CGR 04-04

Modular Approaches to Diagnosis and Verification of Discrete Event Systems

Olivier Contant, Stéphane Lafortune, and
Demosthenis Teneketzis
Department of Electrical Engineering and Computer Science,
The University of Michigan,
1301 Beal Avenue, Ann Arbor, MI 48109–2122 USA
{olivier, stephane, teneket}@eecs.umich.edu
http://www.eecs.umich.edu/umdes/

1 Introduction

Monitoring and diagnosis methodologies that are based upon discrete-event models of dynamic systems have proved successful in many application areas including document processing systems, heating, ventilation, and air-conditioning systems, intelligent transportation systems, chemical process control, and telecommunication networks. Most of the discrete-event fault diagnosis methodologies that have been studied in the literature require a "monolithic" model of the system under consideration for diagnosability analysis and/or construction of the diagnostic protocol; see (Lafortune *et al.*, 2001) and the references therein. Notable exceptions include the approaches developed in (Holloway and Chand, 1994; Ricker and Fabre, 2000; Fabre *et al.*, 2002; García *et al.*, 2002; Su *et al.*, 2002; Debouk *et al.*, 2002; Genc and Lafortune, 2003) where the modular structure of the underlying system is exploited by the respective monitoring or diagnostic protocols. This paper has a similar objective, although the approach adopted is different from those in the above references.

We consider systems that are modeled by the parallel composition of a set of automata. Each individual automaton models a component or subsystem of the overall system. The coupling of these system components is captured by the use of common events. We are concerned with systems where the construction of the complete system model (namely, the *monolithic* model obtained by performing the parallel composition of the set of individual automata) is computationally difficult or intractable, making the use of fault diagnosis methodologies such as the "Diagnoser Approach" in (Sampath *et al.*, 1995; Debouk *et al.*, 2000; Contant *et al.*, 2002) impractical. Therefore, it is necessary to develop *modular approaches* that are computationally tractable for analyzing the diagnosability properties of the system and for synthesizing appropriate diagnostic protocols. By "modular approaches" we mean approaches that exploit the structure of the system as captured by the individual automata and their respective sets of common events.

The first contribution of this paper is to present a new notion of diagnosability that is explicitly geared towards systems with modular representations. This notion of *modular diagnosability*, presented in Section 3, is adapted from the notion of diagnosability introduced in (Sampath *et al.*, 1995). Necessary and sufficient conditions for modular diagnosability are presented and discussed. The second contribution of this paper is the development in Section 4 of a novel algorithm for verifying modular diagnosability. This algorithm (abbreviated MDA hereafter) proceeds incrementally by including the automata models of other system components only if they are required to draw definitive conclusions about the diagnosability of faults within a given system component. A key assumption made in the development of MDA is that all common events among two or more system components are observable.

2 System Model

Let I be the total number of components or subsystems in the given modular system under consideration. Let $T = \{1, \ldots, I\}$ and $S \subseteq T$. Elements i of T are often termed "sites" hereafter. We use the notation

$$G_S = (X_S,\ \Sigma_S,\ \delta_S,\ x_{S_0},\ X_{S_m}) \tag{1}$$

to denote the automaton with state space X_S, set of events Σ_S, (partial) transition function δ_S, initial state x_{S_0}, and set of marked states X_{S_m}. When $S = T$, G_S denotes the global (or complete) system model. When $S = \{i\}$, with $i \in T$, G_S denotes individual component model i. When $S \subset T$, $S \neq \{i\}$, $i \in T$, G_S denotes the partial system model comprised of the individual automata in the set S, which we will call the "system G_S" hereafter. In all of the above cases, system G_S accounts for the normal and failed behavior of the components in S, consistent with the Diagnoser Approach of (Sampath *et al.*, 1995; Sampath *et al.*, 1996). $G_S = \|_{z \in S} G_z$ is obtained by composing the individual automata G_z, $z \in S$, using the parallel composition operation.

The behavior of the system G_S is described by the prefix-closed language $\mathcal{L}(G_S)$ generated by G_S. $\mathcal{L}(G_S)$ is assumed to be live. This means that there is a transition defined at each state x in X_S, i.e., G_S cannot reach a point at which no event is possible. The liveness assumption on $\mathcal{L}(G_S)$ is made for the sake of simplicity. With slight modifications, all the main results of this paper hold true when the liveness assumption is relaxed. Some of the events in Σ_S are observable, i.e., their occurrence can be observed by sensors, while the rest are unobservable. We use the notation Σ_{o_S} and Σ_{uo_S} to represent the set of observable and unobservable events of G_S, respectively, where $\Sigma_{o_S} = \Sigma_S \setminus \Sigma_{uo_S}$. Let Σ_{f_S} denote the set of fault events in the system G_S. Without loss of generality, we assume that $\Sigma_{f_S} \subseteq \Sigma_{uo_S}$, since an observable fault event can be diagnosed trivially.

Due to page limitation, it has been necessary to assume that the reader is familiar with basic notions[1] in languages and automata theory and with the notation and main concepts of the Diagnoser Approach of (Sampath *et al.*, 1995), such as diagnosers, certain and uncertain states, and indeterminate cycles. We add the following definitions regarding notation. The notation $\Sigma = \Sigma' \dot{\cup} \Sigma''$ indicates that the event set Σ is the disjoint union of Σ' and Σ'', i.e., $\Sigma' = \Sigma \setminus \Sigma''$. Let $R \subset T$. The event set Σ_R is partitioned as $\Sigma_R = \Sigma_{CM_R} \dot{\cup} \Sigma_{PV_R}$, where $\Sigma_{CM_R} = \Sigma_R \cap [\cup_{z \in T \setminus R} \Sigma_z]$ represents the set of common events in G_R and where Σ_{PV_R} represents the set of private events in G_R. We use the notation $\Sigma_{CM_{o_R}} = \Sigma_{CM_R} \cap \Sigma_{o_R}$ to represent the set of common and observable events. The notation $CoAc(G_S)$ represents the automaton obtained from G_S by retaining only those states x of G_S that are coaccessible, namely, that can reach a (marked) state in X_{S_m}. This notation will often be specialized to $CoAc(G_S, M_x)$ where M_x will be a particular label associated with the marked states of G_S. In this case, $CoAc(G_S, M_x)$ will be the coaccessible part of G_S with respect to the marked states of G_S that are labeled with M_x.

It will be necessary in many instances to explicitly identify the event set associated with an automaton. By default, the event set Σ_S associated with automaton

[1] See, e.g., Chapter 2 of (Cassandras and Lafortune, 1999).

G_S will be $\Sigma_S := \{\sigma \in s : s \in \mathcal{L}(G_S)\}$, namely, all the events that appear in all the traces in $\mathcal{L}(G_S)$. In special cases it is required to define a larger set of events associated with G_S than the default one. In this regard, the notation (G_S, Σ) will denote the automaton G_S together with the event set Σ such that $\Sigma \supseteq \Sigma_S$. For example, (G_i, Σ_S) implies that the original event set Σ_i of G_i is now augmented by the set $\Sigma_S \setminus \Sigma_i$. While (G_i, Σ_S) has the same language properties as (G_i, Σ_i), it has different behavior when performing parallel composition (or, more generally, any operation using sets of events) with other modules as it will prevent the occurrence of events in $\Sigma_S \setminus \Sigma_i$.

We define

$$Obs(G_S, \Sigma^{obs}) = (X_S^{obs}, \Sigma_S^{obs}, \delta_S^{obs}, x_{S_0}^{obs}) \tag{2}$$

to be the observer of G_S with respect to Σ^{obs} (i.e., the set of specific events to be observed) where $\Sigma^{obs} \subset \cup_{z \in T} \Sigma_z$ and $\Sigma_S^{obs} = \Sigma_S \cap \Sigma^{obs}$. (OLIV: put the definition on the observer or detector as you wish - See (Cassandras and Lafortune, 1999) for the complete definition of an observer.) For example, $Obs(G_i, \Sigma_{CM_l})$, $l \neq i$, is the observer of G_i with respect to Σ_{CM_l}, the common events of G_l with G_i.

