Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Computation of Diagnosable Fault-Occurrence Indices for Systems with Repeatable-Faults

Changyan Zhou and Ratnesh Kumar (czhou,rkumar@iastate.edu)
Department of Electrical & Computer Engineering
Iowa State University, Ames, IA 50014

***Abstract*— We study discrete-event systems prone to such faults which can occur repeatedly. For the diagnosis of repeatable faults, [5] introduced the notion of ℓ-diagnosability requiring the diagnosis of the ℓth occurrence of a fault within a bounded delay. The present paper studies the identification of the set of all indices ℓ for which the system is ℓ-diagnosable. (These are precisely the occurrence indices for which a repeatable-fault can be diagnosed.) We first present a test to verify whether a fault is diagnosable for each occurrence index, i.e., whether it is "$\forall\ell$-diagnosable". For systems that fail this test, we further present a method to compute the set of all indices ℓ for which the system is not ℓ-diagnosable.**

Keywords: Discrete event systems, failure diagnosis, repeated failures.

I. INTRODUCTION

Failure diagnosis of discrete event systems (DESs) has been extensively studied (see for example [9], [8], [1], [4]) to determine whether a fault occurred sometimes in past. Certain faults can occur repeatedly, such as the intermittent or non-persistent faults. For example in a discrete flow network, such as a manufacturing facility or a communication network, the same routing violation may occur repeatedly. It is possible that certain occurrences of a repeatable fault are diagnosable, whereas other occurrences are not diagnosable. In order to study the diagnosis of various occurrences of a fault, the notion of repeated failure diagnosis was introduced by Jiang-Kumar-Garcia [5]. Specifically, they introduced the notions of ℓ-diagnosability (ℓth failure diagnosability), $[1,\ell]$-diagnosability (1 through ℓ failures diagnosability), and $[1,\infty]$-diagnosability (1 through ∞ failures diagnosability). They provided polynomial algorithms for checking these various notions of repeated diagnosability, and also presented a method to construct a diagnoser for the on-line diagnosis of repeated failures. Algorithms of an order better complexity were later reported in [11]. A temporal logic based approach for diagnosing the occurrences of a repeatable-fault is proposed in [3].

The property of $[1,\infty]$-diagnosability guarantees that each occurrence of a repeatable-fault be detected within a bounded delay of its occurrence, and the bound is uniform in the sense that it does not depend on the fault-occurrence number. While this is the most desirable property to have, it is possible that certain systems don't possess this property. So it is desirable to know the set of those fault-occurrence indices ℓ for which a system is ℓ-diagnosable. These are precisely the fault-occurrence indices for which a repeatable-fault can be diagnosed. We first provide an algorithm to check whether the set of all indices for which a system is ℓ-diagnosable spans the set of all numbers, i.e., whether a system is ℓ-diagnosable for every $\ell \geq 1$, which we refer to as "$\forall\ell$-diagnosable". $\forall\ell$-diagnosability is weaker than $[1,\infty]$-diagnosability as it does not require the diagnosis delay bound to be uniform with respect to ℓ, i.e., it is possible that each occurrence of a fault is detectable within a bounded delay but the delay bound grows larger as the fault-occurrence index ℓ grows higher. For the class of systems that are not $\forall\ell$-diagnosable, we next present an algorithm to determine the set of indices ℓ for which the system is not ℓ-diagnosable. This algorithm examines the conditions of $\forall\ell$-diagnosability we have obtained, and identifies all possible ways in which the paths of the underlying testing automaton can violate those conditions.

The notion of $\forall\ell$-diagnosability considered here is not the same as the notion of non-uniform $[1,\infty]$-diagnosability considered in [11] since the "non-uniformity" in the latter refers to the diagnosis delay bound being a function of the failure-traces. In other words, the diagnosis delay bound for the ℓth occurrence of a fault can be different for different traces. In contrast in the case of $\forall\ell$-diagnosability, the diagnosis delay bound for the ℓth occurrence of a fault is the same across all failure-trace, but it does depend on the fault-occurrence index ℓ. In other words, in order for the detection of the ℓth occurrence of a fault, the diagnosis system needs to wait the same amount regardless of the failure-trace executed by the system. This is a convenient feature to have since due to the partial observation of events it is generally not known what trace the system has executed, and how long need a diagnoser wait before it can arrive at a diagnosis decision.

This paper adopts a "state-based" approach (i.e., the occurrence of a failure is specified as the visit of a faulty state). (An "event-based approach" can be transformed to a "state-based" approach [5].) Also, without loss of generality, we assume that there is only one fault, for it is the case that a system is diagnosable with respect to a given set of faults if and only if it is diagnosable with respect to each of the faults individually [5]. For space consideration, all proofs are omitted.

The research was supported in part by the National Science Foundation under the grants NSF-ECS-0218207, NSF-ECS-0244732, NSF-EPNES-0323379, and NSF-0424048, and a DoD-EPSCoR grant through the Office of Naval Research under the grant N000140110621.

