A4, it is not difficult to see that states x_{d_k}, $k = 1, 2, \ldots, n$, that form the cycle that is hidden in x'_d, must all be either faulty or normal. For this reason, hidden cycles will be represented in the state transition diagrams of partial diagnosers by dashed self-loops: indeterminate hidden cycles will be labeled as *ihc* and hidden cycles in normal or certain states will be labeled simply as *hc*, since, as it will be seen in the sequel, they do not interfere in diagnosability under partial observation.

According to the diagnosability condition given in Definition 1, all traces $s \in \Psi(\Sigma_f)$ must be diagnosed by the partial diagnoser G'_d. The following theorem provides a necessary and sufficient condition for diagnosability under partial observation.

Theorem 3: Assuming that a language L is diagnosable with respect to projection P_o and Σ_f, then L will be also diagnosable with respect to projection $P'_o : \Sigma^* \to \Sigma'^*_o$, $\Sigma'_o \subset \Sigma_o$, and $\Sigma_f = \{\sigma_f\}$ if, and only if, G'_d has no indeterminate cycles (hidden cycles included).

Proof: A necessary and sufficient condition for diagnosability has been established by Theorem 1 when G'_d has no hidden cycles. It remains to address the case when G'_d has indeterminate hidden cycles.

Let $UR(x, \Sigma_{uo})$ denote the unobservable reach of x with respect to the unobservable set Σ_{uo} [23, p. 102]. For some $x_{d_{unc}} \in X_d$, define $x'_{d_{unc}} = UR(x_{d_{unc}}, \Sigma_o \setminus \Sigma'_o) \in X'_d$, and assume that, for $l = 1, \ldots, n$, states $x^{(l)}_{d_{cert}} \in x'_{d_{unc}}$ form an indeterminate hidden cycle in $x'_{d_{unc}}$. It is not hard to see that there exists a trace $stu_k \in L$ that satisfies the following conditions:

1) $s \in \Psi(\Sigma_f)$ and $f_d(x_{0_d}, P_o(s)) = x_{d_{unc}}$;

2) $t \in (\Sigma_o \setminus \Sigma'_o)^*$ is such that $f_d(x_{0_d}, P_o(st)) = x^{(1)}_{d_{cert}}$;

3) $u_k \in (\Sigma_o \setminus \Sigma'_o)^*$, $\|u_k\| = k$, where k can be arbitrarily large, is such that $f_d(x^{(1)}_{d_{cert}}, u_k) = x^{((k \bmod n)+1)}_{d_{cert}}$.

It is immediate that $f'_d(x_{0_d}, P'_o(stu_k)) = x'_{d_{unc}}$, which implies that, since $f_d(x_{0_d}, P_o(s)) = x_{d_{unc}}$, then, according to item (ii) of Lemma 1, there exists $w \in P'^{-1}_{o_L}(stu_k)$ such that $\Sigma_f \notin w$, which violates the diagnosability condition. □

Example 1: In order to illustrate the results presented in this section, consider automaton $G = (X, \Sigma, f, \Gamma, x_0, X_m)$ depicted in Figure 1(a), where $\Sigma = \Sigma_o \cup \Sigma_{uo}$, $\Sigma_o = \{a, b, c\}$ and $\Sigma_{uo} = \Sigma_f = \{\sigma_f\}$. The corresponding diagnoser is shown in Figure 1(b), on the left, and it is clear that the language L generated by G is diagnosable with respect to projection P_o and Σ_f. Consider now the partial diagnoser G'_d shown in Figure 1(b), in the middle, whose set of observable events is $\Sigma'_o = \{a, c\}$. Notice that, since there is an indeterminate cycle in state $\{1N, 2Y, 3N\}$, then L is not diagnosable with respect to P'_o and Σ_f. The same conclusion can be drawn when the set of observable events is $\Sigma''_o = \{b, d\}$. This is so because the partial diagnoser G''_d, shown in Figure 1(b), on the right, has an indeterminate hidden cycle in state $\{1N, 2Y, 3N\}$ due to unobservability of event c. It is important to point out that G''_d has two other hidden cycles, one in state $\{1N, 2Y, 3N\}$ and the other in state $\{3Y\}$, both due to event c. □

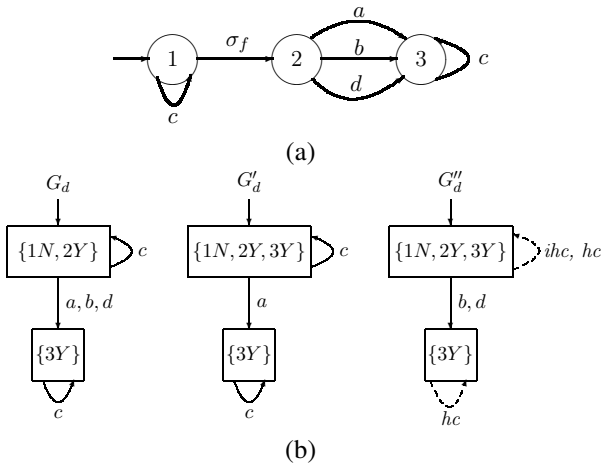

Fig. 1. Automaton G (a) and its centralized diagnoser G_d (b) for example 1.