Let Σ_X and Σ_Y be any sets of events. We define two projection operators relative to these sets, $P_{\{\Sigma_X, \Sigma_Y\}}$ for the usual natural projection and $R_{\{\Sigma_X, \Sigma_Y\}}$ for the so-called "reverse" projection. Specifically, the natural projection $P_{\{\Sigma_X, \Sigma_Y\}} : \Sigma_X^* \to \Sigma_Y^*$ is defined in the usual manner:

$$\begin{aligned} P_{\{\Sigma_X, \Sigma_Y\}}(\epsilon) &:= \epsilon && (3) \\ P_{\{\Sigma_X, \Sigma_Y\}}(\sigma) &:= \begin{cases} \sigma & \text{if } \sigma \in \Sigma_Y \\ \epsilon & \text{if } \sigma \notin \Sigma_Y \end{cases} && (4) \\ P_{\{\Sigma_X, \Sigma_Y\}}(s\sigma) &:= P_{\{\Sigma_X, \Sigma_Y\}}(s) P_{\{\Sigma_X, \Sigma_Y\}}(\sigma) \text{ for } s \in \Sigma_X^*, \sigma \in \Sigma_X. \end{aligned}$$

In contrast, the *reverse projection* $R_{\{\Sigma_X, \Sigma_Y\}}$ is applied to traces of events from Σ_X^* and produces "inverse" traces from Σ_Y^* as is usually done in inverse projection operations. More precisely, $R_{\{\Sigma_X, \Sigma_Y\}} : \Sigma_X^* \to 2^{\Sigma_Y^*}$ is defined as follows:

$$R_{\{\Sigma_X, \Sigma_Y\}}(s) = \{t \in \Sigma_Y^* : P_{\{\Sigma_Y, \Sigma_X\}}(t) = s\}. \tag{5}$$

The natural and reverse projections can also be defined with respect to a particular language L. For $L \subseteq \Sigma_Y^*$,

$$P^L_{\{\Sigma_X, \Sigma_Y\}}(s) = \{t \in L : P_{\{\Sigma_X, \Sigma_Y\}}(s) = t\}, \tag{6}$$

and

$$R^L_{\{\Sigma_X,\Sigma_Y\}}(s) = \{t \in L : P_{\{\Sigma_Y,\Sigma_X\}}(t) = s\}. \tag{7}$$

We conclude this section by stating a key assumption that will be required for the results presented in the remainder of the paper: *all common events are observable.* This implies that all faults are private events (since they are unobservable).

3 Modular Diagnosability

We define the notion of modular diagnosability as follows.

Definition 1 *:* ***Modular Diagnosability***
Let $T = \{1, ..., I\}$, $S \subseteq T$, $G_S = \|_{z\in S}\, G_z$, and $S^- \subseteq S$. The language $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S^-)$ if $\forall i \in S^-$, $\forall f \in \Sigma_{f_i}$, $\forall s \in \mathcal{L}(G_S)$ s.t. s ends with f, $\exists n \in \mathbb{N}$ s.t. $\forall t \in \mathcal{L}(G_S)/s$, $\| P_{\{\Sigma_S,\Sigma_{o_i}\}}(t) \| \geq n$ $\Rightarrow D(st) = 1$. The diagnosability condition function D is given by

$$D(st) = \begin{cases} 1 & \text{if } \omega \in R^{\mathcal{L}(G_S)}_{\{\Sigma_{o_S},\Sigma_S\}}[P_{\{\Sigma_S,\Sigma_{o_S}\}}(st)] \Rightarrow f \in \omega, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

To draw comparisons between our modular diagnosability algorithm, presented in Section 5, and any other potential approach, we give necessary and sufficient conditions for modular diagnosability based on monolithic constructions of the system model and diagnoser for a given set S of system components. G_{d_S} and G'_S denote, respectively, the diagnoser of G_S and the non-deterministic automaton built from G_S by eliminating unobservable events; both are defined in (Sampath *et al.*, 1995).

Definition 2 *:* ***F^{M_i}-indeterminate cycle***
A set of F-uncertain states $q_1, q_2, \ldots, q_n \in Q_{d_S}$ is said to form an F^{M_i}- indeterminate cycle in G_{d_S} if the following condition C1 is satisfied.

C1) States $q_1, q_2, \ldots, q_n \in Q_{d_S}$ form a cycle in G_{d_S} with $\delta_{d_S}(q_u, \sigma_u) = q_{u+1}$, $u = 1, \ldots, n-1$, $\delta_{d_S}(q_n, \sigma_n) = q_1$ where $\sigma_u \in \Sigma_{o_S}$, $u = 1, \ldots, n$ and $\exists l \in \{1, \ldots, n\}$ s.t. $\sigma_l \in \Sigma_{o_i}$.
Considering the states $q_1,\ q_2, \ldots,\ q_n \in Q_{d_S}$, $\exists\ (x^k_u,\ \ell^k_u)$, $(y^r_u,\ \tilde{\ell}^r_u) \in q_u$, $u = 1, \ldots, n$, $k = 1, \ldots, m$, and $r = 1, \ldots, m'$ such that:

(i) $[(F \in \ell_u^k) \wedge (F \notin \tilde{\ell}_u^r)]$, for all u, k, and r, where F represents the label associated with the fault event $f \in \Sigma_{f_i}$, $i \in S$,

(ii) the sequences of states $\{x_u^k\}$, $u = 1, \ldots, n$, $k = 1, \ldots, m$, and $\{y_u^r\}$, $u = 1, \ldots, n$, $r = 1, \ldots, m'$, form cycles in G'_S with

- $(x_u^k, \sigma_u, x_{(u+1)}^k) \in \delta_{G'_S}$, $u = 1, \ldots, n-1$, $k = 1, \ldots, m$, $(x_n^k, \sigma_n, x_1^{k+1}) \in \delta_{G'_S}$, $k = 1, \ldots, m-1$, $(x_n^m, \sigma_n, x_1^1) \in \delta_{G'_S}$, and
- $(y_u^r, \sigma_u, y_{(u+1)}^r) \in \delta_{G'_S}$, $u = 1, \ldots, n-1$, $r = 1, \ldots, m'$, $(y_n^r, \sigma_n, y_1^{r+1}) \in \delta_{G'_S}$, $r = 1, \ldots, m'-1$, $(y_n^{m'}, \sigma_n, y_1^1) \in \delta_{G'_S}$.

Remark 1 *The symbol "M_i", in the notation "F^{M_i}-indeterminate cycle", stands for "Module G_i". F^{M_i}-indeterminate cycles differ slightly from the F-indeterminate cycles introduced in (Sampath* et al.*, 1995) in two respects. First, we require that there exists at least one observable event from module G_i in the cycle of states $q_1, q_2, \ldots, q_n \in Q_{d_S}$: cf. "$\exists l \in \{1, \ldots, n\}$ s.t. $\sigma_l \in \Sigma_{o_i}$" in Condition C1. Second, we require that the label F in hypothesis (i) of Condition C1 represents the label associated with the fault event $f \in \Sigma_{f_i}$, i.e., the fault event f originates from module G_i.*

Theorem 1 *: Consider the language $\mathcal{L}(G_S)$ generated by automaton $G_S = \|_{z \in S} G_z$. $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_i} : i \in S)$, iff there are no F^{M_i}-indeterminate cycles in the diagnoser G_{d_S}.*

The proof of Theorem 1 is similar to the one in (Sampath *et al.*, 1995) and is therefore omitted.

To gain insight into the definition of modular diagnosability, we reformulate with minor modifications the notion of diagnosability introduced in (Sampath *et al.*, 1995) and refer to it from now on as *monolithic diagnosability.*

Definition 3 *:* ***Monolithic Diagnosability***
Let $T = \{1, ..., I\}$, $S \subseteq T$, and $G_S = \|_{z \in S} G_z$. The language $\mathcal{L}(G_S)$ is monolithically diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$ if $\forall i \in S$, $\forall f \in \Sigma_{f_i}$, $\forall s \in \mathcal{L}(G_S)$ s.t. s ends with f, $\exists n \in \mathbb{N}$ s.t. $\forall t \in \mathcal{L}(G_S)/s$, $\| P_{\{\Sigma_S, \Sigma_{o_S}\}}(t) \| \geq n \Rightarrow D(st) = 1$, where the diagnosability condition function D is as in Eq. 8.

Remark 2 *Definition 3 differs from the diagnosability definition introduced in (Sampath* et al.*, 1995) as follows:*

i.) We use the equation $\parallel P_{\{\Sigma_S,\Sigma_{o_S}\}}(t) \parallel \geq n$ instead of $\parallel t \parallel \geq n$. This modification implies that cycles of unobservable events are not taken into account when verifying the diagnosability properties of a system.

ii.) The order of the quantifiers allows one natural number n for each trace s that ends with a fault event, instead of requiring one natural number for each fault event f, i.e., for all traces s ending with f. This change[2] *allows for more precise choices of lower bounds for fault detection and identification.*

In the special case where the considered system G_S is composed of a single module, i.e., $|S| = 1$, there are no distinctions between first, the modular and monolithic definitions, and second, F^{M_i}- and F-indeterminate cycles. In the general case, the main difference between the modular and the monolithic definitions of diagnosability concerns the type of traces that need to be considered. When testing for diagnosability of a fault event f at the end of trace s, we consider projections of any continuation t of length greater than n. For monolithic diagnosability, the projection of t is with respect to the observable events of system G_S, i.e., $\parallel P_{\{\Sigma_S,\Sigma_{o_S}\}}(t) \parallel \geq n$. For modular diagnosability, the projection of t is with respect to the observable events of system G_i, i.e., $\parallel P_{\{\Sigma_S,\Sigma_{o_i}\}}(t) \parallel \geq n$. Therefore, modular diagnosability focuses only on traces where events from module G_i, which is the module where the fault originates, occur with some regularity. Consequently, the notion of modular diagnosability is weaker than the notion of monolithic diagnosability since more languages will satisfy this definition than the monolithic one.

Our primary motivation for defining modular diagnosability is to ensure that after a fault occurs in one of the system modules, detection and isolation of that fault is only required for continuations that involve events from the given module. It is reminiscent of the familiar "persistency of excitation" condition in system identification. In other words, continuations that entirely exclude the module where the fault originates from cannot lead to a violation of modular diagnosability. (Recall that the approach that we propose assumes that faults do not bring the system, or any of its modules, to a halt.)