II. Notation and Preliminaries

A discrete event system G is modeled by a four tuple: $G = (X, \Sigma, \alpha, x_0)$, where X is its set of states, Σ is its set of events, $\alpha : X \times \Sigma \cup \{\epsilon\} \to 2^X$ is its state transition function, and $x_0 \in X$ is its initial state. For an event set Σ, we use $\overline{\Sigma}$ to denote $\Sigma \cup \{\epsilon\}$. A triple $(x, \sigma, x') \in X \times \overline{\Sigma} \times X$ is called a *transition* if $x' \in \alpha(x, \sigma)$; if $\sigma = \epsilon$, the transition is called an ϵ-transition.

A finite state-trace $\pi = (x_1, \cdots, x_n)$ is a *path contained* in G if for all $1 \leq i < n$ there exists $\sigma_i \in \overline{\Sigma}$ such that the transition (x_i, σ_i, x_{i+1}) is in G. In this case we say that a *transition on* σ_i *is contained in* π, and the *state* x_i *is contained in* π (denoted $x_i \in \pi$). The length of π is the number of states contained in π, denoted $|\pi|$. A *path* π *is generated* by G if π is contained in G and it starts from the initial state of G. A *cycle* is a path $cl = (x_1, \cdots, x_n)$ such that the begin state x_1 is a successor of the end state x_n. An *elementary path* is a path with no repeated states, and an *elementary cycle* is an elementary path that is a cycle. We use $Cl(x)$ to denote the set of all elementary cycles containing state x, and $Tr(x)$ to denote the set of all elementary paths containing state x. For a cycle cl, we define $Cl(cl) := \cup_{x \in cl} Cl(x)$.

For the failure diagnosis purposes, the state set of G is partitioned into faulty and non-faulty states, which are identified by a *fault-assignment* function $\Psi : X \to \{0, 1\}$ with the implication that $\Psi(x) = 1$ if and only if state x is a faulty state. For a path π in G, N_π^F denotes the number of states in π that are faulty, in which case π is said to contain N_π^F faults. We assume without loss of any generality that the system G to be diagnosed is non-terminating, i.e., G does not contain states where no transition is defined. Otherwise, we can add self-loops on ϵ on every terminating state of G without altering its diagnosability properties [5].

The events executed by a discrete-event system to be diagnosed are observed using sensors, which can be represented as an observation map, $M : \Sigma \cup \{\epsilon\} \to \Delta \cup \{\epsilon\}$ satisfying $M(\epsilon) = \epsilon$, and where Δ is the set of observed symbols. The observation mask can be extended from events to traces: $M(\epsilon) = \epsilon$ and $\forall s \in \Sigma^*, \sigma \in \Sigma, M(s\sigma) = M(s)M(\sigma)$. A pair of paths $\pi_i = (x_{i0}, \cdots, x_{ik_i}), (i = 1, 2)$ in G are said to be *indistinguishable* if they can generate a common event-trace observation, i.e., $O_{\pi_1} \cap O_{\pi_2} \neq \emptyset$, where $O_{\pi_i} = \{M(s) \in \Delta^* \mid s = (\sigma_{i1} \cdots \sigma_{ik_i}), x_{ij} \in \alpha(x_{i(j-1)}, \sigma_{ij}), 1 \leq j \leq k_i\}$ for $i = 1, 2$.

The following definition of ℓ-diagnosability was introduced in [5] to allow the diagnosis of the ℓth occurrence of a fault within a bounded delay.

Definition 1: Given a system G along with a fault assignment function Ψ, and an observation mask M, G is *ℓ-diagnosable with respect to M and Ψ* if,

$$\begin{aligned}&\exists n_\ell \in N,\\ &[\forall \pi \text{ generated by } G \text{ with } N_\pi^F \geq \ell,\\ &\forall \overline{\pi} = \pi\pi' \text{ in } G \text{ with } |\pi'| \geq n_\ell,\\ &\forall \hat{\pi} \text{ generated by } G \text{ with } O_{\overline{\pi}} \cap O_{\hat{\pi}} \neq \emptyset\\ &\quad \Rightarrow \quad N_{\hat{\pi}}^F \geq \ell].\end{aligned}$$

Further if a system is ℓ-diagnosable for each $\ell \geq 1$, then it is called $\forall\ell$-diagnosable. Note for a $\forall\ell$-diagnosable system the diagnosis delay bound can be a function of ℓ, and although the diagnosis delay bound will be finite for each ℓ, the various delay bounds may not be uniformly bounded.

III. Indistinguishable-Pairs Automaton

Our algorithms are based on a testing automaton that tracks all indistinguishable pairs of traces, and we refer to it as an *indistinguishable-pairs automaton*.

