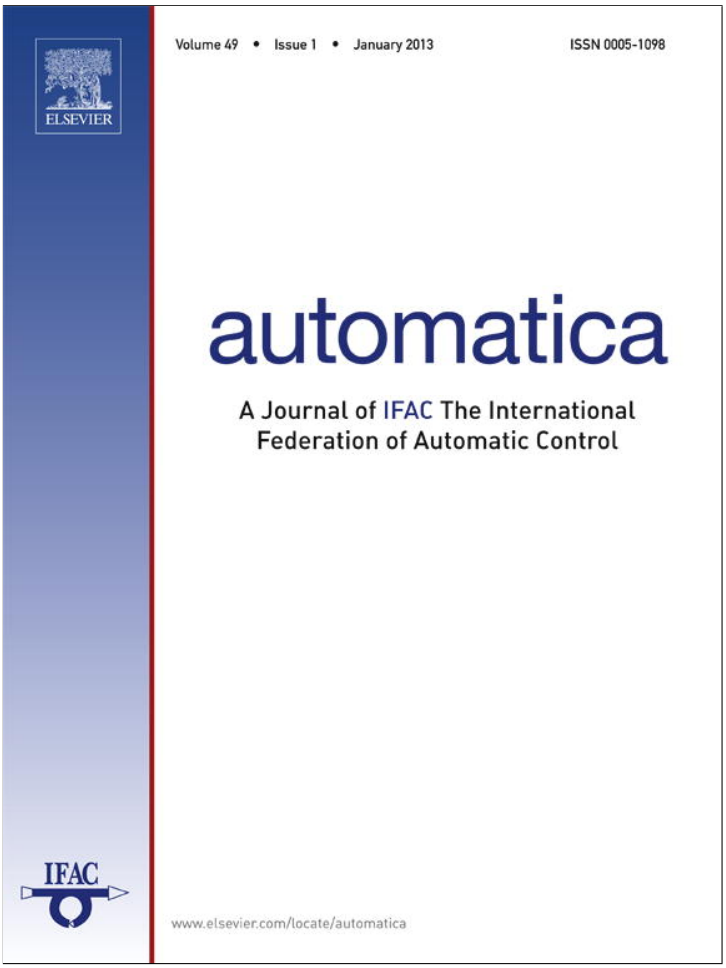

Automatica 49 (2013) 223–231

Contents lists available at SciVerse ScienceDirect

Automatica

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

Brief paper

Robust diagnosis of discrete-event systems against permanent loss of observations☆

Lilian K. Carvalho [a], Marcos V. Moreira [a], João C. Basilio [a,1], Stéphane Lafortune [b]

[a] Universidade Federal do Rio de Janeiro, COPPE - Programa de Engenharia Elétrica, 21949-900, Rio de Janeiro, RJ, Brazil

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

ARTICLE INFO

Article history:
Received 20 July 2011
Received in revised form
20 April 2012
Accepted 30 July 2012
Available online 9 October 2012

Keywords:
Discrete-event systems
Fault diagnosis
Sensor failures
Robust diagnosability

ABSTRACT

We consider the problem of diagnosing the occurrence of a certain unobservable event of interest, the *fault* event, in the operation of a partially-observed discrete-event system subject to permanent loss of observations modeled by a finite-state automaton. Specifically, it is assumed that certain sensors for events that would *a priori* be observable may fail at the outset, thereby resulting in a loss of observable events; the diagnostic engine is not directly aware of such sensor failures. We explore a previous definition of robust diagnosability of a given fault event despite the possibility of permanent (and unknown *a priori*) loss of observations and present a polynomial time verification algorithm to verify robust diagnosability and a methodology to perform online diagnosis in this scenario using a set of partial diagnosers.

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. The property of diagnosability formally captures the ability to always detect at run-time any occurrence of the given fault event, within a finite number of event transitions. There is a very large body of literature on (offline) diagnosability analysis and (online) event diagnosis of discrete-event systems modeled by automata, the modeling formalism considered in this paper; see, e.g., Boel and van Schuppen (2002), Debouk, Lafortune, and Teneketzis (2000), Genc (2008), Jéron, Marchand, Pinchinat, and Cordier (2006), Kumar and Takai (2009), Lin (1994), Lunze and Schröder (2004), Pencolé and Cordier (2005), Qiu and Kumar (2006), Sampath, Sengupta, Lafortune, Sinnamohideen, and Teneketzis (1995), Thorsley and Teneketzis (2005), Tripakis (2002), Wang, Yoo, and Lafortune (2007), Ye, Dague, and Yan (2009), Yoo and Lafortune (2002), Zad, Kwong, and Wonham (2003) and the references contained therein. Two classes of automata derived from the automaton model of the system have been defined in the above works: diagnosers and verifiers. Both diagnosers and verifiers can be used for offline analysis of diagnosability properties; online diagnosis is usually implemented using diagnosers.

Let us assume that the given set of sensors attached to the system 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 and unknown *a priori*. In this case, if online diagnosis is performed using a standard diagnoser built on the basis of all potentially observable events, then this diagnoser could get stuck in some states (e.g., no further observed event, or occurrence of an event not in the current 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 (unknown *a priori*) loss of observations resulting from sensor failures.

Recently, there have been some works on sensor failures in supervisory control of discrete-event systems (see, e.g., Rohloff (2005); Sanchez and Montoya (2006)), on various notions of "robust" diagnosis of discrete-event systems in the presence of potentially faulty sensors, in particular, Basilio and Lafortune (2009), Carvalho, Basilio, and Moreira (2010, 2012), Contant, Lafortune, and Teneketzis (2006), and Takai (2010, 2012) and on

☆ This work was partially supported by the Brazilian Research Council (CNPq) grant 200820/2006-0 and by the US National Science Foundation grant EECS-0624821. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Jan Komenda, under the direction of Editor Ian R. Petersen.

