Robust diagnosis of discrete-event systems subject to permanent sensor failures

Saulo T. S. Lima∗ Joao C. Basilio∗ Stéphane Lafortune**
Marcos V. Moreira∗

∗ COPPE - Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, 21949-900, Rio de Janeiro, RJ, Brasil. (e-mails: saulotelles@poli.ufrj.br, basilio@dee.ufrj.br, moreira@dee.ufrj.br).
** Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA. (e-mail: stephane@eecs.umich.edu)

Abstract: One approach to online fault diagnosis of discrete-event systems is through the use of the diagnosers. Diagnosers are deterministic automata whose states are sets formed with the states of the plant together with labels that indicate if the trace that has occurred so far possesses or not the fault event. The decision regarding fault occurrence is taken based solely on observable events, i.e., events whose occurrences can be recorded by sensors. However, if one or more sensors that provide information on event occurrences fail, the diagnoser may either come to a halt or may even provide wrong information regarding fault occurrence. In order to overcome this deficiency, this paper proposes a robust diagnoser that deploys the redundancy that may exist in a set formed of diagnosis bases (set of events that guarantee fault diagnosability) with a view to ensure the fault diagnosis even in the occurrence of permanent sensor failures.

Keywords: Discrete-event systems, fault diagnosis, sensor failures, robust diagnosability.

1. INTRODUCTION

The basic event diagnosis problem for discrete event systems is to perform model-based inferencing at run-time, using sequences of observable events, and determine, with certainty, if a given unobservable "fault" event has occurred or not in the past. For discrete event systems modeled by automata, diagnoser automata, or simply diagnosers, can be used to this purpose. The property of diagnosability captures the ability to always detect at run-time any occurrence of the given fault event, within a finite number of event transitions. Diagnosers can be used for testing, offline, diagnosability; verifier automata can also be used to this purpose.

Let us assume a given set of sensors is recording all potentially observable events at run-time. We are interested in the situation where sensors for some combinations of (potentially observable) events fail prior to the first occurrence of an event they are monitoring; such failures are assumed to be permanent. In this case, diagnosers could get stuck in some states (e.g., no further observed event, or occurrence of an event not in the active event set) or could even issue incorrect diagnostic decisions; an example is presented in Section 3. We would like to still perform correct diagnosis of the original unobservable fault event despite the presence of sensor failures; note that the original fault event is different from the events whose sensors fail, since the latter are originally observable. We do not require that these sensor failures be themselves diagnosed, although that could be a by-product of the procedure employed.

There is a large body of literature on event diagnosis for discrete event systems modeled by automata; see, e.g., Sampath et al. (1995); Debouk et al. (2000); Boel and van Schuppen (2002); Tripakis (2002); Zad et al. (2003); Thorsley and Teneketzis (2005); Contant et al. (2006); Qiu and Kumar (2006); Wang et al. (2007); Kumar and Takai (2009); Athanasopoulos et al. (2010). Recently, there have been some works on sensor failures in supervisory control of discrete event systems (Rohloff, 2005; Sanchez and Montoya, 2006), and on “robust” diagnosis, when entire diagnosers may fail and cease to operate (Basilio and Lafortune, 2009). Our approach in this paper is different from the latter, which deals with the problem of withstanding permanent failure of one or more sites. Here we deal with sensor failures, equivalently, with the loss of potentially observable events, and, to do so, we propose to essentially run a set of diagnosers in parallel, where each diagnoser has been designed to work correctly under a certain combination of sensor failures. There could be several possible combinations of sensor failures, and it is not known a priori which of these combinations of failures, if any, will occur at run-time. While the different diagnosers are concurrently observing the behavior of the system, some of them may get stuck due to the loss of observable events. We want to ensure that at least one of these diagnosers will issue the correct diagnostic decision about the unobservable fault event under consideration, no
matter which combination of sensor failures has occurred; we also need to know which diagnoser is giving the correct answer.

Our approach exploits the notion of partial diagnosers (Basilio and Lafortune, 2009) and diagnosis bases (Lima, 2010). Let us assume that a given unobservable fault event, \( \sigma_f \), is diagnosable in a given system for the set of all observable events \( E_0 \), in the sense of Sampath et al. (1995). Let \( E_0' \subseteq E_0 \) be a proper subset of \( E_0 \), for which diagnosability still holds. Then \( E_0' \) is called a diagnosis basis and the events in the set \( E_0 \setminus E_0' \) are said to be redundant; we call \( E_0^\text{red} := E_0 \setminus E_0' \) the set of redundant events associated with \( E_0' \); the partial diagnoser built for \( E_0' \) does not record these (potentially observable) events. We propose to employ several partial diagnosers, where each one is built for a particular diagnosis basis, or equivalently, for a particular set of redundant events. In this context, we present a formal definition for the property of “robust diagnosability” given a set of possible combinations of sensor failures (Section 3). We then develop a test for this new property (Section 4). For the purpose of the testing procedure, we first describe a labeling scheme for partial diagnosers that attaches to a state name a label regarding sensor failure information upon entry into that state. We then combine all partial diagnosers into a union diagnoser that accepts the union of the languages of all partial diagnosers. We study the union diagnoser and show how robust diagnosability against sensor failures can be tested using a cycle-based condition on any union diagnoser (Section 5); this test is related to the familiar “indeterminate cycle” test for the property of diagnosability (Sampath et al., 1995). Finally, when robust diagnosability fails, we show how to use the union diagnoser to determine a smaller set of combinations of sensor failures for which the desired property holds.