Definition 2: Given a system $G = (X, \Sigma, \alpha, x_0)$ with a fault assignment function Ψ, and an observation mask M, an *indistinguishable-pairs automaton* I is defined as follows: $I = (Y, \Sigma^T, \beta, y_0)$, where $Y := X \times X$ is the set of states, $\Sigma^T := \overline{\Sigma} \times \overline{\Sigma}$ is the set of events, $y_0 = (x_0, x_0)$ is the initial state, and $\beta : Y \times \Sigma^T \to 2^Y$ is the transition function that is defined as follows:

$$\forall y = (x_+, x_-) \in Y,\ \sigma^T = (\sigma_+, \sigma_-) \in \Sigma^T - \{(\epsilon, \epsilon)\},$$
$$\beta(y, \sigma^T) := \begin{cases} \alpha(x_+, \sigma_+) \times \alpha(x_-, \sigma_-) & \text{if } M(\sigma_+) = M(\sigma_-) \\ \emptyset & \text{otherwise} \end{cases}$$

We write a state $y \in Y$ as $y = (x_+(y), x_-(y))$. It follows from the definition of I that each path $\pi \in Y^*$ in I is a pair (π_+, π_-) such that $\pi_+, \pi_- \in X^*$ are paths in G that are indistinguishable to each other.

Associated with a state $y = (x_+(y), x_-(y)) \in Y$ is the fault-label pair, $(\Psi(x_+(y)), \Psi(x_-(y)))$. Using this we define the following notions of "weights" of a state/path in I.

Definition 3: Consider a state $y \in Y$ and a path $\pi = (y_1, \cdots, y_n)$ in I.

i) The $+$ve and $-$ve weights of y are defined as, $w_+(y) := \Psi(x_+(y))$ and $w_-(y) := \Psi(x_-(y))$, the vector-weight of y is defined as, $\overrightarrow{w}(y) := (w_+(y), w_-(y))$, and the net weight of y is defined as, $w(y) := w_+(y) - w_-(y)$.
ii) The $+$ve and $-$ve weights of π are defined as, $w_+(\pi) = \sum_{y \in \pi} w_+(y)$ and $w_-(\pi) = \sum_{y \in \pi} w_-(y)$, the vector-weight of π is defined as, $\overrightarrow{w}(\pi) := (w_+(\pi), w_-(\pi))$, and the net weight of π is defined as, $w(\pi) := w_+(\pi) - w_-(\pi)$.

We use the expression $w_\pm(\cdot) = c$ to imply either $w_+(\cdot) = c$ or $w_-(\cdot) = c$.

Definition 4: Given a path π (can be a cycle) in I,

i) π is a *zero-path* if $w(\pi) = 0$; otherwise, π is called a *non-zero path*. A non-zero path is a *+ve-path* (resp., *−ve-path*) if $w(\pi) > 0$ (resp., $w(\pi) < 0$).
ii) π is *fault-free* if $w_+(\pi) = 0$ and $w_-(\pi) = 0$. π is called *1-part fault-free* if either $w_+(\pi) = 0$ or $w_-(\pi) = 0$.
iii) π is *+ve-vocal* if exists a transition on (σ_+, σ_-) in π with $\sigma_+ \neq \epsilon$. π is *−ve-vocal* if exists a transition on (σ_+, σ_-) in π with $\sigma_- \neq \epsilon$.

Note that vocality signifies execution of at least one event.

IV. TEST FOR $\forall\ell$-DIAGNOSABILITY

In this section, we present a test to verify whether a repeatable-fault is diagnosable for every occurrence index, i.e., whether it is "$\forall\ell$-diagnosable". By Definition 1, a system is not ℓ-diagnosable if and only if there exists a pair of indistinguishable infinitely-long paths (π_+, π_-) generated by G such that $\min\{N^F_{\pi_+}, N^F_{\pi_-}\} < \ell$ and $\max\{N^F_{\pi_+}, N^F_{\pi_-}\} \geq \ell$, and the path with larger number of faults continues to execute events. Since $\pi = (\pi_+, \pi_-)$ is a path in I and since $N^F_{\pi_+} = w_+(\pi)$ and $N^F_{\pi_-} = w_-(\pi)$, it follows that G is not ℓ-diagnosable if and only if I contains an infinitely-long non-zero path π, and if the path is $+ve$-path (resp., $-ve$-path), then it is "$+ve$-vocal" (resp., "$-ve$-vocal") infinitely-often. (Recall that "vocal" implies execution of events.) Existence of such a path in I and finiteness of I implies one of the two cases mentioned in the following theorem.

Theorem 1: Consider system G with fault assignment function Ψ, and an observation mask M. Then G is not $\forall\ell$-diagnosable if and only if either one of the following holds in I:

(a) Exists a non-zero cycle cl such that cl is 1-part fault-free,

(b) Exists a fault-free cycle cl and a non-zero path π to cl such that cl is $sgn(w(\pi))$-vocal.

Based on the result of Theorem 1, we next present an algorithm for testing $\forall\ell$-diagnosability.

Algorithm 1: (For testing $\forall\ell$-diagnosability of system G under observation mask M and fault assignment function Ψ.)

(i) (Following condition (a) of Theorem 1)
   a) Delete those states y from I for which $w_+(y) = 1$. Denote the resulting state machine I_1.
   b) Detect strongly connected components (SCCs) in I_1 to identify +ve-part fault-free cycles of I.
   c) Check the existence of a SCC that contains a state y for which $w(y) = -1$.

   If the answer is yes, then G is not $\forall\ell$-diagnosable, and stop; else go to step (ii).

(ii) (Following condition (b) of Theorem 1)
   a) Delete those states y from I for which either $w_+(y) = 1$ or $w_-(y) = 1$. Denote the resulting state machine I_2.
   b) Detect SCCs in I_2 to identify fault-free cycles of I.
   c) Compute in I shortest path to fault-free cycles and check its negativity.