For the sake of illustration, let us consider a simple Local Area Network (LAN) composed of several interconnected computers. The LAN is the system to be diagnosed and the computers attached to it represent the local systems or modules. The faults or special events to be detected are "illegal" intrusions into the LAN.

[2] Other researchers have also independently suggested this change (Yoo, 2003).

Therefore if an (unobservable) intrusion occurs at one of the computers and that computer does not exhibit any behavior after the intrusion, i.e., the local site does not supply any observable events, then clearly this intrusion does not need to be diagnosed since it is not exploited. On the other hand, if the intruder takes advantage of its trespass, then it is essential to diagnose the violation. In other words, we expect that the intruder will sufficiently exert the afflicted computer so that the intrusion in the LAN can eventually be detected. This concept is similar to the one used in signature-based Intrusion Detection Systems (IDS) where the signatures are specific sequences of (observable) events, cf. (Coolen and Luiijf, 2002). IDS gather sequences of observable events and verify if these sequences match one sequence in IDS signature databases. In order to potentially match a signature, IDS require arbitrarily long exertion of the local system.

The following example illustrates the difference between modular and monolithic diagnosability.

Example 1 *Let $T = \{1, 2, 3\}$. Consider the system modules G_1, G_2, and G_3, the monolithic system $G_T = G_1 \parallel G_2 \parallel G_3$, the monolithic diagnoser G_{d_T}, and their respective event sets Σ_1, Σ_2, Σ_3, Σ_T, and Σ_{d_T}. The models are depicted in Fig. 1. We have $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a, b, x, y\}$, $\Sigma_1 = \{a, f\}$, $\Sigma_2 = \Sigma_3 = \{a, b, x, y\}$, $\Sigma_T = \{b, f\}$, and $\Sigma_{d_T} = \{b\}$. The diagnoser G_{d_T} contains a cycle of F-uncertain states, where F is the label associated with the fault event $f \in \Sigma_1$. We check the necessary and sufficient conditions of modular and monolithic diagnosability. The diagnoser G_{d_T} contains a cycle formed by the self-loop $b \in \Sigma_2 \cap \Sigma_3$ at the F-uncertain state $q = \{3F, 4N\}$. It can be verified that this is an F-indeterminate cycle in G_{d_T}. Therefore the system G_T is* not monolithically diagnosable *w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$. On the other hand, there does not exist an F^{M_i}-indeterminate cycle in G_{d_T} since $f \in \Sigma_1$ and $b \notin \Sigma_1$. Hence G_T is* modularly diagnosable *w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$. Intuitively, the above results are clear since the (monolithic) diagnoser contains only events from subsystems G_2 and G_3 while the fault to be diagnosed originates from module G_1.* ♢

We formalize the relationship between modular and monolithic diagnosability in the following theorem.

Theorem 2

Part 1. *Let $T = \{1, ..., I\}$, $S \subseteq T$, and $G_S = \|_{z \in S} G_z$. If the language $\mathcal{L}(G_S)$*

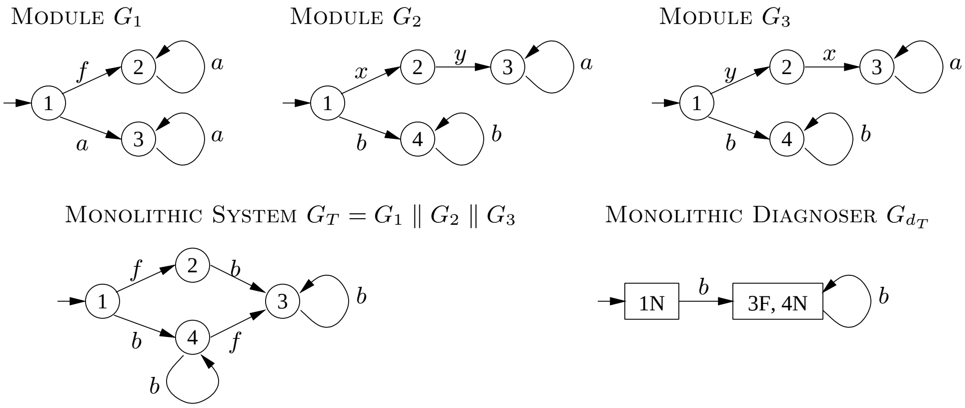

Module G_1
Module G_2
Module G_3
Monolithic System $G_T = G_1 \parallel G_2 \parallel G_3$
Monolithic Diagnoser G_{d_T}
1N
3F, 4N
f
a
x
y
b

Figure 1: Modular vs. Monolithic Diagnosability Example

is monolithically diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ *and* $(\Sigma_{f_z} : z \in S)$ *then* $\mathcal{L}(G_S)$ *is modularly diagnosable w.r.t.* $(\Sigma_{o_z} : z \in S)$ *and* $(\Sigma_{f_z} : z \in S)$*.*

Part 2. *Let* $T = \{1, ..., I\}$*,* $S \subseteq T$*,* $G_S = \|_{z \in S} G_z$*, and* $i \in S$*. If the language* $\mathcal{L}(G_i)$ *is monolithically diagnosable w.r.t.* Σ_{o_i} *and* Σ_{f_i} *then* $\mathcal{L}(G_S)$ *is modularly diagnosable w.r.t.* $(\Sigma_{o_z} : z \in S)$ *and* Σ_{f_i}*.*

Proof: *Theorem 2 Part 1*

We prove the contrapositive, i.e., if $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$, then $\mathcal{L}(G_S)$ is not monolithically diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$.

$\mathcal{L}(G_S)$ not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$ implies that $\exists i \in S$, $\exists f \in \Sigma_{f_i}$, $\exists s \in \mathcal{L}(G_S)$ s.t. s ends with f, $\forall n \in \mathbb{N}$, $\exists t \in \mathcal{L}(G_S)/s$ such that $\parallel P_{\{\Sigma_S, \Sigma_{o_i}\}}(t) \parallel \geq n \Rightarrow D(st) = 0$. Since $\parallel P_{\{\Sigma_S, \Sigma_{o_i}\}}(t) \parallel \geq n$ implies $\parallel P_{\{\Sigma_S, \Sigma_{o_S}\}}(t) \parallel \geq n$, then $\mathcal{L}(G_S)$ is not monolithically diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$. ■

Proof: *Theorem 2 Part 2*

By Part 1 of Theorem 2, if the language $\mathcal{L}(G_i)$ is monolithically diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}, then $\mathcal{L}(G_i)$ is modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}. We now prove the following: if $\mathcal{L}(G_i)$ is modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i} then $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}. We prove the contrapositive of the above statement, i.e., if $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}, then $\mathcal{L}(G_i)$ is not modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}.

$\mathcal{L}(G_S)$ not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i} implies that $\exists f \in \Sigma_{f_i}$, $i \in S$, $\exists s \in \mathcal{L}(G_S)$ s.t. s ends with f, $\forall n \in \mathbb{N}$, $\exists t \in \mathcal{L}(G_S)/s$ such that $\parallel P_{\{\Sigma_S,\Sigma_{o_i}\}}(t) \parallel \geq n \Rightarrow D(st) = 0$, i.e., $\exists \omega_1, \omega_2 \in \mathcal{L}(G_S)$ such that

- $f \in \omega_1$ where $f \in \Sigma_{f_i}$, $i \in S$, $\omega_1 = s_1 t_1$, and s_1 ends with f,
- $f \notin \omega_2$,
- $P_{\{\Sigma_S,\Sigma_{o_S}\}}(\omega_1) = P_{\{\Sigma_S,\Sigma_{o_S}\}}(\omega_2)$, and
- $P_{\{\Sigma_S,\Sigma_{o_i}\}}(t_1)$ is arbitrarily long.

Let $\omega_1^i = P_{\{\Sigma_S,\Sigma_i\}}(\omega_1)$, $s_1^i = P_{\{\Sigma_S,\Sigma_i\}}(s_1)$, $t_1^i = P_{\{\Sigma_S,\Sigma_i\}}(t_1)$, and $\omega_2^i = P_{\{\Sigma_S,\Sigma_i\}}$ (ω_2). Hence we have the following:

- $\omega_1^i, \omega_2^i \in \mathcal{L}(G_i)$,
- $f \in \omega_1^i$ where $\omega_1^i = s_1^i t_1^i$ and s_1^i ends with f,
- $f \notin \omega_2^i$,
- $P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_1^i) = P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_2^i)$, and
- $P_{\{\Sigma_i,\Sigma_{o_i}\}}(t_1^i)$ is arbitrarily long.

Therefore $\mathcal{L}(G_i)$ is not modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}. ∎

4 Properties of Modular Diagnosability

The following lemmata are needed for the proof of Theorem 4. We define $ND = T \setminus D$ where $D = \{z : \mathcal{L}(G_z)$ is monolithically diagnosable w.r.t. Σ_{o_z} and $\Sigma_{f_z}\}$.