2. PRELIMINARIES

Let

\[
G = (X, E, f, \Gamma, x_0),
\]

be a deterministic automaton, where \( X \) denotes the state space, \( E \) the event set, \( f : X \times E \to X \) the state transition function, which is partially defined in its domain, \( \Gamma \) the active event function, and \( x_0 \) the initial state. Let us partition \( E \) as \( E = E_o \cup E_uo \), i.e., \( E_o = E_{uo} \setminus E_uo \), \( E_o \cap E_uo = \emptyset \) and \( E_uo = \emptyset \), where \( E_o \) and \( E_uo \) are, respectively, the set of observable and unobservable events, and let \( E_f = \{ \sigma_f \} \subseteq E_uo \) be a set whose unique element \( \sigma_f \) is the fault event to be detected. Finally, let us denote the language generated by \( G \) as \( L \). The following assumptions are made:

A1. The language \( L \) is live, i.e., \( \Gamma(x_i) \neq \emptyset \) for all \( x_i \in X \).

A2. There is no cycle of unobservable events in \( G \), i.e., \( \forall s \in L, s \in E_uo \implies \exists n \in \mathbb{N} \) such that \( ||s|| \leq n_0 \), where \( ||s|| \) denotes the length of trace \( s \).

The language \( L \) is said to be diagnosable if the occurrence of \( \sigma_f \) can be detected within a finite number of transitions after the occurrence of \( \sigma_f \) using only traces formed with events in \( E_o \). Formally, language diagnosability is defined as follows (Sampath et al., 1995).

Definition 1. The language \( L \) is diagnosable with respect to the natural projection \( P_o : E^* \to E_o^* \) (Ramadge and

\[
\text{Fig. 1. Automaton } G.
\]

\[
\text{Fig. 2. Centralized diagnoser } G_d.
\]

Wonham, 1989) and \( E_f = \{ \sigma_f \} \) if, and only if, the following condition holds true:

\[
(\exists n \in \mathbb{N})(\forall s \in \Psi(E_f))(\forall t \in L/s)(||t|| \geq n \implies D),
\]

where the diagnose condition \( D \) is given as

\[
(\forall \omega \in (P_o^{-1}(P_o(st)) \cap L))(E_f \in \omega),
\]

with \( E^* \) denoting the Kleene closure of \( E \), \( L/s = \{ t \in E^* : st \in L \} \), \( \Psi(E_f) \) the set of all traces of \( L \) that end with event \( \sigma_f \), and \( P_o^{-1} \) the inverse projection of \( P_o \).

One way to verify language diagnosability is by using diagnosers. A diagnoser is a deterministic automaton whose event set is the set of observable events of \( G \) and whose states are formed by adding labels \( Y \) or \( N \) to the states of \( G \) to indicate whether the fault event \( \sigma_f \) has occurred or not. Figure 1 shows the state transition diagram of an automaton \( G \), for which \( E = \{ a, b, c, d, e, \sigma_f \} \), \( E_o = \{ a, b, c, d, e \} \), and \( E_f = \{ \sigma_f \} \). The corresponding diagnoser is depicted in Figure 2.

Let \( G_d = (X_d, E_o, f_d, \Gamma_d, x_{d0}) \) denote the diagnoser associated with \( G \). Then, the states of \( G_d \) can be classified, according to the presence of labels \( Y \) and \( N \), as follows (Sampath et al., 1995).

Definition 2. A state \( x_d \in X_d \) is called certain (or faulty) if \( \ell = Y \) for all \( x \ell \in x_d \), and normal (or non-faulty) if \( \ell = N \) for all \( x \ell \in x_d \). If there exist \( x \ell, y \ell \in x_d \), \( x \) not necessarily distinct from \( y \) such that \( \ell = Y \) and \( \ell = N \), then \( x_d \) is an uncertain state of \( G_d \).
When the diagnoser reaches a certain (resp. normal) state, it is certain that the fault has occurred (resp. not occurred). However, when the diagnoser is in an uncertain state, it cannot draw any conclusion regarding the fault occurrence. If it remains indefinitely in a cycle formed with uncertain states only, then it will not be possible to diagnose the fault occurrence. This leads to the definition of indeterminate cycle as follows.