   If the answer is yes, then G is not $\forall\ell$-diagnosable; otherwise, G is $\forall\ell$-diagnosable.

Remark 1: "Delete states" in I takes $O(|X|^2|\Sigma|^2)$ time. "Detect SCCs" takes $O(|X|^2 + |X|^2|\Sigma|^2)$ time. Step (i)(c) can be checked in $O(|X|^2)$. "Compute shortest paths" can be performed in $O(|X||X|^2|\Sigma|^2)$ time by the algorithm given in [2]. Thus the overall complexity of Algorithm 1 is $O(|X|^3|\Sigma|^2)$.

The following example illustrates Algorithm 1.

Example 1: Consider a system G shown in Figure 1. Suppose the fault assignment function is given by, $\Psi(1') = \Psi(2) = \Psi(3) = \Psi(4') = \Psi(6') = 1$, and $\Psi(\cdot) = 0$ for other states, and suppose the observation mask is given by, $M(a_1) = M(a_2) \neq \epsilon$, $M(a) = \epsilon$, and identity mask for other events.

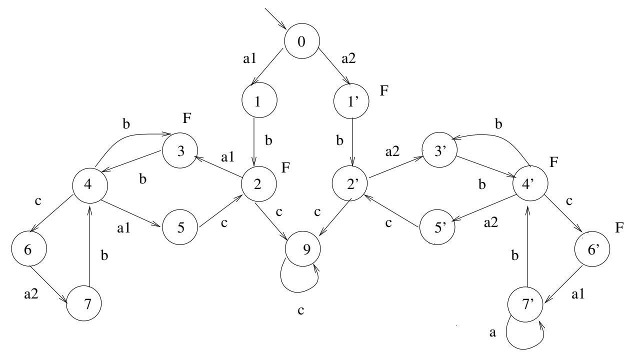

Fig. 1. The system G

The corresponding indistinguishable-pairs automaton I is drawn in Figure 2. Each state in Figure 2 is labeled by its $w(\cdot)$ value. We apply Algorithm 1 to check $\forall\ell$-diagnosability of G.

By step (i)(a) and (b), we obtain SCCs $(44, 66, 77)$, $(44', 66', 77')$, (99), $(7'7)$ and $(7'7')$. By step (i)(c), there exists a SCC $(44', 66', 77')$ containing states $44'$ and $66'$ with $w(44') = w(66') = -1$. Thus, G is not $\forall\ell$-diagnosable. Indeed, one can verify that there exists a pair of indistinguishable infinitely-long paths $\pi = (\pi_+, \pi_-) = (00, 11', 22', 33', 44', 66', (77', 44', 66')^*)$ generated by G. This can be used to argue that G is not 3-diagnosable, implying G is not $\forall\ell$-diagnosable.

V. INDICES VIOLATING ℓ-DIAGNOSABILITY

For systems that fail the $\forall\ell$-diagnosability test, it is desirable to know the set of fault-occurrence indices ℓ for which the system is not ℓ-diagnosable, and we present a method for doing so in this section. We examine the conditions of $\forall\ell$-diagnosability we have obtained, and identify all possible ways in which paths of the underlying testing automaton can violate those conditions. The weights associated with such paths are then examined to determine the indices ℓ for which the system is not ℓ-diagnosable.

A. Identifying Indices Violating ℓ-diagnosability

Suppose condition (a) of Theorem 1 holds, i,e., exists a non-zero cycle cl in I such that it is 1-part fault-free. Then by executing this cycle multiple times a pair of indistinguishable traces can be obtained for which the fault difference count is arbitrarily large. Then given any path π in I ending at cycle cl, the system is not ℓ-diagnosable for all $\ell \in [w_+(\pi)+1, \infty)$ if $w_+(cl) = 0$, and for all $\ell \in [w_-(\pi)+1, \infty)$ if $w_-(cl) = 0$. On the other hand if condition (b) of Theorem 1 holds, i.e., exists a fault-free cycle cl and a non-zero path π to the cycle cl, then a pair of indistinguishable traces can be obtained in which the fault difference count persists at $w(\pi) = w_+(\pi) - w_-(\pi)$. As a result, the system is not ℓ-diagnosable for all