E-mail addresses: lilian@coep.ufrj.br (L.K. Carvalho), moreira@dee.ufrj.br (M.V. Moreira), basilio@poli.ufrj.br (J.C. Basilio), stephane@umich.edu (S. Lafortune).

[1] Tel.: +55 21 2562 8021; fax: +55 21 2562 8627.

0005-1098/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2012.09.017

fault diagnosis under unreliable observations (Athanasopoulou, Lingxi, & Hadjicostis, 2010; Thorsley, Yoo, & Garcia, 2008).

In this paper, we deal with the problem of robust diagnosis against permanent loss of observations. This problem was first introduced by Lima, Basilio, Lafortune, and Moreira (2010), and can be stated as follows. Let us assume that a given unobservable fault event, σ_f, is diagnosable in a given system for the set of all observable events Σ_o, in the sense of Sampath et al. (1995). Let $\Sigma_o' \subset \Sigma_o$ be a proper subset of Σ_o for which diagnosability still holds. Then Σ_o' is called a diagnosis basis (Basilio, Lima, Lafortune, & Moreira, 2012) and the events in the set $\Sigma_o \setminus \Sigma_o'$ are said to be *redundant*; we call $\Sigma_{uo}' := \Sigma_o \setminus \Sigma_o'$ the set of redundant events associated with Σ_o'; the partial diagnoser built for Σ_o' does not record these (potentially observable) events. Lima et al. (2010) present a necessary and sufficient condition for robust diagnosability against permanent sensor failures using a union diagnoser, i.e., a diagnoser that accepts the union of the languages of all partial diagnosers formed with all sets Σ_o' that are diagnosis bases. It is not difficult to see that union diagnosers tend to have huge state spaces, which makes the verification test very computationally demanding.

In order to overcome the potential state space explosion of union diagnosers, we propose in this paper an offline test based on the use of a special type of verifier automata. This procedure avoids the worst-case exponential complexity of diagnosers, as verifiers can be computed in the worst-case polynomial time in the size of the system. We also discuss how to perform online diagnosis for systems that are robust diagnosable.

This paper is structured as follows. In Section 2, we present some background on fault diagnosis of discrete-event systems. In Section 3, we present the definition of robust diagnosability against permanent loss of observations. In Section 4 we develop an offline test for the verification of robust diagnosability and in Section 5 we discuss the online implementation of robust diagnosers. Finally, conclusions are drawn in Section 6.

2. Preliminaries

Let

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

be a deterministic automaton, where X denotes the state space, Σ the event set, $f : X \times \Sigma \to X$ the state transition function, which is partially defined over its domain, Γ the active event set, and x_0 the initial state. Let us partition Σ as $\Sigma = \Sigma_o \dot\cup \Sigma_{uo}$, i.e., $\Sigma = \Sigma_o \cup \Sigma_{uo}$, $\Sigma_o \cap \Sigma_{uo} = \emptyset$ and $\Sigma_{uo} \neq \emptyset$, where Σ_o and Σ_{uo} are, respectively, the set of observable and unobservable events, and let $\Sigma_f = \{\sigma_f\} \subseteq \Sigma_{uo}$ be a set whose unique element σ_f is the fault event to be detected. Finally, let us denote the language generated by G as L. We make the following common assumptions:

A1. 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.

The language L is said to be diagnosable if the occurrence of σ_f can be detected within a finite number of transitions after the occurrence of σ_f using only traces formed with events in Σ_o. Let the function $P_o : \Sigma^* \to \Sigma_o^*$ denote the standard natural projection that erases unobservable events; see Cassandras and Lafortune (2008). In addition, let P_o^{-1} denote the inverse projection of P_o, and assume that $L/s = \{t \in \Sigma^* : st \in L\}$ and that $\Psi(\Sigma_f)$ denotes the set of all traces of L that end with event σ_f. With some abuse of notation $\Sigma_f \in s$ denotes that $\bar s \cap \Psi(\Sigma_f) \neq \emptyset$. Language diagnosability can then be formally defined as follows (Sampath et al., 1995).

Definition 1. L is diagnosable with respect to $P_o : \Sigma^* \to \Sigma_o^*$ if, and only if, the following condition holds true:

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

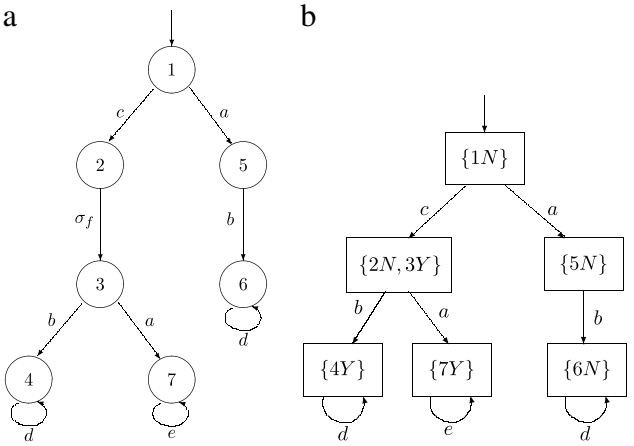

Fig. 1. Automaton G (a) and its diagnoser G_d (b).

where the diagnosability condition D is given as

$$(\forall \omega \in (P_o^{-1}(P_o(st)) \cap L))(\Sigma_f \in \omega).$$

3. Robust diagnosability against permanent loss of observations

Language diagnosability is usually performed in practice using diagnosers. A diagnoser is a deterministic automaton, which is built from the automaton that generates the language to be diagnosed and whose event set is formed with the observable events of G, and whose states are sets of states of G augmented by adding labels Y or N to indicate whether the fault event σ_f has occurred or not in reaching the state. In this regard, a state x_d of the diagnoser is called certain (or faulty) if $\ell = Y$ for all $x\ell \in x_d$, and normal (or non-faulty) if $\ell = N$ for all $x\ell \in x_d$. If there exist $x\ell, y\tilde\ell \in x_d$, x not necessarily distinct from y such that $\ell = Y$ and $\tilde\ell = N$, then x_d is an uncertain state of G_d. When the diagnoser reaches a certain (resp. normal) state, we are certain that the fault has occurred (resp. not occurred). However, when the diagnoser is in an uncertain state, we cannot draw any conclusion regarding the fault occurrence. If the diagnoser remains indefinitely in a cycle formed with uncertain states only, then it will not be possible to diagnose the fault occurrence.[2]