**Definition 3.** A set of uncertain states \( \{ x_d_1, x_d_2, \ldots, x_d_p \} \subset X_d \) forms an indeterminate cycle if the following conditions hold true:

C.1) \( x_d_1, x_d_2, \ldots, x_d_p \) forms a cycle in \( G_d \);

C.2) \( \exists(x_i^{k_1}, Y), (x_i^{k_1}, N) \in x_d, x_i^{k_1} \) not necessarily distinct from \( x_i^{k_2}, l = 1, 2, \ldots, p, k_1 = 1, 2, \ldots, m_i, \) and \( r_1 = 1, 2, \ldots, \tilde{m}_i \) in such a way that the sequence of states \( \{ x_i^{k_1} \} \), \( l = 1, 2, \ldots, p, k_1 = 1, 2, \ldots, m_i \) \( \in \{ x_i^{k_1} \} \), \( l = 1, 2, \ldots, p, r_1 = 1, 2, \ldots, \tilde{m}_i \) form cycles in \( G \).

It is worth noting that not all cycles of uncertain states of \( G_d \) form indeterminate cycles.

A necessary and sufficient condition for language diagnosability using diagnosers is stated as follows (Sampath et al., 1995).

**Theorem 1.** The language \( L \) generated by automaton \( G \) will be diagnosable with respect to projection \( P_o \) and \( E_f = \{ \sigma_f \} \) if, and only if, the corresponding diagnoser \( G_d \) does not have any indeterminate cycles.

According to Theorem 1, the language generated by the automaton of Figure 1 is diagnosable with respect to \( P_o \) and \( E_f \) since the corresponding diagnoser \( G_d \), shown in Figure 2, does not have any indeterminate cycles.

Let us now consider a set \( E_o' \subset E \) and suppose we form the diagnoser for \( G_d \) assuming \( E_o' \) as the set of observable events. The resulting diagnoser is called partial diagnoser and is usually denoted as \( G'_d \). Given that the diagnoser for \( E_o \) has already been obtained, the computation of \( G'_d \) can be obtained from \( G_d \) by eliminating all transitions in \( E_o \). \( G'_d \) merging all states of \( G_d \) connected with transitions labeled with events in \( E_o \), and renaming the remaining states with the union of all sets that name the merged states (Basilio and Lafortune, 2009).

3. ROBUST DIAGNOSABILITY AGAINST SENSOR FAILURES

Let us consider, again, the automaton shown in Figure 1 and assume, for a while, that a permanent failure of the sensor that records the occurrence of event \( e \) took place before the first occurrence of \( e \). Suppose that trace \( \sigma_e = e \tau_a e \tau_b e \), \( n \in \mathbb{N} \), has been generated. Since event \( \sigma_e \) is unobservable, the first event recognized by the diagnoser of Figure 2 is \( a \). When the diagnoser receives the information on the occurrence of \( a \), it updates its state to \( \{ 5N \} \), where it stands still since \( e \) is the only event that occurs next in trace \( \sigma_f \), but \( e \) is not in the active event set of \( \{ 5N \} \). The diagnoser is, therefore, unable to process any further information it may receive regarding event occurrences, and so, will not be able to reach a certain state, as it should, since trace \( \sigma_f \) contains the fault event \( \sigma_f \) and has arbitrarily long length. This incorrect behaviour of the diagnoser in the presence of sensor failure suggests that the diagnoser must be modified in order to tolerate possible sensor failures.

Let us make the following assumptions.

A3. \( L \) is diagnosable with respect to \( P_o : E^* \rightarrow E_o^* \) and \( E_f = \{ \sigma_f \} \).

A4. A sensor failure, when it occurs, takes place before the first occurrence of the event associated with the sensor and is permanent, i.e. the sensor never recovers.

Assumption A4 accounts for the case of cyclical systems that reset constantly. In such cases, sensor failure may not be considered everywhere, but only when the system is turned off.

In view of assumption A4, it is clear that we must seek a diagnoser that is robust against permanent sensor failures; robustness here should be understood in the sense that \( L \) remains diagnosable even in the case of permanent sensor failures. The following definitions are introduced.

**Definition 4.** (Diagnosis basis) A set \( E_o' \subset E_o \) is a diagnosis basis if \( L \) is diagnosable with respect to projection \( P_o' : E^* \rightarrow E_o'^* \) and \( E_f = \{ \sigma_f \} \).

**Definition 5.** (Minimum diagnosis basis) A set \( E_o' \subset E_o \) is a minimum diagnosis basis if \( E_o' \) is a diagnosis basis, and, for any nonempty subset \( E_o'' \) of \( E_o' \), \( L \) is not diagnosable with respect to projection \( P_o'' : E^* \rightarrow E_o''^* \) and \( E_f = \{ \sigma_f \} \).