Lemma 1 *If $\forall i \in S \cap ND$, $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i} then $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$.*

Proof: We assume that $\forall i \in S \cap ND$, $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}. Therefore Definition 1 is satisfied $\forall i \in S \cap ND$ and $\forall f \in \Sigma_{f_i}$. By Part 2 of Theorem 2, Definition 1 is satisfied for $\forall i \in S \cap D$ and $\forall f \in \Sigma_{f_i}$. Thus $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and $(\Sigma_{f_z} : z \in S)$. ∎

Lemma 2 *If $\forall i \in T \cap ND$, $\exists S \subseteq T$ s.t. $i \in S$ and $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i} then $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.*

Proof: We prove the contrapositive: if $\mathcal{L}(G_T)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$ then $\exists i \in T \cap ND$ s.t. $\forall S \subseteq T$ with $i \in S$, $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}.

From Definition 1 and Part 2 of Theorem 2, $\mathcal{L}(G_T)$ not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$ implies that $\exists i \in T \cap ND$, $\exists s, s' \in \mathcal{L}(G_T)$, $\exists f \in \Sigma_{f_i}$ s.t. $f \in s$ and $f \notin s'$, $P_{\{\Sigma_T, \Sigma_{o_T}\}}(s) = P_{\{\Sigma_T, \Sigma_{o_T}\}}(s')$, and $P_{\{\Sigma_T, \Sigma_{o_i}\}}(t)$ is arbitrarily long. Also, $\forall S \subseteq T$ s.t. $i \in S$ we have the following: $P_{\{\Sigma_T, \Sigma_{o_S}\}}(s) = P_{\{\Sigma_T, \Sigma_{o_S}\}}(s')$ and $P_{\{\Sigma_T, \Sigma_{o_i}\}}(s) = P_{\{\Sigma_T, \Sigma_{o_i}\}}(s')$ since $P_{\{\Sigma_T, \Sigma_{o_T}\}}(s) = P_{\{\Sigma_T, \Sigma_{o_T}\}}(s')$. Furthermore, $P_{\{\Sigma_T, \Sigma_{o_i}\}}(s')$ is arbitrarily long since $P_{\{\Sigma_T, \Sigma_{o_i}\}}(s)$ is arbitrarily long. Let $s_x, s'_x \in \mathcal{L}(G_S)$ s.t. $s_x = P_{\{\Sigma_T, \Sigma_S\}}(s)$ and $s'_x = P_{\{\Sigma_T, \Sigma_S\}}(s')$. Then $f \in s_x$ and $f \notin s'_x$. Also $P_{\{\Sigma_S, \Sigma_{o_i}\}}(s_x)$, $P_{\{\Sigma_S, \Sigma_{o_i}\}}(s'_x)$ are arbitrarily long since $P_{\{\Sigma_T, \Sigma_{o_i}\}}(s)$, $P_{\{\Sigma_T, \Sigma_{o_i}\}}(s')$ are arbitrarily long.

In summary, $\exists i \in T \cap ND$ s.t. $\forall S \subseteq T$ with $i \in S$, $\exists s_x, s'_x \in \mathcal{L}(G_S)$, $\exists f \in \Sigma_{f_i}, f \in s_x, f \notin s'_x$, $P_{\{\Sigma_S, \Sigma_{o_S}\}}(s_x) = P_{\{\Sigma_S, \Sigma_{o_S}\}}(s'_x)$, and $P_{\{\Sigma_S, \Sigma_{o_i}\}}(s_x)$ is arbitrarily long. Therefore $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}.

■

Corollary 1 *If $\forall i \in T$, $\mathcal{L}(G_i)$ is monolithically diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}, then $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.*

Proof: Corollary 1 is a particular case of Lemma 2 when $S = \{i\}$. ■

An elementary F_m-indeterminate cycle, $m \in \{1, \ldots, M\}$, is formed by (i) a sequence of F_m-uncertain states and (ii) possibly several sequences of events that form the cycle and satisfy the F_m-indeterminate cycle definition. We call EIC_z, $z \in \{1, \ldots, Z\}$, such cycles and t^z_y, $y \in \{1, \ldots, Y_z\}$, their corresponding sequences of events, where Z represents the total number of elementary F_m-indeterminate cycles in the diagnoser G_{d_i}, $i \in ND$, and Y_z represents the total number of sequences of events that satisfy the indeterminate cycle definition for the particular EIC_z.

For each $i \in ND$, we number and name $SEQ_1, SEQ_2, \ldots, SEQ_{X_i}$ all sequences of events t^z_y, $y = 1, \ldots, Y_z$ and $z = 1, \ldots, Z$. Therefore, for each SEQ_x, $x \in \{1, \ldots, X_i\}$, there are one F_m-indeterminate cycle, one fault of type m, $m \in \{1, \ldots, M\}$, one corresponding sequence of states $Q^x = q_1 \ldots q_{N_x}$, and one sequence of events t^z_y, $z \in \{1, \ldots, Z\}$, $y \in \{1, \ldots, Y_z\}$, that form the cycle. We attach the label M_x, $x \in \{1, ..., X_i\}$, $i \in ND$, to states $q \in Q^x$ in G_{d_i}.

The following lemma is a specialized form of Lemma 2.

Lemma 3 *Consider $S \subseteq T$, SEQ_x, $x \in \{1, ..., X_i\}$, $i \in S \cap ND$, and any two arbitrarily long traces $\omega_x, \omega'_x \in \mathcal{L}(G_i)$ such that: (i) ω_x, ω'_x lead to the indeterminate* *cycle associated with SEQ_x in G_{d_i}; (ii) $P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x) = P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega'_x) = sSEQ_x s_1$ where $SEQ_x = s_1 s_2$; (iii) $f_m \in \omega_x$, $f_m \notin \omega'_x$, and f_m corresponds to the fault type associated with SEQ_x. If $\nexists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$, $P_{\{\Sigma_S,\Sigma_i\}}(\omega'_S) = \omega'_x$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long, then $\nexists \omega, \omega' \in L(G_T)$ such that $P_{\{\Sigma_T,\Sigma_i\}}(\omega) = \omega_x$, $P_{\{\Sigma_T,\Sigma_i\}}(\omega') = \omega'_x$, and $P_{\{\Sigma_T,\Sigma_{o_i}\}}(\omega)$ is arbitrarily long.*

Proof: We prove by contradiction. By assumption, $\exists S \subseteq T$, $\exists SEQ_x$, $x \in \{1, ..., X_i\}$, $i \in S \cap ND$, and there exist two arbitrarily long traces $\omega_x, \omega'_x \in \mathcal{L}(G_i)$ such that: (i) ω_x, ω'_x lead to the indeterminate cycle associated with SEQ_x in G_{d_i}; (ii) $P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x) = P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega'_x) = sSEQ_x s_1$ where $SEQ_x = s_1 s_2$; (iii) $f_m \in \omega_x$, $f_m \notin \omega'_x$, and f_m corresponds to the fault type associated with SEQ_x. Suppose that (iv) $\nexists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$, $P_{\{\Sigma_S,\Sigma_i\}}(\omega'_S) = \omega'_x$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long and (v) $\exists \omega, \omega' \in L(G_T)$ such that $P_{\{\Sigma_T,\Sigma_i\}}(\omega) = \omega_x$, $P_{\{\Sigma_T,\Sigma_i\}}(\omega') = \omega'_x$, and $P_{\{\Sigma_T,\Sigma_{o_i}\}}(\omega)$ is arbitrarily long.

By assumption (v) and the natural projection definition, $\exists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_T,\Sigma_S\}}(\omega) = \omega_S$, $P_{\{\Sigma_T,\Sigma_S\}}(\omega') = \omega'_S$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long. Furthermore we have $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = P_{\{\Sigma_S,\Sigma_i\}}[P_{\{\Sigma_T,\Sigma_S\}}(\omega)] = P_{\{\Sigma_T,\Sigma_i\}}(\omega) = \omega_x$ and $P_{\{\Sigma_S,\Sigma_i\}}(\omega'_S) = \omega'_x$. Therefore $\exists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$, $P_{\{\Sigma_S,\Sigma_i\}}(\omega'_S) = \omega'_x$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long, which yields the desired contradiction. ■

Remark 3 *When the hypothesis of Lemma 3 holds, we say that the indeterminate cycle associated with SEQ_x is "Not Reachable" in G_S and G_T. In other words, the coupling of module G_i with the remainder of the system results in the elimination of the traces in $\mathcal{L}(G_i)$ that lead to that indeterminate cycle.*

Corollary 2 *If the hypothesis of Lemma 3 holds for all x, $x \in \{1, ..., X_i\}$, then $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}.*

Lemma 4 *If $\exists i \in T$, $\exists S \subseteq T$ s.t. $i \in S$, $S^c = T \setminus S$, and $\exists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ such that:*

(i) *ω_S, ω'_S violate the modular diagnosability of $\mathcal{L}(G_S)$ w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S) = sSEQ_x s_1$, $SEQ_x = s_1 s_2$, $x \in \{1, ..., X_i\}$;*

(ii) *$\forall \sigma_x \in \omega_S$, $\forall \sigma_y \in \Sigma_{S^c}$, $\sigma_x \neq \sigma_y$;*

then $\mathcal{L}(G_T)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.