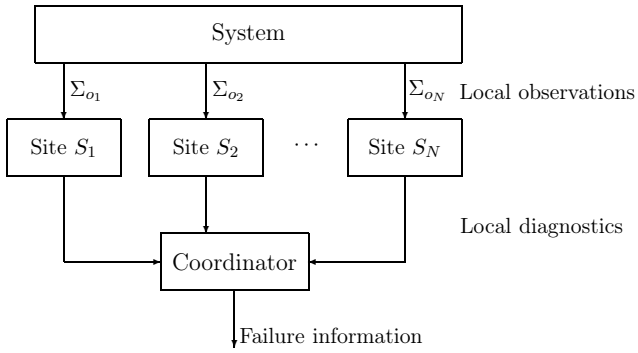

Fig. 2. Coordinated decentralized architecture.

IV. ROBUST CODIAGNOSABILITY

A. Codiagnosability of DES

In practice, due to the distributed nature of some systems, centralized diagnosers cannot always be employed. In order to circumvent this problem, the decentralized architecture depicted in Fig. 2 has been proposed [1]. In this decentralized architecture, sensor readings are no longer centralized, but distributed over different sites S_i, $i = 1, 2, \ldots, n$, each site observing the system behavior based on its available sensing capabilities, or equivalently, on the set of observable events $\Sigma_{o,i}$, $i = 1, 2, \ldots, n$. Each site processes the information received (event occurrences), and, in the decentralized architecture proposed in [1], the sites are only allowed to communicate their diagnostic to a coordinator, which processes this information according to a prescribed rule and takes a decision on the fault occurrence. It is worth remarking that the coordinator has no knowledge of the system model, and is supposed to have limited memory and processing capabilities.

The definition of diagnosability for the coordinated decentralized architecture shown in Fig. 2 depends on the fault event set Σ_f and also on four elements: *(i)* the rules used to generate local diagnostics; *(ii)* the communication rules between sites and coordinator; *(iii)* the fault diagnosis decision rules employed by the coordinator and; *(iv)* the

projections

$$P_{o,i} : \Sigma \to \Sigma_{o,i}, i = 1, 2, \ldots, n,$$

associated with each site S_i. The first three elements are usually referred to in the literature as a *protocol*.

Let C denote the coordinator diagnostic information, and note that, for each path of the system, C is represented by an information set that is protocol-dependent. Then the coordinator diagnostic information is said to be certain if, based on C, the coordinator is certain that a fault in Σ_f has occurred. Therefore, Definition 1 can be changed to accommodate coordinated decentralized systems as follows [1].

Definition 3: A prefix-closed and live language L is said to be diagnosable under a protocol, a set of projections $P_{o,i}$, $i = 1, \ldots, n$, and $\Sigma_f = \{\sigma_f\}$ if the following holds true

$$(\exists n \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall t \in L/s)(\|t\| \geq n \Rightarrow C \text{ is certain}).$$

□

As in the case of centralized diagnosis, the detection of any fault should be achieved by the coordinator within a finite delay of its occurrence.

Three protocols are proposed in [1], and necessary and sufficient conditions for diagnosability with respect to each protocol are presented. In particular, for Protocol 3, a partial diagnoser is implemented in each site, whose state, after the occurrence of an observable event, is the diagnostic information, based on which, the site is supposed to infer the occurrence of faults. When one site observes an event that leads to a certain state in its diagnoser, it communicates the fault occurrence to the coordinator. The coordinator declares the occurrence of a fault whenever at least one site communicates a fault occurrence, and remains silent if there is no report on fault occurrence. It is clear that protocol 3 can be regarded as an extension of centralized diagnosis to coordinated decentralized diagnosis, and, therefore, from now on, diagnosability under Protocol 3 will be referred to as *codiagnosability* and the diagnosers at each site will be called partial diagnosers.

In dealing with codiagnosability, besides Assumptions A1 to A3, the following assumptions are also made:

A5. L is not diagnosable with respect to $P_{o,i}$, $i = 1, 2, \ldots, n$.

A6. There is a reliable communication between the local sites and the coordinator, *i.e.*, all messages sent from a local site are received by the coordinator correctly and in order.

Assumption A5 precludes the trivial case when one site performs as the centralized diagnoser, and Assumption A6 will be removed later in this work. It is worth remarking that assumption A2 of [1] (namely that G has no cycle of unobservable events with respect to $\Sigma_{o,i}$, $i = 1, 2, \ldots, n$) has been removed here. As seen in section III, this may lead to hidden cycles.

B. Analysis of robust codiagnosability using diagnosers

The idea behind codiagnosis is that all traces in $s \in \Psi(\Sigma_f)$ must be diagnosed by at least one partial diagnoser. A trace $s \in \Psi(\Sigma_f)$ that is not diagnosed by any partial diagnoser is called a fully-ambiguous trace. The formal definition of fully-ambiguous trace is given below [1].