According to Definitions 4 and 5, the main difference between a minimum diagnosis basis and a diagnosis basis is that the events in the former are all essential in the sense that if one of them is withdrawn from the basis, diagnosability is lost, whereas the latter may possess redundant events, i.e., not all events may be necessary to keep diagnosability.

We now present a definition of robust diagnosability against sensor failures.

**Definition 6.** (Robust diagnosability against permanent sensor failures) Let \( E_o \subset E_o'' \subset E_o \), \( i = 1, 2, \ldots, m \), where \( E_o'' \) is either a minimum or nonminimum diagnosis bases for \( L \). Define the set

\[
E_{rob} = \{ E_{o_1}, E_{o_2}, \ldots, E_{o_{mn}} \},
\]

where \( E_{o_i} = E_o \setminus E_{o_i}, i = 1, 2, \ldots, m \). Then \( L \) is robustly diagnosable against permanent sensor failures associated with the events in the sets of \( E_{rob} \) with respect to projections \( P_{o_1}, P_{o_2}, \ldots, P_{o_{mn}} \), where \( P_{o_i} : E^* \rightarrow E_{o_i}^* \) and \( E_f = \{ \sigma_f \} \), if the following condition holds true:

\[
(\exists n \in \mathbb{N})(\exists s \in \Psi(E_f))(\forall t \in L/s) \left( \left( ||t|| \geq n \Rightarrow D_p \right) \right),
\]

where the diagnosability condition \( D_p \) is given as

\[
(\forall i, j \in \{ 1, 2, \ldots, m \}, i \neq j) \left( \exists \omega_j \in L \left[ E_f \notin \omega_j \land P_{o_i}(\omega_i) = P_{o_j}(w_j) \right] \right).
\]

The diagnoser that is able to diagnose a fault and satisfies the conditions imposed by Definition 6 will be referred throughout this paper as a “robust diagnoser against permanent sensor failures” or, simply, “robust diagnoser”.

102
The idea behind Definition 6 is that since $L$ is diagnosable with respect to $P', o_{E^*} : E^* \rightarrow E_{o,i}^*$, and $E_f = \{ \sigma_f \}$, only performs properly if all events in $E^P$ fail. In this case, while some partial diagnosers may get stuck, others may continue running, since it is possible that the intersections of the languages generated by two different partial diagnosers be nonempty. This implies that it is possible that an arbitrarily long trace $s_f$ that contains the fault event has the same projection over, say $E^P$ and $E_{o,i}^*$, where the former takes $G_d$, to a certain state but the latter takes $G_{d,s}$ to a normal state. In this case, according to Definition 6, $L$ is not robustly diagnosable against permanent sensor failures associated with the events in $E^P$. Therefore, robust diagnosis and codiagnosis (Debouk et al., 2000) differ in the following aspects: (i) while the partial diagnosers in a robust diagnosis structure have access to all observable events, the diagnosers that form a codiagnosis structure only have access to part of the observable events; (ii) a coordinator, in a robust diagnosis structure, declares fault when all partial diagnosers that are still running call fault, whereas, in a codiagnosis structure, it declares fault if at least one partial diagnoser calls fault.

In view of Definition 6, and in order to distinguish between different partial diagnosers that are being used to account for sensor failures, a robust diagnoser must have the following properties:

**P1.** The language generated by a robust diagnoser must contain the largest possible number of languages generated by the partial diagnosers whose observable events are the bases (minimum and nonminimum) for fault diagnosis;

**P2.** The language generated by the diagnoser must be the union of all languages generated by the partial diagnosers;

**P3.** The robust diagnoser must keep the labels $Y$ and $N$ of the partial diagnosers;

**P4.** The states of the partial diagnosers must include labels that indicate which sensor failures were responsible for taking the robust diagnoser to that state.

### 4. THE UNION DIAGNOSER

In order to satisfy properties **P3** and **P4**, the first step in the robust diagnoser construction must be the construction of the partial diagnosers whose observable event sets are the bases, and to add sensor fault information to their states. In order to do so, assume that $E_o \subset E_o^*$ is a basis for diagnosis and let $E_o^P = E_o \setminus E_o^* = \{ \sigma_1^*, \sigma_2^*, \ldots, \sigma_{p}^* \}$. Consider the following definitions.

**Definition 7.** Let $\sigma_i$ and $\sigma_f$ denote, respectively, the non-occurrence and occurrence of event $\sigma_i$, assuming failure of the corresponding sensor. The sensor failure label set is defined as:

$$M = \{ S_m : m \in \{ \sigma_1^*, \sigma_2^* \} \times \{ \sigma_1^*, \sigma_2^* \} \times \cdots \times \{ \sigma_1^*, \sigma_2^* \} \}.$$

**Definition 8.** The sensor failure label assignment function $S : X_d \times M \times E_o \rightarrow M$. Let $x_d \in X_d$, $S_m \in M$ and $\sigma \in \Gamma_d(x_d)$. Then

$$S(x_d, S_m, \sigma) = S_m',$$

where

$$m' = \begin{cases} m_n, & \text{if } \sigma \notin E_{o,i}^* \\ \sigma_1, \sigma_2, \ldots, \sigma_{k-1}, \sigma_k, \sigma_{k+1}, \ldots, \sigma_p, & \text{if } \sigma \in E_{o,i}^* \end{cases}, \text{ if } \sigma \in E_{o,i}^* \setminus E_{o,i}^*.$$

The first modification to be carried out in diagnosers with a view to accounting for sensor failure is to introduce labels to indicate that either the sensor responsible for recording the event occurrence has failed or the automaton is going through a path that does not possess the event whose sensor failure under consideration. This leads to the following definition.