Proof: We have $G_T = G_S \parallel G_{S^c}$. Build $G_{\tilde{S}}$ s.t. $\mathcal{L}(G_{\tilde{S}}) := \overline{\{\omega_S, \omega'_S\}}$. Define $G_{\tilde{T}} := G_{\tilde{S}} \parallel G_{S^c}$. By the definition of $G_{\tilde{T}}$ and assumption (ii), $\exists \omega, \omega' \in \mathcal{L}(G_{\tilde{T}})$ s.t.

- $P_{\{\Sigma_{\tilde{T}}, \Sigma_{\tilde{S}}\}}(\omega) = \omega_S$, $P_{\{\Sigma_{\tilde{T}}, \Sigma_{\tilde{S}}\}}(\omega') = \omega'_S$,
- $P_{\{\Sigma_{\tilde{T}}, \Sigma_{o_{\tilde{T}}}\}}(\omega) = P_{\{\Sigma_{\tilde{T}}, \Sigma_{o_{\tilde{T}}}\}}(\omega')$,
- $f \in \omega$, $f \notin \omega'$, where $f \in \Sigma_{f_i}$, and
- $P_{\{\Sigma_{\tilde{S}}, \Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long because $P_{\{\Sigma_S, \Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long as it violates modular diagnosability by assumption (i).

Since $\mathcal{L}(G_{\tilde{T}}) \subseteq \mathcal{L}(G_T)$, $\omega, \omega' \in \mathcal{L}(G_{\tilde{T}})$ implies $\omega, \omega' \in \mathcal{L}(G_T)$ and therefore ω, ω' violate the modular diagnosability of $\mathcal{L}(G_T)$ w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.

■

We make the following observation regarding the proof of Lemma 4. Any $\omega \in \mathcal{L}(G_{\tilde{T}})$ is built from $\omega_S \in \mathcal{L}(G_{\tilde{S}})$ by interleaving events from Σ_{S^c} according to the transition structure of G_{S^c}. Hence, since $\Sigma_{\tilde{S}} \cap \Sigma_{S^c} = \emptyset$, we can build ω' from ω'_S by doing the same interleaving as when building ω from ω_S. The resulting ω' necessarily satisfies $P_{\{\Sigma_{\tilde{T}}, \Sigma_{o_{\tilde{T}}}\}}(\omega) = P_{\{\Sigma_{\tilde{T}}, \Sigma_{o_{\tilde{T}}}\}}(\omega')$.

Remark 4 *When the hypothesis of Lemma 4 holds, we say that the indeterminate cycle associated with SEQ_x is "Reachable" in G_S and G_T. In other words, the coupling of module G_i with the remainder of the system results in the propagation of the traces in $\mathcal{L}(G_i)$ that lead to that indeterminate cycle.*

Lemma 5 *If $\exists i \in T$, $\exists S \subseteq T$ s.t. $i \in S$, $S^c = T \setminus S$, and $\exists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ such that:*

(i) *ω_S, ω'_S violate the modular diagnosability of $\mathcal{L}(G_S)$ w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}, and $P_{\{\Sigma_S, \Sigma_{o_i}\}}(\omega_S) = sSEQ_x s_1$, $SEQ_x = s_1 s_2$, $x \in \{1, ..., X_i\}$;*

(ii) *$\exists \sigma_x \in \omega_S$ and $\exists \sigma_y \in \Sigma_{S^c}$ such that $\sigma_x = \sigma_y$;*

then $\mathcal{L}(G_T)$ may or may not be modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}.

Proof: We have $G_T = G_S \parallel G_{S^c}$. By assumption (ii), two (exhaustive) cases are possible. Define $\omega, \omega' \in \Sigma_T^*$ such that:

- $P_{\{\Sigma_T, \Sigma_S\}}(\omega) = \omega_S$, $P_{\{\Sigma_T, \Sigma_S\}}(\omega') = \omega'_S$,

- $P_{\{\Sigma_T,\Sigma_{o_T}\}}(\omega) = P_{\{\Sigma_T,\Sigma_{o_T}\}}(\omega')$,

- $f \in \omega$, $f \notin \omega'$, where $f \in \Sigma_{f_i}$, and

- $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long.

Case 1: If such ω, ω' exist in $\mathcal{L}(G_T)$, then $\mathcal{L}(G_T)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}.

Case 2: On the other hand, if no such ω, ω' exist then $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}. ■

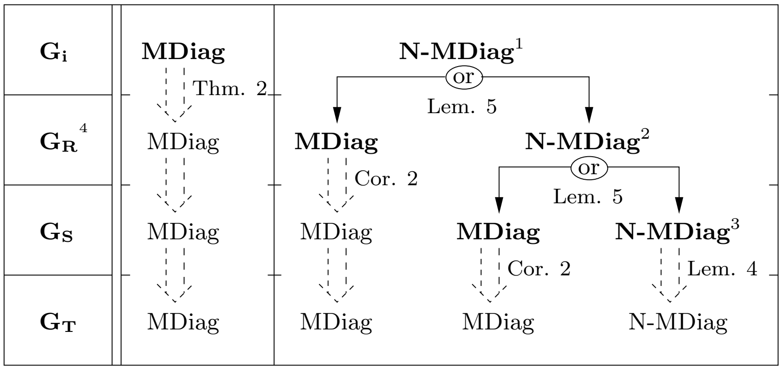
We discuss the intuition behind Lemma 5. Any $\omega \in \mathcal{L}(G_T)$ is built from $\omega_S \in \mathcal{L}(G_S)$ by interleaving events from Σ_{S^c} according to the transition structure of G_{S^c}. Hence, since $\Sigma_S \cap \Sigma_{S^c} \neq \emptyset$, any event $\sigma_x \in \Sigma_S \cap \Sigma_{S^c}$ may or may not be synchronized during the parallel composition $G_T = G_S \parallel G_{S^c}$. The existence of the traces ω, ω' in the proof of Lemma 5 depends on the outcome of such synchronization.

Remark 5 *If $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i} then G_S necessarily satisfies the hypotheses of Lemmata 4 or 5.*

In Fig. 2, we depict the implications of Theorem 2, Corollary 2, and Lemmata 4, 5. The figure shows that if $\mathcal{L}(G_i)$ is modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}, or $\mathcal{L}(G_S)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}, then $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}. If $\mathcal{L}(G_i)$ is not modularly diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}, or $\mathcal{L}(G_S)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in S)$ and Σ_{f_i}, then the output on the modular diagnosability of $\mathcal{L}(G_T)$ w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i} is uncertain unless G_S satisfies Lemma 4.

5 Test for Modular Diagnosability

In Section 3 we presented the notion of modular diagnosability and conditions necessary and sufficient to guarantee it. In the case of large modular discrete event systems, the computational complexity associated with the parallel composition of subsystems and the construction of the diagnoser for the resulting (monolithic) system is forbidding. To deal with this problem we propose a novel approach that tests modular diagnosability by incorporating incrementally, in a systematic manner, subsystems into the test. We prove that our approach provides the correct



Legend:

MDiag	:Modularly Diagnosable w.r.t. $(\Sigma_{o_z} : z \in X)$ and Σ_{f_i}, where G_X is the considered system.
N-MDiag:	Not Modularly Diagnosable w.r.t. $(\Sigma_{o_z} : z \in X)$ and Σ_{f_i}.
○⇉	:Two Possible Outputs (need to construct the pointed module).
⇢	:Direct Implication (no computation or construction needed).
Cor.	:Corollary.
Lem.	:Lemma.
Thm.	:Theorem.

Notes:

1: Subsystem G_i necessarily satisfies hypothesis (ii) of Lemma 5.
2: We assume that G_R satisfies hypothesis (ii) of Lemma 5.
3: We assume that G_S satisfies hypothesis (ii) of Lemma 4.
4: $R \subset S \subseteq T$, $i \in R$.

Figure 2: Properties of Modular Diagnosability

answer to the question "Is $\mathcal{L}(G_T)$ modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in \ T)$?" in a finite number of steps. We proceed as follows. In Section 5.1 we present the algorithm; in Section 5.2 we state and prove its properties; in Sections 5.3 and 5.4 we present a discussion of the key steps of the algorithm and online diagnosis, respectively.

5.1 Modular Diagnosability Algorithm

We present the detailed statement of our Modular Diagnosability Algorithm (MDA). For the sake of clarity, MDA is broken into three algorithms. Algorithm 1 is the core of MDA; it calls Algorithm 2 to perform preliminary steps involving indeterminate cycles that could lead to a violation of modular diagnosability. Algorithm 1 also calls Algorithm 3 where the incremental analysis of each indeterminate cycle is performed. (For the sake of simplicity, some optional improvements related to the computational performance of MDA are not included in the presentation below.)

Algorithm 1 MDA

1.) Let $T = \{1, \ldots, I\}$. Construct the local diagnosers G_{d_i}, $i \in T$, and search for indeterminate cycles. If, $\forall i \in T$, $\mathcal{L}(G_i)$ is monolithically diagnosable w.r.t. Σ_{o_i} and Σ_{f_i}, i.e., none of the local diagnosers G_{d_i} have F-indeterminate cycles, then stop and declare $\mathcal{L}(G_T)$ modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$. Else, go to Step 2.