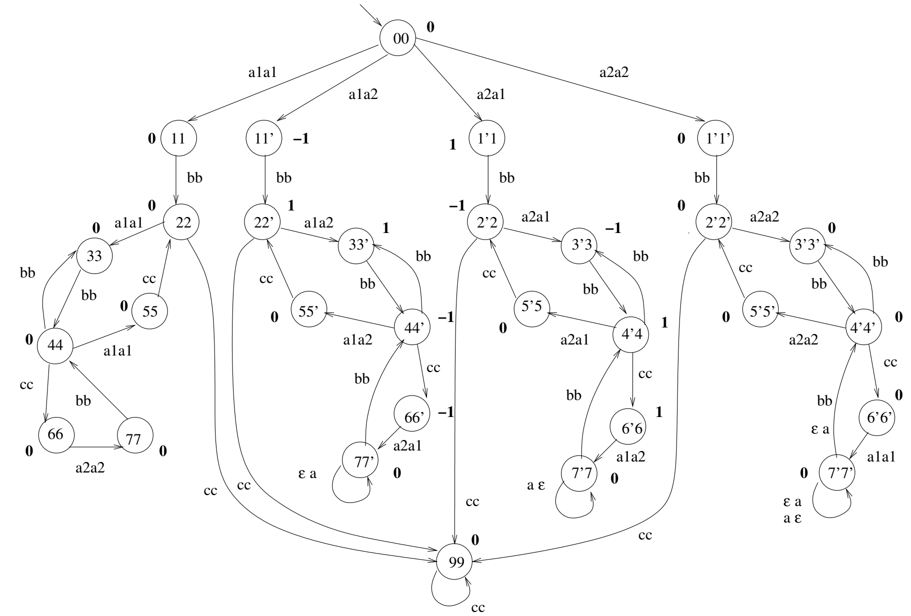

Fig. 2. Indistinguishable-pairs automaton I of G

$\ell \in [\min\{w_+(\pi), w_-(\pi)\}+1, \max\{w_+(\pi), w_-(\pi)\}]$. This is stated in the following theorem.

Theorem 2: Consider G with fault assignment function Ψ, and an observation mask M. G is not ℓ-diagnosable if and only if,

(a) Exists cl satisfying Theorem 1 (a) and

$$\ell \in \begin{cases} \cup_{\{\text{path } \pi \text{ to } cl\}}[w_+(\pi)+1, \infty) & \text{if } w_+(cl)=0 \\ \cup_{\{\text{path } \pi \text{ to } cl\}}[w_-(\pi)+1, \infty) & \text{if } w_-(cl)=0 \end{cases}$$

or

(b) Exists cl and π satisfying Theorem 1 (b) and

$$\ell \in [\min\{w_+(\pi), w_-(\pi)\}+1, \max\{w_+(\pi), w_-(\pi)\}].$$

Both the conditions of Theorem 2 require computation of the weights of certain paths of I. Since there may exist infinitely many such paths, it is not possible to first obtain all such paths and then compute their weights. However, since any path is a concatenation of elementary paths and elementary cycles [6], which can be uniquely determined (algorithms for finding elementary paths and elementary cycles can be found in [7], [10]), we can break down each path into its constituent elementary paths and elementary cycles, so that the weight of a path can be computed as the superposition of the weights of its constituent elementary paths and elementary cycles (the order of appearance of elementary paths or cycles does not matter). Conversely, starting from an elementary path π of a system, a feasible (non-elementary) path can be obtained by inserting appropriate closed paths (the begin state is same as the end state) into π, where a closed path can be obtained by visiting a set of elementary cycles certain times in certain orders. Thus, it is possible to compute the weights of all paths, from feasible superpositions of weights of elementary paths and cycles. Since in a closed path elementary cycles may have to appear in a certain order, we next introduce the notion of state-relative ordering of elementary cycles and present a way to compute it.

B. State-Relative Ordering of Cycles

The visiting of a set of elementary cycles in a closed path can not be done in an arbitrary order, since in order to reach an elementary cycle some other elementary cycles may have to be reached first. To see this, consider state $44'$ in Figure 2, elementary cycles $(22', 33', 44', 55')$, $(33', 44')$, $(44', 66', 77')$ and $(77')$ form the strongly connected component containing $44'$. We are able to obtain a closed path from $44'$ that contains cycle $(77')$, but to reach this cycle, the cycle $(44', 66', 77'')$ must be reached first. In contrast, cycles $(33', 44')$ and $(44', 66', 77'')$ can be reached independently of each other.

Such dependency among cycles gives rise to a constraint on the number of their execution times in a closed path containing those cycles. For example, in any closed path starting from state $44'$, the execution time of cycle $(77')$ can be non-zero only if the execution time of cycle $(44', 66', 77'')$ is non-zero. To reflect such a dependency relation, we introduce a notion of *state-relative ordering relation over cycles*.

Definition 5: For a state y in I, the y-relative ordering relation over cycles is denoted $<_y$ and is inductively defined as follows.

1) Given two cycles cl_0 and cl_1, $cl_0 <_y cl_1$ if

$$cl_0 \in Cl(y) \text{ and } cl_1 \in Cl(cl_0) - Cl(y).$$

2) Given a sequence of cycles $\{cl_i, 0 \le i \le n, n \ge 2\}$, $cl_0 <_y \cdots <_y cl_n$ if

$$\begin{aligned} & cl_0 <_y \cdots <_y cl_{n-1}, \text{ and} \\ & \exists y' \in cl_{n-1} \text{ s.t. } cl_{n-1} <_{y'} cl_n, \text{ and} \\ & cl_n \in Cl(cl_{n-1}) - Cl(\cup_{i=0}^{n-2} cl_i). \end{aligned}$$

The y-relative ordering of cycles can be used to arrange cycles in sequences which obey the ordering as captured in the following definition..