Fig. 1(a) shows the state transition diagram of an automaton G, for which $\Sigma = \{a, b, c, d, e, \sigma_f\}$, $\Sigma_o = \{a, b, c, d, e\}$, and $\Sigma_f = \{\sigma_f\}$. The corresponding diagnoser is depicted in Fig. 1(b). Notice that, since G_d has cycles in certain and normal states only, then we may say that L is diagnosable with respect to P_o and Σ_f. Indeed, if trace $s_Y = c\sigma_f bd^n$ $(n \in \mathbb{N})$ occurs, then the diagnoser goes from the initial state $\{1N\}$ to state $\{4Y\}$, indicating that the fault event σ_f has occurred. Assume now that a permanent loss of observation of the sensor that records the occurrence of event c took place before the first occurrence of c, and suppose that trace $s_Y = c\sigma_f ae^n, n \in \mathbb{N}$, has been generated. Since event σ_f is unobservable, the first event to be recognized by the diagnoser of Fig. 1(b) 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 s_Y and it 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, it will not be able to reach a certain

[2] We refer the reader to Basilio et al. (2012) for a more detailed explanation about diagnosers and how they can be used as an offline test for diagnosability.

state, as it should, since trace s_Y contains the fault event σ_f and has arbitrarily long length. This fact suggests that not only diagnosers should be modified to perform correctly in practice but also a new diagnosability test that takes into account possible permanent loss of observations must be developed.

In order to do so, we will first introduce a language diagnosability condition that takes into account possible permanent loss of observations. Such a diagnosability condition will be referred to here as a robust diagnosability condition against permanent loss of observations. Robustness here should be understood in the sense that L remains diagnosable even in the case of permanent loss of observations.

We make the following additional assumptions.

A3. L is diagnosable with respect to $P_o : \Sigma^* \to \Sigma_o^*$ and $\Sigma_f = \{\sigma_f\}$.

A4. Any loss of observations, when it occurs, takes place before the first occurrence of the (initially observable) event associated with the sensor that has failed, and it is permanent, i.e., the event remains unobservable.

It is worth noting that Assumption A4 is not restrictive in the case of a cyclical system that observably returns to its initial state.

Consider now the following definition (Basilio et al., 2012).

Definition 2 (*Diagnosis Basis*)**.** A set $\Sigma_o' \subseteq \Sigma_o$ is a diagnosis basis if L is also diagnosable with respect to projection $P_o' : \Sigma^* \to \Sigma_o'^*$ and $\Sigma_f = \{\sigma_f\}$. If for any nonempty subset Σ_o'' of Σ_o', L is not diagnosable with respect to projection $P_o'' : \Sigma^* \to \Sigma_o''^*$ and $\Sigma_f = \{\sigma_f\}$ then Σ_o' is a minimal diagnosis basis.

According to Definition 2, the sets that are diagnosis bases ensure diagnosability of L and thus use the redundancy of the events in $\Sigma_o \setminus \Sigma_o'$ to provide some robustness to the diagnosing system. This leads to the definition of robust diagnosability against permanent loss of observations.

Definition 3 (*Robust Diagnosability Against Permanent Loss of Observations*)**.** Let $\Sigma_{db} = \{\Sigma_{o_1}, \Sigma_{o_2}, \ldots, \Sigma_{o_m}\}$, where $\Sigma_{o_i}, i = 1, 2, \ldots, m$ are either minimal or nonminimal diagnosis bases for L. Define the set

$$\Sigma_{rob} = \{\Sigma_{uo_1}, \Sigma_{uo_2}, \ldots, \Sigma_{uo_m}\}, \tag{2}$$

where

$$\Sigma_{uo_i} = \Sigma_o \setminus \Sigma_{o_i}, \quad i = 1, 2, \ldots, m. \tag{3}$$

Then L is robustly diagnosable with respect to projections $P_{o_1}, P_{o_2}, \ldots, P_{o_m}$, where $P_{o_i} : \Sigma^* \to \Sigma_{o_i}^*$, and $\Sigma_f = \{\sigma_f\}$, or equivalently, with respect to permanent loss of observation of the events of all sets Σ_{uo_i}, $i = 1, 2, \ldots, m$, and $\Sigma_f = \{\sigma_f\}$, if the following condition holds true:

$$\begin{aligned}&(\exists n \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall t \in L/s)\\ &(\|t\| \geq n \Rightarrow R_D),\end{aligned} \tag{4}$$

where the robust diagnosability condition R_D is given as

$$\begin{aligned}&(\forall i, j \in \{1, 2, \ldots, m\}, i \neq j)\\ &(\nexists \omega_j \in L) \quad [\Sigma_f \notin \omega_j \wedge P_{o_i}(st) = P_{o_j}(w_j)].\end{aligned}$$

The idea behind Definition 3 is that since L is diagnosable with respect to $P_{o_i} : \Sigma^* \to \Sigma_{o_i}^*$, and $\Sigma_f = \{\sigma_f\}$, and assuming that all partial diagnosers for $\Sigma_{o_k}, k = 1, 2, \ldots, m$ are running simultaneously and have access to all available sensors, any partial diagnoser, say Σ_{o_i}, only performs properly if all events in Σ_{uo_i} become unobservable, i.e., observation of all events in Σ_{uo_i} is lost. 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_Y that contains the fault event has the same projection over, say $\Sigma_{o_i}^*$ and $\Sigma_{o_j}^*$, where the former takes G_{d_i} to a certain state whereas the latter takes G_{d_j} to a normal state. In this case, according to Definition 3, L is not robustly diagnosable against permanent loss of observation of the events in Σ_{uo_i} and Σ_{uo_j}.