**Definition 9.** A. A diagnoser with sensor failure labels is defined as

$$\hat{G}_d(E_o \setminus E_o^*),$$

where $\hat{X}_d \subseteq X_d \times M$, $\hat{x}_d = x_o S_o \bar{S}_1^*, \ldots, \bar{S}_p^*$, and $\hat{f}_d$ and $\hat{\Gamma}_d$ are defined as follows: if $\tilde{x}_d \in S_m \setminus E_f$ and assuming that $f_d(\tilde{x}_d, \sigma) = \tilde{x}_d'$, then $\Gamma_d(\tilde{x}_d) = \Gamma_d(\tilde{x}_d)$ and $f_d(\tilde{x}_d, \sigma) = \tilde{x}_d \in S_m'$, where $S_m' = S(x_d, S_m, \sigma)$.

B. The diagnoser with normal sensor behavior is the diagnoser obtained by adding label $S_0$ (meaning no sensor failure) to all states of $G_d$, and is the case when $E_o^* = E_o$. For this reason this diagnoser is denoted as $G_d(\emptyset)$.

It is clear that $L(\hat{G}_d(E_o \setminus E_o^*)) = L(G_d)$, and thus $\hat{G}_d(E_o \setminus E_o^*)$ does not take into account any possible loss of observability of the events of $E_o \setminus E_o^*$. This is considered by constructing the partial diagnoser that assumes $E_o^*$ as the set of observable events. Such a diagnoser will be referred to as diagnostic with sensor failure labels and will be denoted as $\tilde{G}_d$. Its construction is carried out according to the following algorithm.

**Algorithm 1.** (Computation of partial diagnosers with sensor failure labels) Let $E_o$ be the set of observable events and assume that $E_o^* \subset E_o$ is a basis for the diagnosis of $L$.

**Step 1** Construct the diagnoser with sensor failure labels $G_d(E_o \setminus E_o^*)$.

**Step 2** Compute the observer of $\tilde{G}_d(E_o \setminus E_o^*)$, assuming $E_o^*$ as the set of observable events, and denote it as $\text{obs}(\tilde{G}_d(E_o \setminus E_o^*))$.

**Step 3** Form each state of $\tilde{G}_d(E_o \setminus E_o^*)$ by computing the union of the sets that are the elements of each state of $\text{obs}(\tilde{G}_d(E_o \setminus E_o^*))$.

Let us assume now that all minimum diagnosis bases have been found (Lima, 2010), and let $E_{o,i} \subset E_o$, $i = 1, 2, \ldots, N_b$, denote all minimum diagnosis bases for $L$. Define the set

$$E_{o,i} = E_o \setminus E_{o,i}, \text{ } i = 1, 2, \ldots, N_b,$

and form its power set $2^{E_{o,i}}$. Then, the set $E_{d,\max}$ that contains all minimum and nonminimum diagnosis bases can be formed as follows:

$$E_{d,\max} = \cup_{i=1}^{N_b} E_{d,i},$$

where

$$E_{d,i} = \{ \tilde{E} : (\exists E_{\text{pow}} \in 2^{E_{o,i}})(\tilde{E} = E_{o,i} \cup E_{\text{pow}}) \}.\]
It is clear from Definition 9 and Algorithm 1 that in order to satisfy properties \( P_1 \) and \( P_2 \), it is necessary to obtain the centralized diagnoser with no sensor failure labels and all partial diagnosers with sensor failure labels associated with \( E_{db,max} \). In the sequel, to build a diagnoser whose generated language is the union of the languages generated by the centralized diagnoser with no sensor failure label and all partial diagnosers with sensor failure labels.

**Definition 10.** (Union diagnoser) Let 
\[
E_{db,max} = \{E_{0a}, E_{0b}, \ldots, E_{0a}, E_{0b}\}
\]
denote the set of all bases for the diagnosis of \( L \), and let \( G'_d, i = 1, \ldots, q \), denote the partial diagnosers with sensor failure labels and \( G_{du} \), the centralized diagnoser with no sensor failure. The union diagnoser, denoted as \( G_{du}(E_{db,max}) \), is the diagnoser whose generated language is the union of the languages generated by \( G'_d, i = 0,1, \ldots, q \). \( \square \)

The construction of the union diagnoser can be carried out in a straightforward way as follows (Cassandras and Lafortune, 2008, p. 94): create a new initial state and connect it with \( e \)-transitions to the initial states of \( G'_d, i = 0,1, \ldots, q \). This results in a nondeterministic automaton whose generated language is the union of the languages generated by \( G'_d, i = 0,1, \ldots, q \). The corresponding deterministic automaton is obtained by performing the computation of the observer automaton with respect to \( E_o \).