Definition 4: A trace $s \in L$ is said to be fully-ambiguous with respect to projections $P_{o,i}$, $i = 1, 2, \ldots, n$, and σ_f, if there exist n arbitrarily long traces $s_1, s_2, \ldots, s_n \in L$, not necessarily distinct, such that

1) $\Sigma_f \in s$ but $\Sigma_f \notin s_i$, $i = 1, 2, \ldots, n$;

2) $P_{o,i}(s) = P_{o,i}(s_i)$, $i = 1, 2, \ldots, n$. □

Therefore, in order to verify whether L is codiagnosable or not with respect to projections $P_{o,i}$, $i = 1, 2, \ldots, n$ and $\Sigma_f = \{\sigma_f\}$, it is enough to identify the existence of fully-ambiguous traces. In order to derive a test for detecting whether fully-ambiguous traces exist or not, let $G_{d_i} = (X_{d_i}, \Sigma_{o,i}, f_{d_i}, \Gamma_{d_i}, x_{0_{d_i}})$ denote the partial diagnoser for site S_i, $i = 1, 2, \ldots, n$, and let G_d denote the centralized diagnoser. Consider the diagnoser G_{test_n} defined as follows:

$$G_{\text{test}_n} = (\|_{i=1}^{n} G_{d_i}) \| G_d. \tag{8}$$

It is not hard to see that

$$L(G_{\text{test}_n}) = \{\bigcap_{i=1}^{n} P_i^{-1}[L(G_{d_i})]\} \bigcap L(G_d),$$

where P_i^{-1}, $i = 1, 2, \ldots, n$ is with respect to Σ_o and not Σ. Therefore,

$$L(G_{\text{test}_n}) = L(G_d), \tag{9}$$

which shows that G_{test_n} provides the means to identify the current state of diagnosers G_{d_i}, $i = 1, 2, \ldots, n$ after the execution of a trace in the language L. Note that state x_{t_n} of G_{test_n} has the following structure

$$x_{t_n} = (x_{d_1}, x_{d_2}, \ldots, x_{d_n}, x_d),$$

where $x_{d_i} \in X_{d_i}$ and $x_d \in X_d$. Therefore, the definition of uncertain state and indeterminate cycle can be extended to coordinated decentralized diagnosability, as follows [1].

Definition 5: A state x_t of G_{test_n} is certain if x_d is certain and x_{d_i} is certain for some $i \in \{1, 2, \ldots, n\}$, and is uncertain if x_d is certain and x_{d_i} is uncertain for all $i \in \{1, 2, \ldots, n\}$. □

Definition 6: A cycle in G_{test_n} is said to be indeterminate if all the corresponding cycles (hidden cycles included) in G_{d_i}, $i = 1, 2, \ldots, n$ are indeterminate. □

According to Definition 4, a fully-ambiguous trace s is a trace $s \in \Psi(\Sigma_f)$ that leads to indeterminate cycles in all partial diagnosers G_{d_i}. In addition, due to Assumption A4, for any $s \in \Psi(\Sigma_f)$, there always exists a finite length string t such that st leads to a certain state of G_d. Therefore, G_{test_n} can be used to verify the existence of fully-ambiguous traces, as stated in the following result, which is a variation of Theorem 11 in [1], that accounts for the modified Definition 6 above and the notion of hidden cycles introduced in this paper.

Theorem 4: A live and prefix-closed language L is codiagnosable with respect to the set of projections $P_{o,i} : \Sigma^* \to \Sigma_{o,i}^*$, $i = 1, 2, \ldots, n$ and $\Sigma_f = \{\sigma_f\}$ if, and only if, G_{test_n} does not have any indeterminate cycles. □

Proof: The proof presented in [1] is also valid here, since Definition 6 has been modified to account for indeterminate hidden cycles. □

Let us now remove Assumption A6, that is, let us assume from this point onwards that communication between the local sites and the coordinator is not reliable. This could be due to a break down of the communication channel between a site and the coordinator, or due to the break down of the diagnoser itself at a site. A question arises immediately: Is it still possible to continue using the decentralized architecture and Protocol 3 to diagnose de language generated by an automaton G with respect to $P_{o,i}$, $i = 1, 2, \ldots, n$ and $\Sigma_f = \{\sigma_f\}$ even in the case of permanent loss of communication between one or more sites and the coordinator? This leads to the concept of *robust codiagnosability*.