Example 1. Let us consider automaton G whose state transition diagram is shown in Fig. 1(a) and the following subsets of the set of observable events of G: $\Sigma_{o_1} = \{a, d, e\}$ and $\Sigma_{o_2} = \{c, d, e\}$. The partial diagnosers $G_{d_1}(\{b, c\})$ and $G_{d_2}(\{a, b\})$ that observe Σ_{o_1} and Σ_{o_2} are shown in Fig. 2(a) and (b), respectively. We can therefore conclude that L (the language generated by G) is diagnosable with respect to Σ_f, P_{o_1} and P_{o_2}. Consider now the following traces: $s_Y = c\sigma_f bd^n$ and $s_N = abd^n$. It is clear that if trace s_Y occurs, both diagnosers will, in the end, be at the certain state $\{4Y\}$, indicating that the fault event σ_f has occurred. Similarly, if trace s_N occurs, then both diagnosers will end up at state $\{6N\}$, indicating that the fault has not occurred. However, since $P_{o_1}(s_Y) = P_{o_2}(s_N) = d^n$ then when both partial diagnosers work concurrently and assuming that sensors may fail, it is not possible to state that either trace s_Y occurred and sensors b and c have failed or trace s_N occurred and sensors a and b have failed. Therefore, L is not robustly diagnosable with respect to projections P_{o_1}, P_{o_2} and Σ_f or, equivalently, with respect to permanent loss of observation of all events in $\Sigma_{uo_1} = \{b, c\}$, $\Sigma_{uo_2} = \{a, b\}$ and Σ_f. □

In summary, Definition 3 ensures that if a fault occurs, all partial diagnosers that are still running will eventually agree, i.e., they will all be in certain states, ensuring the unambiguous detection of the fault.

4. Verification of robust diagnosability against loss of observations using verifiers

Verification of language diagnosability can be performed in polynomial time in the number of states and events of the system by using verifiers (Jiang, Huang, Chandra, & Kumar, 2001; Moreira, Jesus, & Basilio, 2010, 2011; Qiu & Kumar, 2006; Yoo & Lafortune, 2002), in contrast to the construction of diagnosers that has worst-case exponential time in the number of states of the system. In this section, we will develop a test for robust diagnosability with respect to permanent loss of observations using suitably-defined verifier automata.

Let us split the nominal automaton G into several automata, each one having as observable events a diagnosis bases. Let

$$\Sigma_{db} = \{\Sigma_{o_1}, \Sigma_{o_2}, \ldots, \Sigma_{o_m}\},$$

denote a set of diagnosis bases, where $\Sigma_{o_i}, i = 1, 2, \ldots, m$ is either a minimal or a nonminimal diagnosis basis for L. Consider the following renaming of events, which is used in Moreira et al. (2010, 2011):

$$R_i(\sigma) = \begin{cases}\sigma, & \text{if } \sigma \in \Sigma_{o_i} \cup \Sigma_f\\ \sigma_{R_i}, & \text{if } \sigma \in (\Sigma_{uo} \setminus \Sigma_f) \cup \Sigma_{uo_i},\end{cases} \tag{5}$$

where Σ_{uo_i} is defined according to Eq. (3). Notice that R_i only renames events in $(\Sigma_{uo} \setminus \Sigma_f) \cup \Sigma_{uo_i}$. The domain of R_i can be extended to Σ^* as follows: (i) $R_i(\varepsilon) = \varepsilon$; and (ii) $R_i(s\sigma) = R_i(s)R_i(\sigma)$, $\forall s \in \Sigma^*$ and $\forall \sigma \in \Sigma$. Similarly, R_i can be extended to languages $L \subseteq \Sigma^*$ by applying Eq. (5) to all traces of L.

Denoting

$$\begin{aligned}\Sigma_{R_i} = \Sigma_{o_i} \cup \Sigma_f \cup \{\sigma_{R_i} : (\exists \sigma \in (\Sigma_{uo} \setminus \Sigma_f) \cup \Sigma_{uo_i})\\ [\sigma_{R_i} = R_i(\sigma)]\},\end{aligned} \tag{6}$$

then we can define the inverse renaming function R_i^{-1}, as follows:

$$\begin{aligned}&R_i^{-1} : \Sigma_{R_i} \to \Sigma\\ &\sigma_{R_i} \mapsto \sigma : \sigma_{R_i} = R_i(\sigma).\end{aligned} \tag{7}$$

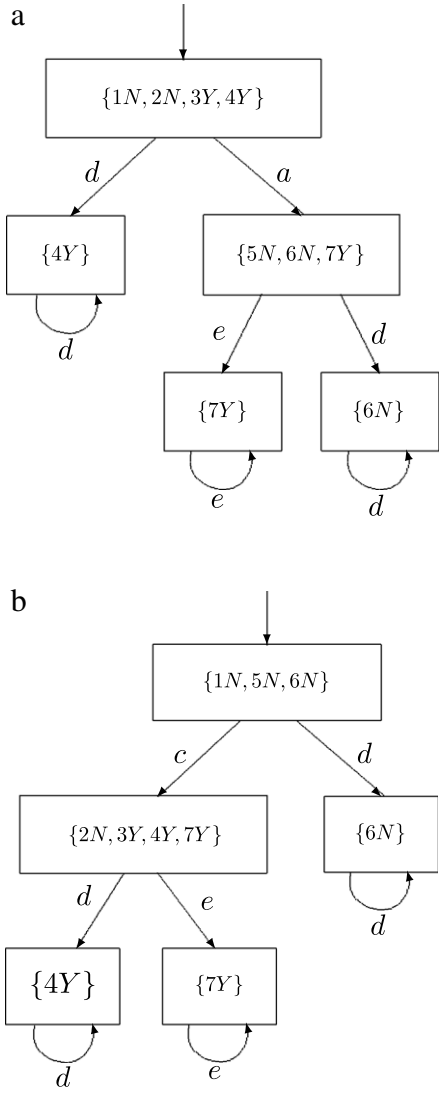

Fig. 2. $G_{d_1}(\{b, c\})$ (a) and $G_{d_2}(\{a, b\})$ (b).