The construction of the union diagnoser is illustrated with the following example.

**Example 1.** Let us consider automaton \( G = (X, E, f, \Gamma, x_0) \) whose state transition diagram is depicted in Figure 1, where \( E_o = \{a, b, c, d, e\} \) and \( E_f = \{\sigma\} \). As calculated in Lima (2010), the minimum bases for the diagnosis of \( L \) are the elements of the following set:
\[
E_{mdb} = \{\{a, b, c\}, \{c, d, e\}, \{a, c, d\}, \{a, d, e\}, \{a, b, e\}, \{b, c, e\}\}. \tag{6}
\]

The nonminimum bases for the diagnosis of \( L \) can be obtained in a straightforward way, leading to the following set:
\[
E_{nmdb} = \{\{a, b, c, d\}, \{a, b, e\}, \{a, b, d, e\}, \{a, c, d, e\}, \{a, c, d, e\}, \{a, d, e\}, \{a, b, e\}, \{b, c, e\}\}. \tag{7}
\]

Therefore,
\[
E_{db,max} = E_{mdb} \cup E_{nmdb}. \tag{8}
\]

The first step to build the union diagnoser is to obtain the diagnoser with no sensor failure label, which is shown in Figure 3. For notation convenience, the states of \( G_d \) have been renamed as follows: \( x_0 = \{1N\} \), \( x_1 = \{2N, 3'\} \), \( x_2 = \{5N\} \), \( x_3 = \{4'Y\} \), \( x_4 = \{Y\} \) and \( x_5 = \{6N\} \). The next step is to construct all partial diagnosers with sensor failure labels according to Algorithm 1, being denoted as follows: \( \tilde{G}'_d(\{d, e\}), \tilde{G}'_d(\{a, b\}), \tilde{G}'_d(\{b, c\}), \tilde{G}'_d(\{a, b, c\}), \tilde{G}'_d(\{c, d\}), \tilde{G}'_d(\{a, d\}), \tilde{G}'_d(\{c\}), \tilde{G}'_d(\{b\}), \) and \( \tilde{G}'_d(\{a\}) \). Figures 4(a) and (b) show, respectively, the diagnoser \( \tilde{G}'_d(\{b, c\}) \), obtained according to Definition 9A, and \( \tilde{G}'_d(\{(b, c)\}) \), which has been constructed by following Steps 2 and 3 of Algorithm 1. The resulting union diagnoser \( \tilde{G}_{du}(E_{db,max}) \) is depicted in Figure 5.

It is worth noting that the sensor failure labels in the partial diagnosers are elements of \( E_{uo} \) whereas the transitions in \( \tilde{G}'_d(\{d, e\}) \) are labeled with events in \( E_{uo} \). It is, therefore, possible to identify by inspection of the states of the union diagnoser, which partial diagnoser with sensor failure labels a component comes from. Table 1 presents the guidance to identify the basis for diagnosis used in this example from redundant event sets that appear in the sensor labels. \( \square \)
Fig. 5. Union diagnoser $G_{du}(E_{db,max})$.

Table 1. Relationship between redundant events and diagnosis bases

<table>
<thead>
<tr>
<th>Redundant event set</th>
<th>Diagnosis basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>${d, e}$</td>
<td>${a, b, c}$</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>${c, d, e}$</td>
</tr>
<tr>
<td>${b, c}$</td>
<td>${a, d}$</td>
</tr>
<tr>
<td>${c, d}$</td>
<td>${a, b, e}$</td>
</tr>
<tr>
<td>${a, d}$</td>
<td>${b, c, d}$</td>
</tr>
<tr>
<td>${a}$</td>
<td>${b, c, e}$</td>
</tr>
<tr>
<td>${b}$</td>
<td>${a, c, d}$</td>
</tr>
<tr>
<td>${c}$</td>
<td>${a, b, d}$</td>
</tr>
<tr>
<td>${d}$</td>
<td>${a, b, c}$</td>
</tr>
<tr>
<td>${e}$</td>
<td>${a, b, d, e}$</td>
</tr>
</tbody>
</table>