Definition 7: ($\mathcal{R}_m$-robust codiagnosability) A prefix-closed and live language L that is diagnosable with respect to Protocol 3, a set of projections $P_{o,i}$, $i = 1, 2, \ldots, n$ and $\Sigma_f = \{\sigma_f\}$ is $\mathcal{R}_m$-robust if it is still diagnosable after m sites no longer communicate with the coordinator. □

It is clear from Assumption A6 that robust codiagnosability requires the existence of at least $m+2$ sites. It is also immediate from Definition 7 that L is $\mathcal{R}_m$-robust codiagnosable if, and only, it is diagnosable with respect to Protocol 3, $\Sigma_f = \{\sigma_f\}$, and all sets of projections $P_{o,j_1}, P_{o,j_2}, \ldots, P_{o,j_{n-m}}$, where $\{j_1, j_2, \ldots, j_{n-m}\} \in \mathcal{P}_{n,m}$ with $\mathcal{P}_{n,m}$ being the set formed with all $\begin{pmatrix} n \\ n-m \end{pmatrix}$ combinations of $\{1, 2, \ldots, n\}$ taken $n-m$ at a time. The following result is a consequence of this fact.

Lemma 2: L is $\mathcal{R}_m$-robust codiagnosable with respect to Protocol 3, a set of projections $P_{o,i}$, $i = 1, 2, \ldots, n$ and $\Sigma_f = \{\sigma_f\}$ if, and only if,

$$G_{\text{test}}^{j_1,j_2,\ldots,j_{n-m}} = G_{d_{j_1}} \| G_{d_{j_2}} \| \ldots \| G_{d_{j_{n-m}}} \| G_d \qquad (10)$$

does not have any indeterminate cycle for all sets $\{j_1, j_2, \ldots, j_{n-m}\} \in \mathcal{P}_{n,m}$.

Proof: The proof is straightforward and can be performed through repeated applications of Theorem 4. □

Lemma 2 shows that in order to check whether or not a decentralized architecture is $\mathcal{R}_m$-robust codiagnosable, it is necessary to perform $\begin{pmatrix} n \\ n-m \end{pmatrix}$ parallel compositions formed according to (10). Even in the simplest case of $\mathcal{R}_1$-robust codiagnosability, it is not difficult to see that robustness verification involves the computation of the n parallel compositions

$$G_{\text{test}}^{1,\ldots,k-1,k+1,\ldots,n} = (\|_{i=1,i\neq k}^{n} G_{d_i}) \| G_d,\ k = 1, 2, \ldots, n. \qquad (11)$$

Therefore, it would be useful to generate a less computationally demanding test for this verification. As in the case of codiagnosability, a simpler test for robust codiagnosability requires the definition of indeterminate $\mathcal{R}_m$-robust cycle. For the sake of simplicity, $m = 1$ will be assumed from this point onwards.

Definition 8: (Indeterminate $\mathcal{R}_1$-robust cycle) Let L be diagnosable with respect to Protocol 3, a set of projections $P_{o,i}$, $i = 1, 2, \ldots, n$ and $\Sigma_f = \{\sigma_f\}$. A cycle in G_{test_n} formed by states $x_{t_1}, x_{t_2}, \ldots, x_{t_p}$, where $x_{t_k} = (x_{d_1}^{(k)}, x_{d_2}^{(k)}, \ldots, x_{d_n}^{(k)}, x_d^{(k)})$, $x_{d_i}^{(k)} \in X_{d_i}$ and $x_d^{(k)} \in X_d$, is indeterminate $\mathcal{R}_1$-robust if there exists $\ell \in \{1, 2, \ldots, n\}$ such that $x_{d_\ell}^{(k)}$, $k = 1, 2, \ldots, p$, do not form an indeterminate cycle, and all $n-1$ components $x_{d_i}^{(k)}$, $i \neq \ell$, of states x_{t_k}, $k = 1, 2, \ldots, p$, form indeterminate cycles (hidden cycles included) in G_{d_i}. □

Definition 8 leads to the following result.

Theorem 5: A language L, diagnosable with respect to Protocol 3, a set of projections and $\Sigma_f = \{\sigma_f\}$, is $\mathcal{R}_1$-robust codiagnosable if, and only if, G_{test_n} does not have any indeterminate $\mathcal{R}_1$-robust cycle.

Proof ($\Longrightarrow$) Assume that G_{test_n} has an indeterminate $\mathcal{R}_1$-robust cycle formed by states $x_{t_1}, x_{t_2}, \ldots, x_{t_p}$ of G_{test_n} where $x_{t_k} = (x_{d_1}^{(k)}, x_{d_2}^{(k)}, \ldots, x_{d_n}^{(k)}, x_d^{(k)})$. Therefore, according to Definition 8, there exists an $\ell \in \{1, 2, \ldots, N\}$ such that $x_{d_\ell}^{(k)}$, $k = 1, 2, \ldots, p$, do not form an indeterminate cycle, and all $n-1$ components $x_{d_i}^{(k)}$, $i \neq \ell$, of states x_{t_k}, $k = 1, 2, \ldots, p$, form indeterminate cycles (hidden cycles included) in G_{d_i}. Note that, since $G_{\text{test}_n} = G_{d_1} \| G_{d_2} \| \ldots \| G_{d_\ell} \| \ldots \| G_{d_n} \| G_d$, then, it is not difficult to see that $G_{test}^{1,2,\ldots,\ell-1,\ell+1,\ldots,n}$, defined in accordance with Equation (10), will have an indeterminate cycle formed by states $x'_{t_k} = (x_{d_1}^{(k)}, x_{d_2}^{(k)}, \ldots, x_{d_{\ell-1}}^{(k)}, x_{d_{\ell+1}}^{(k)}, \ldots, x_{d_n}^{(k)}, x_d^{(k)})$, $k = 1, 2, \ldots, p$, which implies that L is not diagnosable with respect to Protocol 3, the set of projections $P_{o,i}$, $i = 1, 2, \ldots, n$, $i \neq \ell$ and $\Sigma_f = \{\sigma_f\}$.