As for R_i, we can extend R_i^{-1} to domain $\Sigma_{R_i}^*$, as follows: (i) $R_i^{-1}(s_{R_i}\sigma_{R_i}) = R_i^{-1}(s_{R_i})R_i^{-1}(\sigma_{R_i})$ for all $s_{R_i} \in \Sigma_{R_i}^*$ and $\sigma_{R_i} \in \Sigma_{R_i}$; and (ii) $R_i^{-1}(\varepsilon) = \varepsilon$.

Let us define

$$\Sigma_{db,o} = \bigcup_{i=1}^{m} \Sigma_{o_i}, \tag{8}$$

and assume that G_i denotes an automaton whose generated language is $L_i = R_i(L)$. A verifier to check robust diagnosability with respect to permanent loss of observations can be constructed according to the following algorithm.

Algorithm 1 (*Robust Diagnosability Verification*)**.**

Step 1 Build the set of automata $G_i = (X, \Sigma_{R_i}, f_i, \Gamma_i, x_0)$, $i = 1, 2, \ldots, m$, where Σ_{R_i} is defined in Eq. (6), $\Gamma_i(x) = R_i[\Gamma(x)]$, and $f_i(x, R_i(\sigma)) = f(x, \sigma)$ for all $x \in X$ and $\sigma \in \Gamma(x)$.

Step 2 Build the set of automata G_{F_i}, $i = 1, 2, \ldots, m$ where each G_{F_i} models the failure behavior of G_i, as follows:

- Step 2.1. Construct label automaton $A_\ell = (X_\ell, \Sigma_f, f_\ell, x_{0,\ell})$, where $X_\ell = \{N, Y\}$, $x_{0,\ell} = \{N\}$, $f_\ell(N, \sigma_f) = Y$ and $f_\ell(Y, \sigma_f) = Y$, for all $\sigma_f \in \Sigma_f$.
- Step 2.2. Compute $G_{\ell_i} = G_i \parallel A_\ell$, $i = 1, 2, \ldots, m$, and mark all states of G_{ℓ_i} whose second component is Y.
- Step 2.3. Obtain the failure automaton $G_{F_i} = CoAc(G_{\ell_i}) = (X_{F_i}, \Sigma_{R_i}, f_{F_i}, x_{0,F_i})$, $i = 1, 2, \ldots, m$, where $CoAc(G_{\ell_i})$ denotes the coaccessible part of automaton G_{ℓ_i}, i.e., the automaton whose states are all coaccessible.[3]
- Step 2.4. Redefine the event set of G_{F_i} as $\Sigma_{F_i} = \Sigma_{R_i} \cup \Sigma_{db,o}$.
- Step 2.5. Unmark all marked states of G_{F_i}.

Step 3 Build the set of normal behavior automata G_{N_i}, $i = 1, 2, \ldots, m$, as follows:

- Step 3.1. Define $\Sigma_{A_{N_i}} = \Sigma_{R_i} \setminus \Sigma_f$, and construct automaton $A_{N_i} = (\{N\}, \Sigma_{A_{N_i}}, f_{A_{N_i}}, \Sigma_{A_{N_i}}, N)$, composed of a single state N with a self-loop labeled with all events in Σ_{R_i} except the fault event.
- Step 3.2. Construct the nonfailure automaton $G_{N_i} = G_i \times A_{N_i} = (X_{N_i}, \Sigma_{R_i}, f_{N_i}, \Gamma_{N_i}, x_{0,N_i})$.
- Step 3.3. Redefine the event set of G_{N_i} as $\Sigma_{N_i} = (\Sigma_{R_i} \setminus \Sigma_f) \cup \Sigma_{db,o}$.

Step 4 For each pair (i, j), $i, j = 1, 2, \ldots, m$, $j \neq i$, construct the verifier automaton

$$G_{V_{ij}} = G_{F_i} \parallel G_{N_j}$$

whose states are of the form

$$x_{V_{ij}} = (x_{F_i}, x_{N_j}),$$

where x_{F_i} and x_{N_j} are states of G_{F_i} and G_{N_j}, respectively, and $x_{F_i} = (x, x_\ell)$ where $x \in X$ and $x_\ell \in \{N, Y\}$.

Step 5 Test for the existence of a cyclic path

$$cl = (x_{V_{ij}}^k, \sigma_k, x_{V_{ij}}^{k+1}, \sigma_{k+1}, \ldots, \sigma_l, x_{V_{ij}}^l),$$

where $l \geq k > 0$, in at least one verifier $G_{V_{ij}}$, for $i, j = 1, , 2, \ldots, m$, $j \neq i$, satisfying the following conditions:

$$\exists q \in \{k, k+1, \ldots, l\} \quad \text{s.t. } (x_\ell^q = Y) \wedge (\sigma_q \in \Sigma_{R_i}).$$

If such a cl exists, then L is not robustly diagnosable with respect to projections P_{o_i} and P_{o_j}, and Σ_f. Otherwise, L is robustly diagnosable. □

Remark 1. According to Steps 2 and 3 of Algorithm 1, the event sets of G_{F_i} and G_{N_j} are given as $\Sigma_{R_i} \cup \Sigma_{db,o}$ and $(\Sigma_{R_j} \setminus \Sigma_f) \cup \Sigma_{db,o}$, respectively. Thus, the renamed events of Σ_{R_i} and Σ_{R_j} are private events of G_{F_i} and G_{N_j}, respectively, whereas the events in $\Sigma_{db,o}$ belong to both automata. However, since the event traces of G_{F_i} belong to $\Sigma_{R_i}^*$ and the traces of G_{N_j} belong to $(\Sigma_{R_j} \setminus \Sigma_f)^*$ and $G_{V_{ij}} = G_{F_i} \parallel G_{N_j}$, then an event $\sigma \in \Sigma_{db,o}$ belongs to a trace $s \in L(G_{V_{ij}})$, if and only if $\sigma \in \Sigma_{o_i} \cap \Sigma_{o_j}$. □

We now present the correctness proof of Algorithm 1.