Definition 6: For $y \in Y$, the set of sequences of cycles obeying the y-relative ordering of cycles is given by,

$$ClSeq(y) := \{\langle cl_0, \cdots, cl_n\rangle \mid n \geq 0, cl_0 \in Cl(y), \text{ and } cl_0 <_y \cdots <_y cl_n\}.$$

$ClSeq(y)$ can be computed using a depth-first search algorithm, which is omitted here due to the space limitation.

The collection of all cycles reachable from y through cycles is given by,

$$Cl^*(y) := \{cl \in Cl(Y) \mid \exists clseq \in ClSeq(y) \text{ containing } cl\}.$$

It is easy to see that cycles in $Cl^*(y)$ are reachable from each other, and $Cl^*(y)$ is a maximal collection of cycles that have such property. I.e., $Cl^*(y)$ is a collection of cycles that forms a SCC. This fact is stated in the following lemma.

Lemma 1: $Cl^*(y)$ is a collection of cycles that forms a strongly connected component containing state y, where $Cl^*(y)$ is defined in Definition 6.

C. Computing Weight of a Path

As discussed above, an arbitrary feasible path can be obtained by inserting appropriate closed paths into an elementary path, where the closed paths are composed of elementary cycles. Thus, any path can be viewed as being "generated" from an elementary path (via insertions of closed paths composed of a set of elementary cycles). The set of paths *generated by an elementary path* differ in their constituent elementary cycles and corresponding execution times.

The following example illustrates the concept of paths generated by an elementary path.

Example 2: Consider the indistinguishable-pairs automaton I shown in Figure 2, and consider the path $\pi_1 = (00, 11', 22', 33', 44', 55', 22', 33', 44', 66', 77')$. Then $\pi_1 = (00, 11') \cdot (22', 33', 44', 55') \cdot (22', 33', 44', 66', 77')$, i.e., path π_1 can be decomposed into elementary paths $(00, 11')$ and $(22', 33', 44', 66', 77')$, and an elementary cycle $cl_1 = (22', 33', 44', 55')$. Further, elementary paths $(00, 11')$ and $(22', 33', 44', 66', 77')$ can be concatenated to yield a single elementary path $(00, 11', 22', 33', 44', 66', 77')$.

Similarly, consider another path $\pi_2 = (00, 11', 22', 33', 44', 33', 44', 66', 77', 44', 66', 77')$. Then π_2 can be decomposed into a single elementary path $(00, 11', 22', 33', 44', 66', 77')$ and elementary cycles $cl_2 = (33', 44')$ and $cl_3 = (66', 77', 44')$.

Thus, both π_1 and π_2 are generated by the elementary path $(00, 11', 22', 33', 44', 66', 77')$.

We say two paths π_1 and π_2 are related, denoted $\pi_1 R \pi_2$ if they can be generated by a common elementary path $tr \in Tr(Y)$. The following lemma states that this relation is an equivalence relation and implies that each path is generated by a unique elementary path.

Lemma 2: The set of all paths in I can be partitioned in such a way that each partition consists of paths generated by a unique elementary path.

Using the facts that (i) any path can be generated by a unique elementary path $tr \in Tr(Y)$ by inserting certain closed paths, and (ii) each closed path is obtained by executing certain elementary cycles certain times in certain order, and (iii) all the elementary cycles that can be possibly included in closed paths starting from y belong to the SCC $Cl^*(y)$, and (iv) elementary cycles in $Cl^*(y)$ may only be chosen according to the y-relative ordering, the set of weights for the set of paths generated by tr, denoted $\vec{W}(tr)$, can be obtained as:

$$\vec{W}(tr) := \Big\{ \sum_{y_i \in tr} \sum_{cl_j^{(i)} \in Cl^*(y_i)} \vec{w}(tr) + m_j^{(i)} \vec{w}(cl_j^{(i)}) \mid m_j^{(i)} \geq 0, \text{ and } [m_j^{(i)} > 0] \Rightarrow [\exists \langle cl_{j_0}^{(i)}, \cdots, cl_{j_n}^{(i)} \rangle \in ClSeq(y_i) : (cl_{j_n}^{(i)} = cl_j^{(i)}) \wedge (m_{j_k}^{(i)} > 0, \forall 0 \leq k < n)] \Big\}. \quad (1)$$

Also for $\widehat{Tr} \subseteq Tr(Y)$, $\vec{W}(\widehat{Tr}) := \cup_{tr \in \widehat{Tr}} \vec{W}(tr)$.

Here y_i represents i^{th} state in the elementary path tr, $cl_j^{(i)}$ represents the j^{th} elementary cycle appearing along a closed path inserted at state y_i, and $m_j^{(i)}$ is the number of times such a cycle is executed along such a closed path. Clearly, $m_j^{(i)} \geq 0$ for all i and j. Also if $m_j^{(i)}$ is postive (i.e., if $cl_j^{(i)}$ is executed at least once), then all cycles along a cycle-sequence ending at $cl_j^{(i)}$ must also be executed at least once.

D. Computing Indices Violating ℓ-Diagnosability

Application of Theorem 2 for computation of indices ℓ violating ℓ-diagnosability requires computation of the weights of all paths that satisfy either condition (a) or (b). Let $Cl^{\circ}(Y)$ denote the set of fault-free cycles in I, and let $Tr(y_0, cl) \subseteq Tr(y_0)$ denote the set of elementary paths that start from initial state y_0 and end at a cycle cl.