*2.) Let $ND = T \setminus D$, where $D = \{z : \mathcal{L}(G_z)$ is monolithically diagnosable w.r.t. Σ_{o_z} and $\Sigma_{f_z}\}$. Call **Preliminary Function**. For each local diagnoser G_{d_i}, $i \in ND$, and for each sequence of traces SEQ_x, $x \in \{1, ..., X_i\}$, perform the Steps 2-a to 2-d:*

2-a.) Mark with the label M_x, $x \in \{1, ..., X_i\}$, $i \in ND$, the states $q \in Q^x$ in G_{d_i}. The label M_x stands for "State of G_{d_i} part of the indeterminate cycle associated with SEQ_x".

2-b.) Construct

$$G_{CMi} = Obs(G_{d_i}, \Sigma_{CM_i}). \tag{9}$$

A state of G_{CMi} is marked with label M_x if one or more of its state components are marked with M_x.

2-c.) Construct

$$G_{ICM_x} = CoAc(G_{CMi}, M_x). \tag{10}$$

The resulting event set of machine G_{ICM_x} is denoted by Σ_{ICM_x}. Enlarge the set of events Σ_{ICM_x} by adding $\Sigma_{CM_i} \setminus \Sigma_{ICM_x}$ to it. The machine G_{ICM_x} and its newly associated event set Σ_{CM_i} are hereafter represented by the notation $(G_{ICM_x}, \Sigma_{CM_i})$.

*2-d.) Call **Reachability Function** with argument $\{i, SEQ_x, G_{ICM_x}, \Sigma_{ICM_x}, \Sigma_{CM_i}\}$. If the Reachability Function returns "Reachable" then stop and declare $\mathcal{L}(G_T)$ not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.*

3.) Stop and declare $\mathcal{L}(G_T)$ modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$. $\Diamond$

Algorithm 2 - Preliminary Function

I.) For each $i \in ND$, do the following:

i.) Call EIC_z, $z \in \{1, \ldots, Z\}$, the elementary[3] indeterminate cycles in G_{d_i} and t^z_y, $y \in \{1, \ldots, Y_z\}$, their corresponding sequences of events, where Z represents the total number of elementary indeterminate cycles in diagnoser G_{d_i}, $i \in ND$, and

[3] A cycle is called elementary if no state appears more than once in it.

Y_z represents the total number of sequences of events that satisfy the indeterminate cycle definition for the particular EIC_z.

ii.) Number and name $SEQ_1, SEQ_2, \ldots, SEQ_{X_i}$ all sequences of events t_y^z, $y = 1, \ldots, Y_z$ and $z = 1, \ldots, Z$. To each SEQ_x, $x \in \{1, \ldots, X_i\}$, associate its corresponding F_m-indeterminate cycle, $m \in \{1, \ldots, M\}$, its corresponding sequence of states $Q^x = q_1 \ldots q_{N_x}$, and its corresponding sequence of events t_y^z, $z \in \{1, \ldots, Z\}$, $y \in \{1, \ldots, Y_z\}$, that form the cycle.

II) Return to MDA with SEQ_x and Q^x, $\forall x \in \{1, \ldots, X_i\}$, $\forall i \in ND$. $\Diamond$

Algorithm 3 - Reachability Function $\{i,\ SEQ_x,\ G_{ICM_x},\ \Sigma_{ICM_x},\ \Sigma_{CM_i}\}$

A.) Let $c := 1$, $B_c^x = \{i\}$, $S_c = B_c^x$, and

$$\tilde{s}^x = P_{\{\Sigma_{o_i}, \Sigma_{CM_{o_i}}\}}(SEQ_x). \tag{11}$$

Let $c := c + 1$,

$$B_c^x = \{l : [\Sigma_{CM_l} \cap \Sigma_{ICM_x} \neq \emptyset \vee (l \in B_{c-1}^x)], l \in T\}, \tag{12}$$

and

$$S_c = B_c^x \setminus B_{c-1}^x. \tag{13}$$

B.) Construct

$$G_{mod_c^x} = (G_{ICM_x}, \Sigma_{CM_i}) \parallel (\|_{l \in B_c^x, l \neq i}\, G_{CM\,l}). \tag{14}$$

If there does not exist in $G_{mod_c^x}$ a cycle of states labeled M_x then return to MDA; otherwise denote by $s_1^c, s_2^c, \ldots, s_P^c$ the sequences of events that describe such cycles and go to step C.

C.) $\forall p \in \{1, \ldots, P\}$, let

$$\tilde{s}_p^c = P_{\{\Sigma_{CM_{o_S}}, \Sigma_{CM_{o_i}}\}}(s_p^c). \tag{15}$$

If $\exists p \in \{1, \ldots, P\}$ such that $\tilde{s}^x = \tilde{s}_p^c$ or if $\exists s'^x, s''^x$, and $p \in \{1, \ldots, P\}$ such that $\tilde{s}^x = s'^x s''^x$ and $s''^x s'^x = \tilde{s}_p^c$, then go to Step D; otherwise return to MDA.

D.) Construct

$$\widetilde{G}_c = CoAc(G_{mod_c^x}, M_x). \tag{16}$$

Let $\widetilde{\Sigma}_c$ be the event set of $\widetilde{G}_c$.

E.) Let $c := c + 1$ and

$$B_c^x = \{l : [(\Sigma_{CM_l} \cap \widetilde{\Sigma}_{c-1} \neq \emptyset) \vee (l \in B_{c-1}^x)], l \in T\}. \tag{17}$$

Define

$$S_c = B_c^x \setminus B_{c-1}^x. \tag{18}$$

If $S_c \neq \emptyset$ then go to step B; otherwise declare the indeterminate cycle associated with SEQ_x "Reachable" and return to MDA. $\Diamond$

5.2 Properties of MDA

Theorem 3 *MDA returns an answer in a finite number of steps.*

Proof: Since there is a finite number I of subsystems and B_c^x is monotonically increasing by equation 17, the **Reachability Function** returns an answer in a finite number of steps for every sequence of events SEQ_x. Since there is a finite number of sequences of events SEQ_x, MDA returns an answer in a finite number of steps. $\blacksquare$

Theorem 4 *MDA returns the correct answer, namely, whether $\mathcal{L}(G_T)$ is or is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.*

Proof: We prove that Steps 1, 2-d, and 3 of MDA and steps B, C, and E of the Reachability Function return the correct answer. The correctness of Steps 2-d and 3 of MDA depends on the correctness of the Reachability Function. Thus, we proceed as follows. We first prove that the Reachability Function returns the correct answer to the question: "Is the indeterminate cycle associated with SEQ_x reachable in the global system behavior $\mathcal{L}(G_T)$?". Then we prove the correctness of Steps 1, 2-d, and 3 of MDA, using the correctness of the Reachability Function.

Correctness of Step B of the Reachability Function: By construction, $G_{mod_c^x}$ is composed of projections of subsystems G_z, where $z \in S$ and $S = B_c^x$. As a reminder, the states of the indeterminate cycle associated with SEQ_x in G_{d_i} are marked with the label M_x and by construction the states of the machines G_{d_S} and $G_{mod_c^x}$ are marked with labels M_x if one or more state components are marked with the label M_x. Consider any two arbitrarily long traces $\omega_x, \omega'_x \in \mathcal{L}(G_i)$ such that: (i) $f_m \in \omega_x$ where f_m corresponds to the fault type associated with SEQ_x; (ii) $f_m \notin \omega'_x$; (iii) ω_x, ω'_x lead to the indeterminate cycle associated with SEQ_x in

G_{d_i} (where states Q^x are labeled M_x); and (iv) $P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x) = P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega'_x)$. If there does not exist in $G_{mod^x_c}$ a cycle of states labeled M_x then, by construction of $G_{mod^x_c}$, $\nexists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$, $P_{\{\Sigma_S,\Sigma_i\}}(\omega'_S) = \omega'_x$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long. Hence, by Lemma 3 and Remark 3, the indeterminate cycle associated with SEQ_x is "Not Reachable" and we return to MDA. If, in Step B, there exists in $G_{mod^x_c}$ a cycle of states labelled M_x then we cannot conclude on the reachability of the indeterminate cycle; thus we number $s^c_1, s^c_2, \ldots, s^c_p$ the sequences of events that describe such cycles and go to Step C.