5. THE ROBUST DIAGNOSER

In the union diagnoser $G_{du}(E_{db,max})$ of Figure 5, we can see the self-loops in following uncertain states: $\{x_pS_a; x_2S_b\} = \{4^1S_a; 6NS_b\}$ and $\{x_3S_ab; x_3S_bc\} = \{6NS_ab; 4Y_{bc}\}$. Notice that when $G_{du}(E_{db,max})$ reaches these states, it will not be sure if the fault has occurred or not, since state $x_3$ is a certain state whereas state $x_2$ is a normal state of $G_d$. In addition, in state $\{x_3S_a; x_3S_b; x_3S_ab; x_3S_ad; x_3S_ba; x_3S_d\}$, the active event sets of the components that come from $G_{du}'(\{c, d\})$ and $G_d'(\{a, d\})$ are both empty due to the existence of hidden cycles in states $\{x_3S_d; x_3S_ad\}$ and $\{x_3S_ba; x_3S_d\}$, respectively. Therefore, it is possible to find two traces with the same projections over $\{a, b, c\}$ and $\{b, c, e\}$, respectively, one with arbitrarily long length containing event $\sigma_f$, and the other one of either finite length or arbitrarily long which does not have the fault event. Therefore, according to Definition 6, L is not robustly diagnosable against permanent sensor failures associated with the sets $E_{\sigma_o}^*; E_{\sigma_o}^\prime, L_{\sigma_o}; L_{\sigma_o}^\prime$, $E_{\sigma_o}^*; E_{\sigma_o}^\prime$, where $P_{a_i}^*; E_{\sigma_o}^*; E_{\sigma_o}^\prime, and E_{\sigma_o} = \{\sigma_f\}$.

It can be concluded, therefore, that the union diagnoser is not necessarily robust, in which case, it must be pruned, in the sense that partial diagnosers with sensor failure labels must be withdrawn in order to make L robustly diagnosable. In this section, besides identifying those partial diagnosers that should be removed from the union diagnoser, we will also present a necessary and sufficient condition for robust diagnosability stated in terms of indeterminate cycles of the union diagnoser, as is the case of regular diagnosers, i.e., those that do not account for sensor failure.

Consider the following definitions.

Definition 11. Let $E_{db}$ denote a set whose elements are diagnosis bases for $L$ and let $G_{du}(E_{db}) = (X_{du}, E_{du}, I_{du}, \Gamma_{du}, x_{0,du})$ denote the union diagnoser formed with these bases. A state $x_{du} \in X_{du}$ is called certain if for all $x_dS_m \in x_{du}$, $x_d$ is a certain state. If there exist $x_dS_m, y_dS_m \in x_{du}$ such that $x_d$ is certain and $y_d$ is either normal or uncertain, then $x_{du}$ is an uncertain state of $G_{du}(E_{db})$.

Definition 12. A set of uncertain states $\{x_{du,1}, x_{du,2}, \ldots, x_{du,p}\}$ of $G_{du}$ forms an indeterminate observed cycle if the following conditions are met:

U.1) $\{x_{du,1}, x_{du,2}, \ldots, x_{du,p}\}$ forms a cycle in $G_{du}(E_{db})$;
U.2) $\exists x_dS_m^{(i)}, \ldots, x_dS_m^{(l)} \in x_{du},$ where $x_dS_m^{(i)}$ is certain and $x_dS_m^{(l)}$ is either uncertain or normal, for $l = 1, 2, \ldots, p$, $k_l = 1, 2, \ldots, q$, and $r_l = 1, 2, \ldots, q_i$ such that the sequences of states $\{x_dS_m^{(i)}\}; l = 1, 2, \ldots, p, k_l = 1, 2, \ldots, q_l$ and
{\tilde{x}_{d_i}, n_{d_i}}$, $i = 1, 2, \ldots, p$, $n_{r_i} = 1, 2, \ldots, q_i$ form cycles in two different partial diagnosers with sensor failure labels.

Since partial diagnosers may have hidden cycles\(^1\), it is possible that the union diagnoser might also have hidden cycles, as follows.

**Definition 13.** There exists an indeterminate hidden cycle in an uncertain state of $G_{d_1}(E_{db})$ if one component of this state is a certain state of a partial diagnoser with sensor failure label in which there is a hidden cycle.

The following result may be stated.

**Theorem 2.** Let $L$ be the language generated by automaton $G$ and assume that $E_{db} = \{E_{o_1}, E_{o_2}, \ldots, E_{o_m}\}$, where $E_{o_i}$, $i = 1, 2, \ldots, m$ are either minimum or nonminimum diagnosis bases for $L$ and let $E_{rob}$ be defined as in Equation (2). Then $L$ is robustly diagnosable against permanent sensor failures associated with the events in the sets of $E_{rob}$ with respect to projections $P^r_{o_1}, P^r_{o_2}, \ldots, P^r_{o_m}$ and $E_y = \{\sigma_f\}$, if, and only if, the union diagnoser $G_{d_1}(E_{db})$ has no indeterminate cycles (observed or hidden).