Theorem 1. *L is not robustly diagnosable with respect to projections P_{o_i}, $i = 1, 2, \ldots, m$, and Σ_f if and only if there exists a cyclic path*

$$cl = (x_{V_{ij}}^k, \sigma_k, x_{V_{ij}}^{k+1}, \sigma_{k+1}, \ldots, \sigma_l, x_{V_{ij}}^l),$$

where $l \geq k > 0$, in some verifier $G_{V_{ij}}$, $i, j \in I_m := \{1, 2, \ldots, m\}$, $j \neq i$, satisfying the following conditions:

$$\exists q \in \{k, k+1, \ldots, l\} : (x_{V_{ij}} = ((x^q, Y), x_{N_j})) \wedge (\sigma_q \in \Sigma_{R_i}), \tag{9}$$

where (x^q, Y) and x_{N_j} are states of G_{F_i} and G_{N_j}, respectively.

Proof. ($\Longleftarrow$) Let us assume that there exists a cyclic path $cl = (x_{V_{ij}}^k, \sigma_k, x_{V_{ij}}^{k+1}, \sigma_{k+1}, \ldots, \sigma_l, x_{V_{ij}}^l)$, where $l \geq k > 0$, in verifier $G_{V_{ij}}$ satisfying condition (9). Since $x_{F_i}^q = (x^q, Y)$ for some $q \in \{k, k+1, \ldots, l\}$, then, from the construction of $G_{V_{ij}}$ it can be seen

[3] A state x of an automaton $G = (X, \Sigma, f, \Gamma, x_0, X_m)$ is coaccessible if there exists a trace $s \in \Sigma^*$ such that $f(x, s) \in X_m$.

that $x^q_{F_i} = (x^q, Y)$ for all $q \in \{k, k+1, \ldots, l\}$. This implies that there exists a trace $s't' \in L(G_{V_{ij}})$, such that s' contains the fault event, and $t' = (\sigma_k\sigma_{k+1}\ldots\sigma_l)^p$, $p \in \mathbb{N}$, where $|t'| > n, \forall n \in \mathbb{N}$.

Define now the following projection operations:

$$P_{F_i} : \Sigma_{F_i} \cup \Sigma_{N_j} \to \Sigma_{F_i},$$
$$P_{N_j} : \Sigma_{F_i} \cup \Sigma_{N_j} \to \Sigma_{N_j},$$
$$P : \Sigma_{F_i} \cup \Sigma_{N_j} \to \Sigma_{o_i} \cap \Sigma_{o_j},$$
$$P_i : \Sigma_{R_i} \to \Sigma_{o_i},$$
$$P_j : \Sigma_{R_j} \to \Sigma_{o_j}.$$

Notice that P_i and P_j become, respectively, equivalent to P_{o_i} and P_{o_j} if the renaming is removed.

Since $G_{V_{ij}} = G_{F_i} \parallel G_{N_j}$, then $\mathcal{L}(G_{V_{ij}}) = P^{-1}_{F_i}[L(G_{F_i})] \cap P^{-1}_{N_j}[L(G_{N_j})]$, which implies that $s't' \in P^{-1}_{F_i}[L(G_{F_i})]$. Let $\tilde{s}\tilde{t} = P_{F_i}(s't')$, where $\tilde{s} = P_{F_i}(s')$ and $\tilde{t} = P_{F_i}(t')$. Thus, since $P_{F_i}\left[P^{-1}_{F_i}(L(G_{F_i}))\right] = L(G_{F_i})$, then $\tilde{s}\tilde{t} \in L(G_{F_i})$. In addition, since $t' = (\sigma_k\sigma_{k+1}\ldots\sigma_l)^p$, where $|t'| > n, \forall n \in \mathbb{N}$, and, by assumption, there exists an event $\sigma_q \in \Sigma_{R_i}$ for $q \in \{k, k+1, \ldots, l\}$ and $\Sigma_{R_i} \subset \Sigma_{F_i}$, then the event sequence $\tilde{t} = P_{F_i}(t')$ also has arbitrarily long length, which implies that $\tilde{s}\tilde{t} \in L(G_{F_i})$ also has arbitrarily long length after the occurrence of the fault event σ_f. Notice that G_i is obtained from G after renaming the event set Σ as Σ_{R_i}. Thus, there exists a fault trace st of arbitrarily long length after a fault event $\sigma_f \in \Sigma_f$, such that $P_{o_i}(st) = P_i(\tilde{s}\tilde{t})$.

Let $\tilde{w} = P_{N_j}(s't')$. Since $s't' \in L(G_{V_{ij}})$, then $s't' \in P^{-1}_{N_j}[L(G_{N_j})]$. In addition, $P_{N_j}[P^{-1}_{N_j}(L(G_{N_j}))] = L(G_{N_j})$, which implies that $\tilde{w} \in \mathcal{L}(G_{N_j})$. Notice that G_j is obtained from G after renaming the events of Σ according to function R_j. Thus, there exists a trace $w \in L(G)$, where $\Sigma_f \notin w$, such that $P_{o_j}(w) = P_j(\tilde{w})$.

To conclude the proof, notice that

$$P(\tilde{s}\tilde{t}) = P[P_{F_i}(s't')] = P_{F_i}[P(s't')] = P(s't'),$$

and

$$P(\tilde{w}) = P[P_{N_j}(s't')] = P_{N_j}[P(s't')] = P(s't'),$$

and thus, $P(\tilde{s}\tilde{t}) = P(\tilde{w})$. According to Remark 1, an event $\sigma \in \Sigma_{db,o}$ belongs to $s't'$ if and only if $\sigma \in \Sigma_{o_i} \cap \Sigma_{o_j}$. Therefore, $P(\tilde{s}\tilde{t}) = P_i(\tilde{s}\tilde{t})$ and $P(\tilde{w}) = P_j(\tilde{w})$, which implies that there exists a trace $st \in L(G)$ of arbitrarily long length after the occurrence of the fault event and a nonfaulty trace $w \in L(G)$, such that $P_{o_i}(st) = P_{o_j}(w)$. Thus, robust diagnosability is violated.