($\Longleftarrow$) Assume that G_{test_n} has no indeterminate $\mathcal{R}_1$-robust cycle. Then, for any set of states $x_{t_1}, x_{t_2}, \ldots, x_{t_p}$ of G_{test_n}, where $x_{t_k} = (x_{d_1}^{(k)}, x_{d_2}^{(k)}, \ldots, x_{d_n}^{(k)}, x_d^{(k)})$, that form cycles in G_{test_n}, there always exist $\ell_1, \ell_2 \in \{1, 2, \ldots, n\}$, $\ell_1 \neq \ell_2$, such that $x_{d_{\ell_1}}^{(k)}$ and $x_{d_{\ell_2}}^{(k)}$, $k = 1, 2, \ldots, p$, do not form indeterminate cycles. It is therefore not difficult to see that $G_{\text{test}}^{1,\ldots,k-1,k+1,\ldots,n} = (\|_{i=1,i\neq k}^{n} G_{d_i}) \| G_d$, $k = 1, 2, \ldots, n$, will have no indeterminate cycle since either $G_{d_{\ell_1}}$, $G_{d_{\ell_2}}$ or both will appear in $G_{\text{test}}^{1,\ldots,k-1,k+1,\ldots,n}$ for any k. Thus, according to Lemma 2, L will be $\mathcal{R}_1$-robust codiagnosable with respect to Protocol 3, a set of projections $P_{o,i}$, $i = 1, 2, \ldots, n$ and $\Sigma_f = \{\sigma_f\}$. □

Example 2: In order to illustrate the results presented in this section, consider automaton G depicted in Figure 1(a). As was seen in Example 1, the language L generated by G is diagnosable with respect to projection P_o and Σ_f.

Consider initially the design of a decentralized architecture composed of three sites. The observable event sets for each site are as follows: $\Sigma_{o_1} = \{a, c\}$, $\Sigma_{o_2} = \{b, c\}$ and $\Sigma_{o_3} = \{c, d\}$. Building the partial diagnosers for each one of these observable event sets, it can be seen that L is not diagnosable with respect to $P_{o,i}$, $i = 1, 2, 3$ due to the existence of indeterminate cycles. In spite of this fact, the decentralized structure is diagnosable with respect to Protocol 3, $P_{o,i}$, $i = 1, 2, 3$ and Σ_f, as one can see in Figure 3(a). However, since all cycles that appear in G_{test_3} are indeterminate $\mathcal{R}_1$-robust, this decentralized diagnoser is not $\mathcal{R}_1$-robust.

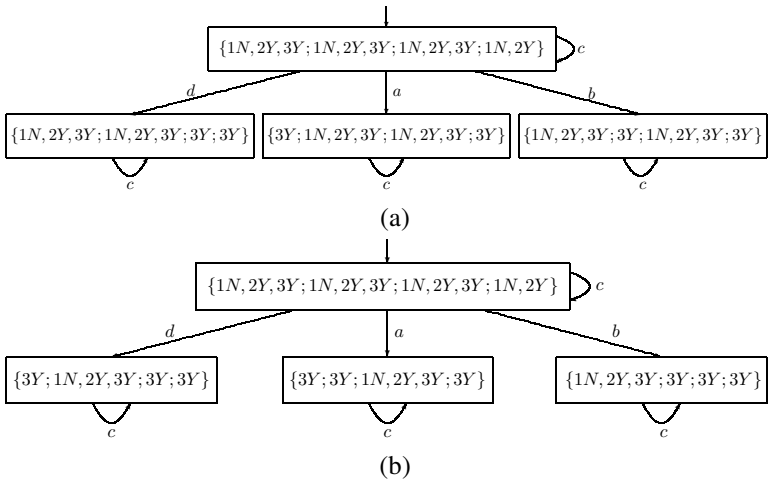

Fig. 3. Test automata for the design of a three-site $\mathcal{R}_1$-robust codiagnoser for the following choices of observable events by each site: $\Sigma_{o_1} = \{a, c\}$, $\Sigma_{o_2} = \{b, c\}$ and $\Sigma_{o_3} = \{c, d\}$ (a), and $\Sigma_{o_1} = \{a, c, d\}$, $\Sigma_{o_2} = \{a, b, c\}$ and $\Sigma_{o_3} = \{b, c, d\}$ (b).