Correctness of Step C of the Reachability Function: Consider any two arbitrarily long traces $\omega_x, \omega'_x \in \mathcal{L}(G_i)$ such that: (i) $f_m \in \omega_x$ where f_m corresponds to the fault type associated with SEQ_x; (ii) $f_m \notin \omega'_x$; (iii) ω_x, ω'_x lead to the indeterminate cycle associated with SEQ_x in G_{d_i}; and (iv) $P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega_x) = P_{\{\Sigma_i,\Sigma_{o_i}\}}(\omega'_x) = sSEQ_x s_1$ where $SEQ_x = s_1 s_2$. From Eqs 11 and 15, we have that $\tilde{s}^x = P_{\{\Sigma_{o_i},\Sigma_{CM_{o_i}}\}}(SEQ_x)$ and, $\forall p \in \{1,\ldots,P\}$, $\tilde{s}^c_p = P_{\{\Sigma_{CM_{o_S}},\Sigma_{CM_{o_i}}\}}(s^c_p)$. If $\nexists p \in \{1,\ldots,P\}$ such that $\tilde{s}^x = \tilde{s}^c_p$ and if $\nexists s'^x, s''^x$, and $p \in \{1,\ldots,P\}$ such that $\tilde{s}^x = s'^x s''^x$ and $s''^x s'^x = \tilde{s}^c_p$, then, by construction of $G_{mod^x_c}$, $\nexists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ such that $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$, $P_{\{\Sigma_S,\Sigma_i\}}(\omega'_S) = \omega'_x$, and $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long. Hence, by Lemma 3 and Remark 3, the indeterminate cycle associated with SEQ_x is "Not Reachable" and we return to MDA. If $\exists p \in \{1,\ldots,P\}$ such that $\tilde{s}^x = \tilde{s}^c_p$ or if $\exists s'^x, s''^x$, and $p \in \{1,\ldots,P\}$ such that $\tilde{s}^x = s'^x s''^x$ and $s''^x s'^x = \tilde{s}^c_p$, then we cannot conclude on the reachability of the indeterminate cycle and go to Step D.

Correctness of Step E of the Reachability Function: Consider any two arbitrarily long traces $\omega_x, \omega'_x \in \mathcal{L}(G_i)$ as defined above in the proof of correctness of Step C. Consider $G_{mod^x_{c-1}}$, which is composed of projections of subsystems G_z, where $z \in S$ and $S = B^x_{c-1}$. Since there exist in $G_{mod^x_{c-1}}$ one or more cycles of states labeled M_x that correspond to a projection of SEQ_x, then, by construction of $G_{mod^x_{c-1}}$, $\exists \omega_S, \omega'_S \in \mathcal{L}(G_S)$ that violate the modular diagnosability of $\mathcal{L}(G_S)$ w.r.t. $(\Sigma_{o_z} : z \in S)$ and f_m, and moreover satisfy the following conditions: (i) $f_m \in \omega_S$ where f_m is the fault associated with SEQ_x; (ii) $f_m \notin \omega'_S$; (iii) $P_{\{\Sigma_S,\Sigma_i\}}(\omega_S) = \omega_x$; (iv) $P_{\{\Sigma_S,\Sigma_i\}}(\omega'_S) = \omega'_x$; (v) $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S) = P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega'_S) = sSEQ_x s_1$, $SEQ_x = s_1 s_2$, $x \in \{1, ..., X_i\}$; and (vi) $P_{\{\Sigma_S,\Sigma_{o_i}\}}(\omega_S)$ is arbitrarily long. Therefore hypothesis (i) of Lemma 4 is satisfied. The condition $S_c = \emptyset$ implies that there does not exist any subsystem G_l, $l \notin S$, that contains common events with the machine $\widetilde{G}_{c-1}$; thus hypothesis (ii) of Lemma 4 is satisfied. Then, by Lemma 4 and Remark 4, we declare the indeterminate cycle associated with SEQ_x "Reachable" in G_S and G_T and return to MDA. If $S_c \neq \emptyset$ then we cannot decide on the reachability of the indeterminate cycle (cf. Lemma 5) and go to Step B.

We have proven that the Reachability Function returns the correct answer to the question: "Is the indeterminate cycle associated with SEQ_x reachable in the global system behavior $\mathcal{L}(G_T)$?". We use this to complete the proof of the correctness of MDA.

Correctness of Step 1 of MDA: The correctness of Step 1 follows directly from Corollary 1.

Correctness of Step 2-d of MDA: If, in the Reachability Function, we declare the indeterminate cycle associated with SEQ_x "Reachable" then we conclude that, by Lemma 4, $\mathcal{L}(G_T)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}, which also implies that $\mathcal{L}(G_T)$ is not modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$.

Correctness of Step 3 of MDA: If, in the Reachability Function, we declare the indeterminate cycles associated with SEQ_x, $x = 1, \ldots, X_i$, "Not Reachable" then, by Corollary 2, we conclude that $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and Σ_{f_i}. If the above is true for all $i \in ND$ then, by Lemma 1, $\mathcal{L}(G_T)$ is modularly diagnosable w.r.t. $(\Sigma_{o_z} : z \in T)$ and $(\Sigma_{f_z} : z \in T)$. ■

5.3 Discussion

To give more insight into MDA, we present its flowchart, cf. Fig. 3, and discuss the key steps of its operation. The procedure starts by building local diagnosers for each module of the system and checking if they are monolithically diagnosable or not. Clearly, if each individual module is (monolithically/modularly) diagnosable, then the complete system is both monolithically and modularly diagnosable. Therefore, we need only focus on the modules that are not diagnosable in order to find out if a corresponding violation of modular diagnosability occurs or not when the given module is coupled with the rest of the system.

To do so, we concentrate on the traces that form indeterminate cycles in local diagnosers. We need to test these traces one by one and determine if they survive in the diagnoser of the complete system, without constructing this monolithic diagnoser.

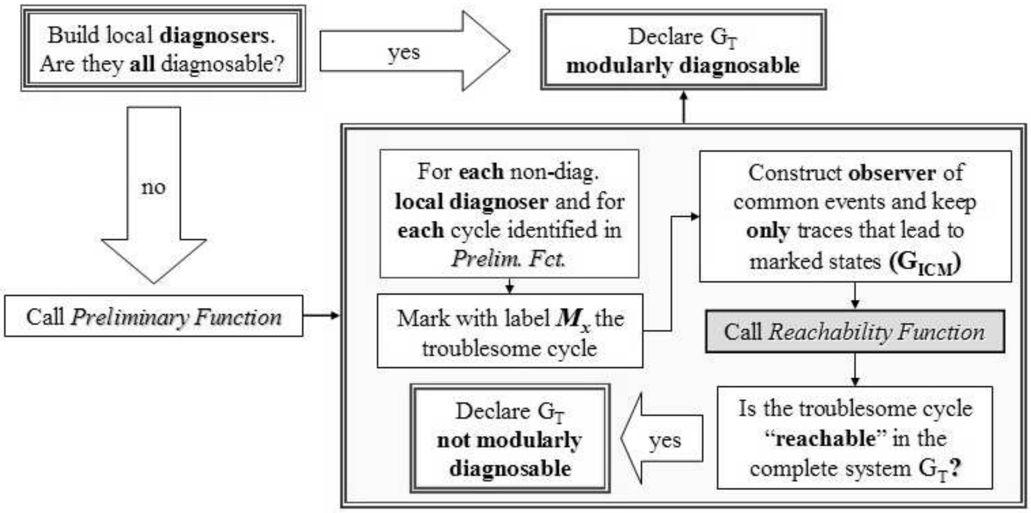

Build local **diagnosers**.
Are they **all** diagnosable?
yes
Declare G_T **modularly diagnosable**
no
Call *Preliminary Function*
For **each** non-diag. **local diagnoser** and for **each** cycle identified in *Prelim. Fct.*
Mark with label M_x the troublesome cycle
Construct **observer** of common events and keep **only** traces that lead to marked states (G_{ICM})
Call *Reachability Function*
Is the troublesome cycle "**reachable**" in the complete system G_T?
yes
Declare G_T **not modularly diagnosable**

Figure 3: Algorithm Flowchart

The testing procedure starts by selecting one indeterminate cycle in a given (non-diagnosable) local diagnoser and isolating all the so-called "troublesome traces", namely, the traces that lead to and form indeterminate cycles in local diagnosers; there could be more than one troublesome trace depending on the accessibility of the indeterminate cycle in the transition structure of the local diagnoser. For each troublesome trace, we select all other modules that contain an event common with the ones in the troublesome trace, build machines (observers for common events - cf. Step 2-b of Algorithm 1) for each module selected, perform the parallel composition of these machines, and finally check if the indeterminate cycle under consideration survives (cf. Step B of Algorithm 3). If it does not survive at this stage then it will not survive if we were to construct the monolithic diagnoser. However, if it does survive, then we need to consider the effect of other modules, namely those that have common events with the result of the above parallel composition. This is the heart of the incremental procedure performed in Algorithm 3. We iterate using essentially the same steps as described above - cf. the loop formed by Steps B through E of Algorithm 3.

The incremental procedure in Algorithm 3 ceases to add local modules and consequently stops when either (i) it has been determined that the indeterminate cycle under consideration is not reachable in the complete system - if this holds for all indeterminate cycles then the monolithic system is modularly diagnosable or

(ii) no other module is added in the incremental process at Step E of Algorithm 3 - in which case the monolithic system is not modularly diagnosable. Note that the latter conclusion can be reached *without* having to consider all modules in the set T. This potential computational gain depends on the structure of the machine $G_{mod_c^x}$ and its co-accessibility properties with respect to the indeterminate cycle under consideration, as determined in Steps C and D of Algorithm 3.

Figure 4 describes the architecture of the modular diagnosability decision process with respect to Module 1. The process has to be repeated for all modules in the system in order to infer on the modular diagnosability of the monolithic system.