Algorithm 2: (For finding the set of indices for which ℓ-diagnosability of system G is violated under observation mask M and fault assignment function Ψ.)

(i) (Set of indices satisfying condition (a) of Theorem 2)

$$\mathcal{L}_a := \bigcup_{cl \in Cl^{\circ}(Y)} \bigcup_{tr \in Tr(y_0, cl)} [w_+(tr) + 1, \infty) = \left[\min_{cl \in Cl^{\circ}(Y), tr \in Tr(y_0, cl)} (w_+(tr) + 1), \infty\right].$$

Then $\mathcal{L}_a$ is the set of indices satisfying Theorem 2, condition (a).

(ii) (Set of indices satisfying condition (b) of Theorem 2)

$$\mathcal{L}_b := \bigcup_{cl \in Cl^{\circ}(Y)} \{[\min(w_+, w_-) + 1, \max(w_+, w_-)] \mid (w_+, w_-) \in \vec{W}(Tr(y_0, cl)), \text{ and } cl \text{ is } sgn(w_+ - w_-)\text{-vocal}\}.$$

Then $\mathcal{L}_b$ is the set of indices satisfying Theorem 2, condition (b).

The following theorem establishes the correctness of Algorithm 2.

Theorem 3: Algorithm 2 is correct.

The following example illustrates Algorithm 2.

Example 3: Consider system G in Example 1 and its indistinguishable-pairs automaton I shown in Figure 2. We compute the set of all indices ℓ for which G is not ℓ-diagnosable.

By step (i) of Algorithm 1, we can obtain a +ve-part fault-free non-zero cycle $cl = (44', 66', 77')$. The set of elementary paths to cl is $\{(00, 11', 22', 33', 44'), (00, 11', 22', 33', 44', 66'), (00, 11', 22', 33', 44', 66', 77')\}$. Thus, $\mathcal{L}_a = [\min\{2, 2, 2\} + 1, \infty) = [3, \infty)$.

By step (ii) of Algorithm 1, we can obtain fault-free cycles $Cl^{\circ}(Y) = \{(44, 66, 77), (77'), (7'7), (7'7'), (99)\}$. Let $cl = (77')$, $cl_1 = (22', 33', 44', 55')$, $cl_2 = (33', 44')$, $cl_3 = (44', 66', 77')$.

Then $Tr(y_0, cl) = \{(00, 11', 22', 33', 44', 66', 77')\} = \{tr\}$. By definitions of $ClSeq(\cdot)$ and $Cl^*(\cdot)$, we have:

ClSeq(00)=ClSeq$(11') = \emptyset$,

$Cl^*(00) = Cl^*(11') = \emptyset$,

ClSeq$(22') = \{\langle cl_1, cl_2\rangle, \langle cl_1, cl_3, cl\rangle, \langle cl_1\rangle, \langle cl_1, cl_3\rangle\}$,

$Cl^*(22') = \{cl_1, cl_2, cl_3, cl\}$,

ClSeq$(33') = \{\langle cl_1, cl_3, cl\rangle, \langle cl_2, cl_3, cl\rangle, \langle cl_1\rangle, \langle cl_1, cl_3\rangle, \langle cl_2\rangle, \langle cl_2, cl_3\rangle\}$,

$Cl^*(33') = \{cl_1, cl_3, cl, cl_3\}$,

ClSeq$(44') = \{\langle cl_2\rangle, \langle cl_1\rangle, \langle cl_3, cl\rangle, \langle cl_3\rangle\}$,

$Cl^*(44') = \{cl_2, cl_1, cl_3, cl\}$,

ClSeq$(66') = \{\langle cl_3, cl_2\rangle, \langle cl_3, cl_1\rangle, \langle cl_3\rangle, \langle cl_3, cl\rangle\}$,

$Cl^*(66') = \{cl_3, cl_2, cl_1, cl\}$,

ClSeq$(77') = \{\langle cl_3, cl_2\rangle, \langle cl_3, cl_1\rangle, \langle cl\rangle, \langle cl_3\rangle\}$,

$Cl^*(77') = \{cl_3, cl_2, cl_1, cl\}$.

By Equation (1), we have

$$\begin{aligned}
&\vec{W}(tr) = \{(2,3) + \\
&[m_{cl_1}^{(22')} + m_{cl_1}^{(33')} + m_{cl_1}^{(44')} + m_{cl_1}^{(66')} + m_{cl_1}^{(77')}] \times (2,1) \\
&+[m_{cl_2}^{(22')} + m_{cl_2}^{(33')} + m_{cl_2}^{(66')} + m_{cl_2}^{(44')} + m_{cl_2}^{(77')}] \times (1,1) \\
&+[m_{cl_3}^{(22')} + m_{cl_3}^{(33')} + m_{cl_3}^{(44')} + m_{cl_3}^{(66')} + m_{cl_3}^{(77')}] \times (0,2) \\
&\mid m_{cl_2}^{(22')} > 0 \Rightarrow m_{cl_1}^{(22')} > 0,\ m_{cl_3}^{(22')} > 0 \Rightarrow m_{cl_1}^{(22')} > 0, \\
&m_{cl_3}^{(33')} > 0 \Rightarrow m_{cl_1}^{(33')} > 0, \quad m_{cl_3}^{(33')} > 0 \Rightarrow m_{cl_2}^{(33')} > 0, \\
&m_{cl_2}^{(66')} > 0 \Rightarrow m_{cl_3}^{(66')} > 0, \quad m_{cl_1}^{(66')} > 0 \Rightarrow m_{cl_3}^{(66')} > 0, \\
&m_{cl_2}^{(77')} > 0 \Rightarrow m_{cl_3}^{(77')} > 0, \quad m_{cl_1}^{(77')} > 0 \Rightarrow m_{cl_3}^{(77')} > 0\}
\end{aligned}$$