($\Longrightarrow$) Suppose now that L is not robustly diagnosable with respect to P_{o_i}, $i = 1, 2, \ldots, m$, and Σ_f. Thus, there exists a trace $\tilde{s}\tilde{t} \in L(G_{F_i})$, where $\sigma_f \in \tilde{s}$ and $|\tilde{t}| > n, \forall n \in \mathbb{N}$, and $\tilde{w} \in L(G_{N_j})$, such that $P_i(\tilde{s}\tilde{t}) = P_j(\tilde{w})$, which implies that the observable events in $\tilde{s}\tilde{t}$ must all belong to $\Sigma_{o_i} \cap \Sigma_{o_j}$. We will show that $G_{V_{ij}}$ has a cyclic path that satisfies condition (9), and for this purpose, we split the proof in two parts, as follows:

Part I. We show that there exists an arbitrarily long length trace $s't' \in L(G_{V_{ij}})$ such that $P_{F_i}(s't') = \tilde{s}\tilde{t}$ and $P_{N_j}(s't') = \tilde{w}$;
Part II. We prove that there exists a cyclic path cl, associated with trace $s't'$, satisfying condition (9).

In order to prove part I, let us suppose that there exists a state in $G_{V_{ij}}$, $x_{V_{ij}} = (x_{F_i}, x_{N_j})$, reachable from the initial state $x_{0,V_{ij}}$ after the execution of a trace $u \in L(G_{V_{ij}})$, where $P_{F_i}(u)$ is in the prefix-closure of $\tilde{s}\tilde{t}$. Notice that this state $x_{V_{ij}}$ always exists since u can be the empty trace and, in such a case, $x_{V_{ij}} = x_{0,V_{ij}}$. Now, let $\sigma_q \in \Sigma_{R_i}$ be a feasible event of x_{F_i}, such that $P_{F_i}(u)\sigma_q \in \overline{\{\tilde{s}\tilde{t}\}}$, and consider the problem of finding a state of $G_{V_{ij}}$, $\hat{x}_{V_{ij}}$, reachable from $x_{V_{ij}}$, that has σ_q as a feasible event. Two cases are possible:

(a) σ_q is an observable event of $\Sigma_{R_i} \cap \Sigma_{R_j}$;
(b) σ_q is an unobservable event of Σ_{R_i}; notice that in this case σ_q cannot be a renamed event of Σ_{R_j}.

Let us first consider case (a). In this case, σ_q will be a feasible event of $x_{V_{ij}}$ if and only if it is feasible for the corresponding state of G_{N_j}. Since $P_i(\tilde{s}\tilde{t}) = P_j(\tilde{w})$, then σ_q will be feasible for some state of $G_{V_{ij}}$, $\hat{x}_{V_{ij}} = (x_{F_i}, \hat{x}_{N_j})$, after the occurrence of a finite trace from $(\Sigma_{R_j} \setminus \Sigma_{o_j})^*$. Consider now case (b), i.e., $\sigma_q \in \Sigma_{R_i} \setminus \Sigma_{o_i}$. In this case, since self-loops labeled with all events in the set $\Sigma_{R_i} \setminus \Sigma_{o_i}$ are added to each state of G_{N_j} in order to form the parallel composition $G_{V_{ij}} = G_{F_i} \parallel G_{N_j}$, we may conclude that σ_q is already feasible for $x_{V_{ij}} = (x_{F_i}, x_{N_j})$. Therefore, it can be seen that there exists an arbitrarily long trace $s't'$ associated with $\tilde{s}\tilde{t}$ such that $s't' \in P^{-1}_{F_i}(\tilde{s}\tilde{t}) \cap P^{-1}_{N_j}(\tilde{w})$, which implies that $P_{F_i}(s't') = \tilde{s}\tilde{t}$ and $P_{N_j}(s't') = \tilde{w}$.

In order to prove part II, i.e., that there exists a cyclic path cl in $G_{V_{ij}}$ whose first components of its states are faulty states and at least one of the events of the cyclic path belongs to Σ_{R_i}, let us assume, without loss of generality, that $\tilde{s} = P_{F_i}(s')$ and $\tilde{t} = P_{F_i}(t')$. Therefore, t' is also an arbitrarily long trace of $L(G_{V_{ij}})$. Notice that since $G_{V_{ij}}$ is a finite state automaton, t' must be associated with a cyclic path cl of $G_{V_{ij}}$ whose first components are faulty states. Any cyclic path cl in $G_{V_{ij}}$ must satisfy one of the following three cases:

(i) cl is associated with two cyclic paths, one in G_{F_i} and another one in G_{N_j};
(ii) cl is associated with a cyclic path in G_{F_i} only, i.e., with no cyclic path in G_{N_j};
(iii) cl is associated with a cyclic path in G_{N_j} only, i.e., with no cyclic path in G_{F_i}.

If condition (iii) holds true, then all states of cl will have the same first component $x_{F_i} \in X_{F_i}$. Therefore $\nexists\sigma_q \in \Sigma_{R_i}$ such that σ_q is an event of the cyclic path cl, which contradicts part I of the proof. On the other hand, when either condition (i) or (ii) holds true, then, as shown in the above proof of part I, $\exists\sigma_q \in \Sigma_{R_i}$ in the cyclic path cl, which concludes the proof. □

Table 1
Computational complexity of Algorithm 1.