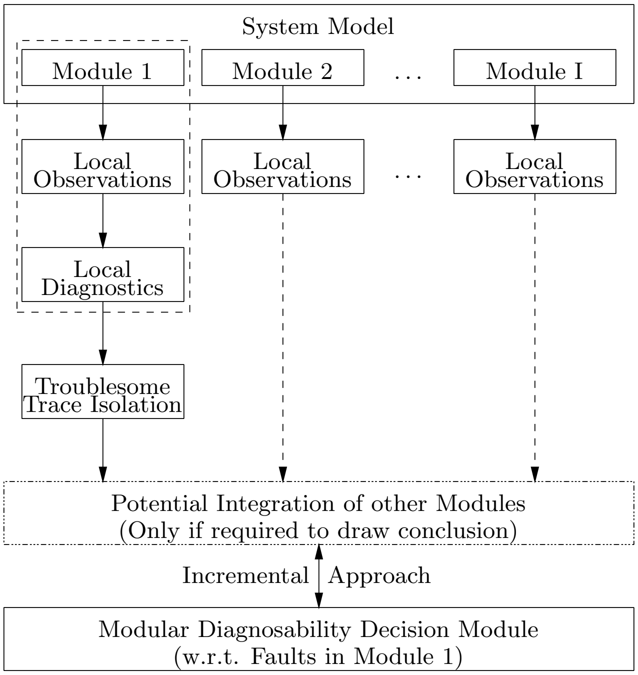

System Model
Module 1
Module 2
...
Module I
Local Observations
Local Observations
...
Local Observations
Local Diagnostics
Troublesome Trace Isolation
Potential Integration of other Modules
(Only if required to draw conclusion)
Incremental Approach
Modular Diagnosability Decision Module
(w.r.t. Faults in Module 1)

Figure 4: Modular Diagnosability Verification

The main feature exploited within MDA is the incremental addition of modules by considering only those that are necessary to reach a decision on the modular diagnosability of the monolithic system. Depending on the structure of the system, MDA may consider a smaller number of modules rather than all of T when performing parallel composition operations. Nevertheless, the worst case can possibly occur and yield

$$|B_c^x| = |T|, \tag{19}$$

which implies that every system module is considered in the parallel composition for obtaining $G_{mod_c^x}$. However, it should be emphasized that the parallel compositions performed in MDA involve only machines with common events (observers) built from the individual modules. Therefore the resulting automata are likely to have a smaller state spaces than corresponding ones built using entire system modules. It should also be emphasized that in MDA, we construct diagnosers only for the individual G_i modules. The process of building diagnosers may result in a large state space growth in the worst case, since subsets of states of the model under consideration must be accounted for. Therefore, it is clearly advantageous to build diagnosers for local subsystems as opposed to building the diagnoser of the monolithic system G_T. The same advantage holds for online diagnosis, cf. Section 5.4. Practical experience with models of modular discrete event systems will be key to gaining insight into the potential computational advantages of MDA over a monolithic approach. This is the object of current research.

Finally, the whole procedure followed in MDA not only exploits the modular structure of the given system, but also may provide insight into causes of non diagnosability and possible remedies for it through coupling of system modules with one another. Thus MDA could be a useful tool in modular system design.

5.4 Online Diagnosis

If the system G_T, $T = \{1, \ldots, I\}$, is modularly diagnosable, we can perform *online diagnosis* by simply running the local diagnosers G_{d_i}, $i \in T$, at each local site, cf. Fig. 5. We know from the property of modular diagnosability that even if the local diagnoser G_{d_i} contains an indeterminate cycle, the local observations at site i will never stay forever in this cycle when the complete system G_T is functioning.

If MDA outputs that the system is not modularly diagnosable, then we can still partially diagnose the system online as follows. Each indeterminate cycle is associated to a fault $f_m \in \Sigma_{f_T}$, $m \in \{1, \ldots, M\}$. From MDA, we know which indeterminate cycles of G_{d_i} are reachable and which are blocked. If the local diagnoser G_{d_i} contains an indeterminate cycle that is "reachable" in the complete system G_T, then local observations will stay forever in this cycle. Therefore we mark as "f_m inactive" the states of G_{d_i} that correspond to the reachable indeterminate cycle associated to the fault f_m. We run at each local site the modified version of the local diagnoser G_{d_i}, i.e., the one with the labels "f_m inactive". Then, when a local diagnoser reaches an "f_m inactive" state, the local site broadcasts that there is a

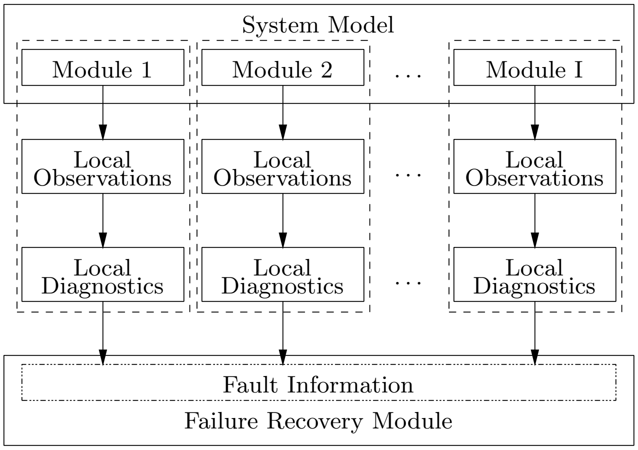


Figure 5: Online Modular Diagnosis

potential fault f_m that cannot be diagnosed with certainty.

6 Conclusion

We have proposed a notion of modular diagnosability that is suitable for systems that have modular structure expressed in terms of the parallel composition of individual automata, where each individual automaton models the behavior of the system component at the corresponding site. If modular diagnosability holds, then on-line fault diagnosis of the modular system is straightforward as it suffices to run a local diagnoser at each site, where the local diagnoser is built using only the local automaton model and ignoring the remainder of the system model. It is guaranteed that, after sufficient local observable events, any fault at a site will be diagnosed. However, the verification of modular diagnosability requires in general the joint consideration of multiple system components. We have presented an algorithm that correctly verifies if modular diagnosability holds or not and does so by incrementally including the automata models of other system components only if they are required to draw definitive conclusions about the diagnosability of faults within a given system component. This property of the algorithm makes it potentially computationally advantageous for large complex modular systems. Moreover, even if the modular diagnosability property of the algorithm does not hold, the algorithm provides insight into possible structural changes to the system in order to render it modularly diagnosable.

Acknowledgments

This research is supported in part by NSF grants ECS-0080406, CCR-0082784, and CCR-0325571, and by a grant from the Xerox University Affairs Committee.

References

Cassandras, C. G. and S. Lafortune (1999). *Introduction to Discrete Event Systems*. Kluwer Academic Publishers.

Contant, O., S. Lafortune and D. Teneketzis (2002). Failure diagnosis of discrete event systems: The case of intermittent faults. In: *Proc. 41st IEEE Conf. on Decision and Control*. Las Vegas, NV, USA. pp. 4006–4011.

Coolen, R. and H. Luiijf (2002). Intrusion detection: Generics and state-of-the-art. Technical Report RTO-TR-049. Research and Technology Organisation, NATO. Neuilly-sur-Seine, France.

Debouk, R., R. Malik and B. Brandin (2002). A modular architecture for diagnosis of discrete event systems. In: *Proc. 41st IEEE Conf. on Decision and Control*. Las Vegas, NV, USA. pp. 417–422.

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

Fabre, E., A. Benveniste and C. Jard (2002). Distributed diagnosis for large discrete event dynamic systems. In: *Proc. 15th IFAC World Congress*. Barcelona, Spain.

García, E., F. Morant, R. Blasco-Gimnez, A. Correcher and E. Quiles (2002). Centralized modular diagnosis and the phenomenon of coupling. In: *Proc. of the 2002 International Workshop on Discrete Event Systems - WODES'02*. Zaragoza, Spain. pp. 161–168.

Genc, S. and S. Lafortune (2003). Distributed diagnosis of discrete-event systems using Petri nets. In: *Proc. 2003 International Conf. on Application and Theory of Petri Nets*. Eindhoven, The Netherlands. pp. 316–336.

Holloway, L. E. and S. Chand (1994). Time templates for discrete event fault monitoring in manufacturing systems. In: *Proc. 1994 American Control Conference.* Baltimore, MD, USA. pp. 701–706.

Lafortune, S., D. Teneketzis, M. Sampath, R. Sengupta and K. Sinnamohideen (2001). Failure diagnosis of dynamic systems: An approach based on discrete event systems. In: *Proc. 2001 American Control Conference*. Arlington, VA, USA. pp. 2058–2071.

Ricker, L. S. and E. Fabre (2000). On the construction of modular observers and diagnosers for discrete event systems. In: *Proc. 39th IEEE Conf. on Decision and Control*. Sydney, Australia. pp. 2240–2244.

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

Sampath, M., R. Sengupta, K. Sinnamohideen S. Lafortune and D. Teneketzis (1996). Failure diagnosis using discrete event models. *IEEE Trans. Control Systems Technology* **4**(2), 105–124.

Su, R., W. M. Wonham, J. Kurien and X. Koutsoukos (2002). Distributed diagnosis for qualitative systems. In: *Proc. of the 2002 International Workshop on Discrete Event Systems - WODES'02*. Zaragoza, Spain. pp. 169–174.

Yoo, T.-S. (2003). Private Communication.