When the values of all variables $m_j^{(i)}$ are zero, $(w_+, w_-) = (2, 3)$ (equaling the weight of the path tr). Since cycle cl is $-$ve-vocal, $[\min(w_+, w_-) + 1, \max(w_+, w_-)] = [w_+ + 1, w_-] = [3, 3] = \{3\} \in \mathcal{L}_b$.

When $m_{cl_2}^{(33')} = 1$ and all other $m_j^{(i)}$'s are zero, $(w_+, w_-) = (2, 3) + (1, 1) = (3, 4)$ (equaling the weight of the path $(00, 11', 22', 33', 44', 33', 44', 66', 77')$). Since cycle cl is $-$ve-vocal, $[\min(w_+, w_-) + 1, \max(w_+, w_-)] = [w_+ + 1, w_-] = [4, 4] = \{4\} \in \mathcal{L}_b$.

Note when $m_{cl_1}^{(22')} = 2$ and all other $m_j^{(i)}$'s are zero, $(w_+, w_-) = (2, 3) + (4, 2) = (6, 5)$ and $[\min(w_+, w_-) + 1, \max(w_+, w_-)] = [w_- + 1, w_+] = [6, 6] = \{6\}$. However, cycle cl is $-$ve-vocal. So this interval does not belong to $\mathcal{L}_b$.

A complete analysis can be used to conclude that G is not $[3, \infty)$-diagnosable.

VI. Conclusion

The goal of the paper is the identification of fault-occurrence indices ℓ of a repeatable-fault for which the given system is not ℓ-diagnosable. For this, we provided a condition to check whether a system is ℓ-diagnosable for every $\ell \geq 1$, i.e., $\forall\ell$-diagnosable. For the class of systems that are not $\forall\ell$-diagnosable, we next presented a condition to determine the set of indices ℓ for which the system is not ℓ-diagnosable by identifying all possible ways in which the paths of the underlying testing automaton can violate the ℓ-diagnosability property. The weights associated with such paths were then examined to determine the indices ℓ for which the system is not ℓ-diagnosable. In order to compute the weight of a path, we viewed it as being composed of an elementary path and a set of elementary cycles. Since elementary cycles may have to appear in a certain order along a path, the notion of state-relative ordering of cycles was introduced and used for imposing constraints over the execution times of cycles appearing in a path. Then the weight of a path was computed by using as "basis" the weights of its constituent elementary path and elementary cycles obeying such constraints.

References

[1] S. R. Das and L. E. Holloway. Characterizing a confidence space for discrete event timings for fault monitoring using discrete sensing and actuation signals. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 30(1):52–66, 2000.

[2] A. V. Goldberg. Scaling algorithms for the shortest paths problem. *SIAM Journal on Computing*, 24:494–504, 1995.

[3] S. Jiang and R. Kumar. Diagnosis of repeated failures for discrete event systems with linear-time temporal logic specifications. In *Proceedings of IEEE Conference on Decision and Control*, pages 3221–3226, Maui, Hawaii, 2003.

[4] S. Jiang and R. Kumar. Failure diagnosis of discrete event systems with linear-time temporal logic fault specifications. *IEEE Transactions on Automatic Control*, 49(6):934–945, 2004.

[5] S. Jiang, R. Kumar, and H. E. Garcia. Diagnosis of repeated/intermittent failures in discrete event systems. *IEEE Transactions on Automatic Control*, 19(2):310–323, 2003.

[6] P. F. Stadler P. M. Gleiss, J. Leydold. Circuit bases of strongly connected digraphs. *Discussiones Mathematicae Graph Theory*, 23(2):241–260, 2003.

[7] R.C. Read and R.E. Tarjan. Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. *Networks*, 5:357–372, 1975.

[8] M. Sampath and S. Lafortune. Active diagnosis of discrete event systems. *IEEE Transactions on Automatic Control*, 43(7):908–929, 1998.

[9] M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, September 1995.

[10] Herbert Weinblatt. A new search algorithm for finding the simple cycles of a finite directed graph. *Journal of the ACM*, 19(1):43–56, 1972.

[11] T. Yoo and H. E. Garcia. Event diagnosis of discrete-event systems with uniformly and nonuniformly bounded diagnosis delays. In *Proceedings of 2004 American Control Conference*, pages 5102–5107, Boston, MA, June 2004.