Assume now that the observable events for each site are $\Sigma_{o_1} = \{a, c, d\}$, $\Sigma_{o_2} = \{a, b, c\}$ and $\Sigma_{o_3} = \{b, c, d\}$. In this case, the language L is $\mathcal{R}_1$-robust codiagnosable since, as shown in Figure 3(b), G_{test_3} does not have any indeterminate $\mathcal{R}_1$-robust cycle. It is clear in Figure 3(b) that the fault occurrence is always detected by two sites, meaning that if one site breaks down, the fault occurrence will always be reported to the coordinator by at least one of the two remaining sites. □

Remark 1: For systems that are $\mathcal{R}_1$-robust codiagnosable, Protocol 3 can be used again at runtime. In this case, at least two diagnosers will diagnose each fault (if none are down) and at least one will (if one is down). □

C. Verification of robust codiagnosability using verifiers

A test for robust codiagnosability using diagnoser automata was proposed in the previous subsection. However, it is well known that the state space of the diagnoser is, in the worst case, exponential in the cardinality of the state space of the system model. This problem can be circumvented with the use of verifiers [27], [25], [4]. Verifiers are nondeterministic automata, whose state space is polynomial in the cardinality of the state space of the system model; the verification of diagnosability however requires polynomial time for both, diagnoser and verifier automata.

Consider a system modeled by a deterministic automaton defined according to Equation (1), and let $\Sigma_{o,i}$, $i = 1, 2, \ldots, n$ be the sets of observable events for each site. The one-level verifier to test if L is diagnosable with respect to projections $P_{o,i}$, $i = 1, 2, \ldots, n$, and $\Sigma_f = \{\sigma_f\}$ is the nondeteministic automaton $V_1 = (X^{V_1}, \Sigma^{V_1}, f^{V_1}, x_0^{V_1})$, where $X^{V_1} = (X \times \{N, Y\})^{n+1}$, $\Sigma^{V_1} = (\Sigma \cup \{\epsilon\})^{n+1}$, $x_0^{V_1} = (x_0, N, x_0, N, \ldots, x_0, N)$ is a $(2n+2)$-dimensional vector, and f^{V_1} is defined as follows: *(i)* let $x^{V_1} = (x_1, \ell_1, x_2, \ell_2, \ldots, x_{n+1}, \ell_{n+1})$, where $x_i \in X$, $i = 1, \ldots, n$, and assume that $x_i' = f(x_i, \sigma)$, $\sigma \in \Gamma(x_i)$; *(ii)* for a set $J \subseteq I_{n+1} = \{1, 2, \ldots, n+1\}$, define the sequence $\sigma^J = \sigma_1 \sigma_2 \ldots \sigma_{n+1}$ such that $\sigma_i = \sigma$, $\forall i \in J$ and $\sigma_i = \epsilon$, otherwise; *(iii)* then, for each $\sigma \in \cup_{i=1}^{n} \Gamma(x_i)$:

1) if $\sigma \in \Sigma_o$, define the sets of indices $L = \{l_1, l_2, \ldots, l_n\} \subseteq \{1, 2, \ldots, n\}$, where $l_i \in L$ if and only if $\sigma \in \Sigma_{o,i}$, $K = \{k\}$ and $M = L \cup \{n+1\}$. Then

$$\begin{aligned} f^{V_1}(x^{V_1}, \sigma^M) &= x_M^{V_1}, \\ f^{V_1}(x^{V_1}, \sigma^K) &= x_K^{V_1}, \forall k \in I_{n+1} \setminus M, \end{aligned}$$

where $x_M^{V_1}$ and $x_K^{V_1}$ are defined similarly, *i.e.*, for any set $W \subseteq I_{n+1}$, $x_W^{V_1} = (x_{W,1}, \ell_1, \ldots, x_{W,n+1}, \ell_{n+1})$, with $x_{W,i} = x_i'$, $\forall i \in W$ and $x_{W,i} = x_i$, otherwise.

2) if $\sigma \in \Sigma_{uo} \setminus \Sigma_f$, then

$$f^{V_1}(x^{V_1}, \sigma^K) = x_K^{V_1}, \forall k \in I_{n+1},$$

where $x_K^{V_1}$ is defined as in 1) above.

3) if $\sigma = \sigma_f$, then

$$f^{V_1}(x^{V_1}, \sigma^K) = x_{K_f}^{V_1}, \forall k \in I_{n+1},$$

where $x_{K_f}^{V_1} = (x_{K_f,1}^{V_1}, \ell_1^K, \ldots, x_{K_f,n+1}^{V_1}, \ell_{n+1}^K)$, with $x_{K_f,k}^{V_1} = x_k'$, $\ell_k^K = Y$, and $x_{K_f,i}^{V_1} = x_i$, $\ell_i^K = \ell_i$, otherwise.

The idea behind the construction of one-level verifiers is summarized by the following proposition.