**Proof.** See Lima (2010).

It is therefore possible to conclude from Theorem 2 and from the way an union diagnoser is built that $L$ can be made robustly diagnosable against permanent sensor failures associated with $E'_{uo}$, where $E'_{o} \in E_{db}$, with $E_{rob}$, by removing, from the union diagnoser, the partial diagnosers with sensor failure labels with states that are components of states of the union diagnoser that form indeterminate (observed or hidden) cycles. Indeed, a more conservative approach would be the removal of all partial diagnosers that take part in indeterminate cycles. However, since the main objective of a fault diagnosis system is to infer fault occurrences, only the partial diagnosers with normal state components in indeterminate cycles of the union diagnoser will be removed. The following algorithm describes how to prune the union diagnoser so as to obtain a robust diagnoser that satisfies properties P1-P4.

**Algorithm 2.**

1. **Step 1** Find all indeterminate (observed and hidden) cycles of $G_{d_1}(E_{db,max})$ and identify all partial diagnosers with sensor labels with normal components in the states that form the indeterminate cycles.

2. **Step 2** Define a new set $E_{ic} = E_{db,max} \setminus E_{ic}$, where

$$E_{ic} = \{E_{o} \in E_{db,max} : G_{d_1}(E_{o}) \setminus E_{db,max} \}$$

3. **Step 3** Obtain another union diagnoser formed with the partial diagnosers with sensor labels formed with the sets in $E_{db}$.

It is worth pointing out that there is no need to iterate over the steps of Algorithm 2 and that the resulting diagnoser is guaranteed to be robust.

The following example explains how to obtain a robust diagnoser from a non-robust union diagnoser.

**Example 2.** Let us consider the union diagnoser shown in Figure 5. As mentioned before, states $\{x_3S_3; x_3S_4\}$ and $\{x_3S_{ad}; x_3S_{ad}\}$ form observed indeterminate cycles due to the partial diagnosers with sensor failure labels $G_{d_1}(\{x\})$ and $G_{d_1}(\{a, b\})$ for the former and to $G_{d_1}(\{a, b\})$ and $G_{d_1}(\{b, c\})$ for the latter. In addition, state $\{x_3S_3; x_3S_4; x_3S_{ad}; x_3S_{ad}; x_3S_{ad}\}$ has indeterminate hidden cycles since $G_{d_1}(\{c, d\})$ and $G_{d_1}(\{a, d\})$ have hidden cycles in states $\{x_3S_{ad}; x_3S_{ad}\}$ and $\{x_3S_{ad}; x_3S_{ad}\}$, respectively.

Therefore, according to Theorem 2, we may conclude that $L$ is not robustly diagnosable against permanent sensor failure associated with the sets $E'_{uo}$, for all $E'_{o} \in E_{db,max}$ with respect to projections $P^r_{o_1}, \ldots, P^r_{o_m}$, and $E_y = \{\sigma_f\}$. In accordance with Step 2 of Algorithm 2, the following partial diagnosers with sensor failure labels should be removed from $G_{d_1}(E_{db,max})$: $G_{d_1}(\{a, d\})$, $G_{d_1}(\{a, b, c\})$, $G_{d_1}(\{a, b, c, d\})$. This implies that:

$$E_{ic} = \{\{b, c, d, e\}, \{c, d, e\}, \{b, c, e\}\}$$

and, thus, the the set of diagnosis bases to be used in the construction of the robust diagnoser is given by:

$$E_{db} = \{\{a, b, c\}, \{a, d, e\}, \{a, b, c\}, \{a, b, d, c\}, \{a, c, d, e\}, E_{o}\}$$

The corresponding robust diagnoser $G_{rob}(E_{db})$ is depicted in Figure 6. Notice that since $G_{rob}(E_{db})$ has no indeterminate cycles, $L$ is robustly diagnosable with respect to all $E'_{uo} \in E_{rob}$, where

$$E_{rob} = \{\{d, e\}, \{b, e\}, \{b, c, d, e\}, \{c, e\}, \{c, d, e\}\}$$

and $P^r_{o} : E^* \rightarrow E^*$ for all $E'_{o} \in E_{db}$, and $E_y = \{\sigma_f\}$.

**Remark 1.** For online diagnosis it is not necessary to use the union diagnoser. Instead, all partial diagnosers can run in parallel, starting at their initial states and after the occurrence of an observable event, those partial diagnosers whose active event sets of the initial states possess the event that has just occurred move to the next state, whereas all the other partial diagnosers are discarded. This process continues until the system is reset, when all partial diagnosers come into play again.

6. CONCLUSION

We have proposed a robust diagnoser that deploys the redundancy that may exist in a set formed of diagnosis bases with a view to ensure the fault diagnosis even in the occurrence of permanent sensor failures. To achieve robustness, we proposed an approach where several partial diagnosers, each one built for a particular diagnosis basis, are deployed. We have given necessary and sufficient conditions for robust diagnosability.

REFERENCES

Fig. 6. Robust diagnoser $G_{rob}(E_{th})$.