Proposition 1: [4] There exists a path from $x_0^{V_1}$ to $x^{V_1} = (x_1, \ell_1, x_2, \ell_2, \ldots, x_{n+1}, \ell_{n+1}) \in X^{V_1}$ obtained by using the transition rules of V_1 defined above if and only if the traces s_i, $i = 1, \ldots, n$, and s formed, respectively, with the $i-th$ and $(n+1)-st$ component of the transitions along the path satisfy the following three conditions: *1)* s_i, $i = 1, \ldots, n$, and s reach states $x_1, x_2, \ldots, x_{n+1} \in X$; *2)* s_i is faulty if and only if $\ell_i = Y$, $i = 1, 2, \ldots, n+1$; *3)* $P_{o,i}(s_i) = P_{o,i}(s)$, $i = 1, 2, \ldots, n$ □

A one-level verifier state $x^{V_1} = (x_1, \ell_1, x_2, \ell_2, \ldots, x_{n+1}, \ell_{n+1})$ is called an $(\ell_1, \ell_2, \ldots, \ell_{n+1})$-state (for example, the initial state $x_0^{V_1}$ is an $(N, N, \ldots, N)$-state). A strongly connected component (SCC) of V_1 is called an $(\ell_1, \ell_2, \ldots, \ell_{n+1})$-SCC if every state in the SCC is an $(\ell_1, \ell_2, \ldots, \ell_{n+1})$-state. With that in mind, a test for codiagnosability using the one-level verifier is provided by the following theorem.

Theorem 6: [4], [2] The language L is not diagnosable with respect to projections $P_{o,i}$, $i = 1, 2, \ldots, n$, and $\Sigma_f =$

$\{\sigma_f\}$ if and only if the one-level verifier V_1 of G has an $(N, N, \ldots, N, Y)$-SCC, in which there exists an edge $\sigma_1\sigma_2\ldots\sigma_n\sigma$ such that $\sigma \neq \epsilon$. □

Following the same reasoning as for robust codiagnosability using the G_{test_n} automaton, the following result (which is a counterpart of Lemma 2) can be stated.

Lemma 3: L is not $\mathcal{R}_m$-robust codiagnosable with respect to Protocol 3, a set of projections $P_{o,i}$, $i = 1, 2, \ldots, n$ and $\Sigma_f = \{\sigma_f\}$ if, and only if, for at least one set of projections P_{o,j_i}, where $j_i \in \mathcal{P}_{n,m}$, $i = 1, 2, \ldots, n-m$, and $\mathcal{P}_{n,m}$ is the set of all $\begin{pmatrix} n \\ n-m \end{pmatrix}$ combinations of $\{1, 2, \ldots, n\}$ taken $n-m$ at a time, the corresponding one-level verifiers $V_1^{j_1,j_2,\ldots,j_{n-m}} = (X^{V_1}, \Sigma^{V_1}, f^{V_1}, x_0^{V_1})$, where $X^{V_1} = (X \times \{N, Y\})^{n-m+1}$, $\Sigma^{V_1} = (\Sigma \cup \{\epsilon\})^{n-m+1}$, has an $(N, N, \ldots, N, Y)$-SCC, in which there exists an edge $\sigma_1\sigma_2\ldots\sigma_{n-m}\sigma$ such that $\sigma \neq \epsilon$. □

Assuming $m = 1$, indeterminate R_1-robust cycles become non-R_1-robust SCC, as follows.

Definition 9: (non-R_1-robust SCC) Assume that the language L is diagnosable with respect to projections $P_{o,i}$, $i = 1, 2, \ldots, n$, and $\Sigma_f = \{\sigma_f\}$. An $(\ell_1, \ell_2, \ldots, \ell_n, Y)$-SCC is a non-$R_1$-robust SCC if there exists exactly one $i \in \{1, 2, \ldots, n\}$ such that $\ell_i = Y$. □

A necessary and sufficient condition for R_1-robust codiagnosability using verifiers is provided by the following theorem

Theorem 7: Assume that the language L is diagnosable with respect to projections $P_{o,i}$, $i = 1, 2, \ldots, n$, and $\Sigma_f = \{\sigma_f\}$. Then L will be not R_1-robust codiagnosable if and only if the corresponding verifier V_1 has at least one non-R_1-robust SCC.

Proof: The proof is omitted. If follows by adapting and combining the methodologies of the proofs of Theorems 6 and 5, and by using Lemma 3. □

V. CONCLUSION

We have revisited the problem of centralized diagnosability and used the notions of partial diagnosers and hidden cycles to characterize the effect of reducing the set of observable events. We have proposed an intuitive notion of robustness in the context of decentralized diagnostic architectures. We have developed necessary and sufficient conditions for testing robust codiagnosability using familiar tools such as diagnosers and verifiers. An interesting direction for future investigations is the problem of selecting/adjusting the local observable event sets in order to achieve robust codiagnosability.

REFERENCES

[1] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discrete Event Dynamic Systems*, vol. 10, pp. 33–86, 2000.

[2] Y. Wang, T. S. Yoo, and S. Lafortune, "Diagnosis of discrete event systems using decentralized architectures," *Discrete Event Dynamic Systems*, vol. 17, pp. 233–263, 2007.

[3] R. Kumar and S. Takai, "Inference-based ambiguity management in decentralized decision-making: Decentralized diagnosis of discrete event systems," in *Proceedings of the 2006 American Control Conference*, Minneapolis, USA,, 2006, pp. 6069–6074.

[4] W. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," *IEEE Trans. on Systems, Man and Cybernetics – Part A: Systems and Humans*, vol. 36, pp. 384–395, 2006.

[5] R. Sengupta, "Diagnosis and communication in distributed systems," in *Proceedings of 1998 International Workshop on Discrete Event Systems – WODES'98*, Cagliari, Italy, 1998.

[6] D. Thorsley, T.-S. Yoo, and H. Garcia, "Diagnosability of stochastic discrete-event systems under unreliable observations," in *Proceedings of the 2008 American Control Conference*, Seatle, WA, 2008, pp. 1158–1365.

[7] S. Genc, "Formal methods for intrusion detection of windows NT attacks," in *3rd Annual Symposium on Information Assurance (ASIA '08) & 11th Annual NYS Cyber Security Conference*, Albany, NY, 2008.

[8] J. Chen and R. J. Patton, *Robust model-based fault diagnosis for dynamic systems*. Norwell, MA: Kluwer Academic Publishers, 1999.

[9] R. S. Mangoubi, *Robust Estimation and Failure Detection: A Concise Treatment*. Secaucus, NJ: Springer-Verlag, 1998.

[10] E. Athanasopoulou, L. Lingxi, and C. N. Hadjicostis, "Probabilistic failure diagnosis in finite state machines under unreliable observations," in *Proceedings of the 8th International Workshop on Discrete Event Systems – WODES'06*, Ann Arbor, MI, pp. 301–306.

[11] L. Travé-Massuyè, T. Escobet, and X. Olive, "Diagnosability analysis based on component-supported analytical redundancy relations," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 36, pp. 1146–1160, 2006.

[12] Q. Wen, R. Kumar, J. Huang, and H. Liu, "Fault-tolerant supervisory control of of discrete event systems: formalisation and existence results," in *Proceedings of the 1st IFAC Workshop on Dependable Control of Discrete Systems*, Paris, France, 2007.

[13] A. Paoli, M. Sartini, and S. Lafortune, "A fault tolerant architecture for supervisory control of discrete event systems," in *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, 2008, pp. 6542–6547.

[14] S. Takai and T. Ushio, "Reliable decentralized supervisory control of discrete event systems," *IEEE Transactions on Systems, Mand, and Cybernetics – Part B: Cybernetics*, vol. 30, pp. 661–667, 2000.

[15] R. K. Boel and J. H. van Schuppen, "Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers," in *Proc. of the 2002 International Workshop on Discrete Event Systesm – WODES'02*, Zaragoza, Spain, 2002.

[16] Y. Pencolé and M. O. Cordier, "A formal framework for the decentralized diagnosis of large scale discrete event systems and its applications to telecommunication networks," *Artificial Intelligence*, vol. 164, pp. 121–170, 2005.

[17] E. Fabre, A. Benveniste, S. Haar, and C. Jard, "Distributed monitoring of concurrent and asynchronous systems," *Discrete Event Dynamic Systems*, vol. 15, pp. 33–84, 2005.

[18] E. Fabre, A. Benveniste, C. Jard, L. Ricker, and M. Smith, "Distributed stare reconstruction for discrete event systems," in *IEEE Conference on Decision and Control*, 2000, pp. 2252–2257.

[19] S. Genc and S. Lafortune, "Distributed diagnosis of place-bordered petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 4, pp. 206–219, 2007.

[20] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. on Automatic Control*, vol. 40, pp. 1555–1575, 1995.

[21] ——, "Failure diagnosis using discrete event models," *IEEE Trans. on Control Systems Technology*, vol. 4, pp. 105–124, 1996.

[22] S. Lafortune, D. Teneketzis, M. Sampath, R. Sengupta, and K. Sinnamohideen, "Failure diagnosis of dynamic systems: an approach based on discrete event systems," in *Proceedings of the American Control Conference*, 2001, pp. 2058–2071.

[23] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York: Springer, 2007.

[24] P. J. Ramadge and W. M. Wonham, "The control of discrete-event systems," *Proceedings of the IEEE*, vol. 77, pp. 81–98, 1989.

[25] T. S. Yoo and S. Lafortune, "Polynomial-time verification of diagnosability of partially observed discrete-event systems," *IEEE Trans. on Automatic Control*, vol. 47, pp. 1491–1495, 2002.

[26] J. C. Basilio and S. Lafortune, "Robust diagnosis of discrete event systems," UMICH, Department of Electrical Engineering and Computer Science, Tech. Rep., 2008.

[27] S. Jiang, Z. Huang, V. Chandra, and R. Kumar, "Polynomial algorithm for testing diagnosability of discrete-event systems," *IEEE Trans. on Automatic Control*, vol. 46, pp. 1318–1321, 2001.