Aut.	Number of states	Number of transitions
G_i	$\|X\|$	$\|X\|\|\Sigma\|$
A_l	2	2
G_{l_i}	$2\|X\|$	$2\|X\|\|\Sigma\|$
G_{F_i}	$2\|X\|$	$2\|X\|\|\Sigma\|$
A_{N_i}	1	$\|\Sigma\| - \|\Sigma_f\|$
G_{N_i}	$\|X\|$	$\|X\|(\|\Sigma\| - \|\Sigma_f\|)$
$G_{V_{ij}}$	$2\|X\|^2$	$2\|X\|^2[2(\|\Sigma\|-\|\Sigma_f\|)+\|\Sigma_f\|]$
Computational complexity	$O(m^2\|X\|^2\|\Sigma\|)$	

4.1. Computational complexity of Algorithm 1

Table 1 shows the maximum number of states and transitions of all automata that must be computed in order to obtain the verifier automaton $G_{V_{ij}}$ according to Algorithm 1 assuming that there are m diagnosis bases, or equivalently, m models G_i for G.

In the first step of Algorithm 1, we build automaton G_i. Since G_i is the same as G except for the renaming of the events in Σ_{uo_i}, the number of states and transitions in G_i are equal to those of G. In the second step of Algorithm 1, we construct automaton G_{F_i}. In order to do so, it is first necessary to build automaton A_ℓ, which has two states, N and Y, and whose transitions are labeled with the fault events, and, after that, we obtain $G_{\ell_i} = G_i \parallel A_\ell$. Notice that the states of G_{ℓ_i} are either (x, N) or (x, Y), where $x \in X$. Therefore, the maximum number of states of G_{ℓ_i} is $2|X|$. Finally, since G_{F_i} is formed by taking the coaccessible part of G_{ℓ_i}, the maximum number of states and transitions of G_{F_i} are equal to

$2|X|$ and $2|X||\Sigma|$, respectively. In step 3, a single state automaton A_{N_i}, whose unique state has a self-loop labeled with all events in $\Sigma_{R_i} \setminus \Sigma_f$ is used to obtain the normal behavior automaton $G_{N_i} = G_i \times A_{N_i}$. Therefore, the maximum number of states and transitions of G_{N_i} are $|X|$ and $|X|(|\Sigma| - |\Sigma_f|)$, respectively. In step 4, verifier $G_{V_{ij}}$ is obtained by performing the parallel composition of G_{F_i} and G_{N_j}, $i \neq j$. Therefore, in the worst case, the number of states and transitions of $G_{V_{ij}}$ are equal to $2|X|^2$ and $2|X|^2[2(|\Sigma| - |\Sigma_f|) + |\Sigma_f|]$, respectively. Finally, the test in step 5 can be done for each verifier by first finding its strongly connected components, which has linear complexity in the number of states plus transitions, and then by examining each state within each strongly connected component, together with the active event set of the state, to determine if the condition in Eq. (9) holds true or not. Thus, step 5 is linear in the size of each $G_{V_{ij}}$. Since a total of $m(m-1)$ verifiers need to be constructed, the overall computational complexity of Algorithm 1 is $O(m^2|X|^2|\Sigma|)$.

Remark 2. Instead of building $m(m-1)$ verifiers $G_{V_{ij}}$, as required by Algorithm 1, we could build m verifiers by performing for each one the parallel composition between G_{F_i} and $G^a_{N_j}, j = 1, 2, \ldots, m, j \neq i$, as proposed in Carvalho, Moreira, and Basilio (2011), where $G^a_{N_j}$ is the augmented automaton obtained from G_{N_j} by adding a dump state and completing the transition function with respect to Σ_o. The computation of all verifiers according to Carvalho et al. (2011) is also polynomial time, having worst-case computational complexity of $O(m^2|X|^m|\Sigma|)$. Since the computational complexity for constructing all verifiers according to Algorithm 1 is $O(m^2|X|^2|\Sigma|)$, when the number of diagnosis bases is greater than 2, the verification algorithm proposed here outperforms that proposed in Carvalho et al. (2011). □

Example 2. Consider again automaton G of Fig. 1(a), where $\Sigma = \{a, b, c, d, e, \sigma_f\}$ and $\Sigma_o = \{a, b, c, d, e\}$. Following the algorithm proposed in Basilio et al. (2012), we find the following diagnosis bases for L:

$$\begin{aligned}\Sigma_{db} = \{&\{a, b, c\}, \{c, d, e\}, \{a, c, d\}, \{a, d, e\}, \{a, b, e\},\\ &\{b, c, e\}, \{a, b, c, d\}, \{a, b, c, e\}, \{a, b, d, e\},\\ &\{a, c, d, e\}, \{b, c, d, e\}, \Sigma_o\}. \end{aligned} \quad (10)$$

Let $\Sigma_{o_i}, i = 1, 2, \ldots, 12$, denote each element of Σ_{db} in the order listed above; for instance $\Sigma_{o_3} = \{a, c, d\}$ and $\Sigma_{o_6} = \{b, c, e\}$. According to Algorithm 1, the verification of the robust diagnosability of L with respect to P_{o_i} and Σ_f for all $\Sigma_{o_i} \in \Sigma_{db}$ is carried out by finding verifiers $G_{V_{ij}}, i, j = 1, 2, \ldots, 12, j \neq i$. In this example we will construct only four verifiers: $G_{V_{1,2}}$, $G_{V_{4,2}}$, $G_{V_{5,6}}$, and $G_{V_{9,11}}$; verifier $G_{V_{1,2}}$ will be used to illustrate the use of Algorithm 1 and the other verifiers will be used to show the action to be taken when the language is not robustly diagnosable.

Let us consider the construction of verifier $G_{V_{1,2}} = G_{F_1} \parallel G_{N_2}$. According to Algorithm 1, we must, initially, construct renamed automata G_1 and G_2, which are depicted in Fig. 3, by renaming the events of G that belong to Σ_{uo_1} and Σ_{uo_2}, respectively. Notice that events d and e are renamed as d_{R_1} and e_{R_1}, in automaton G_1, and event a and b as a_{R_2} and b_{R_2} in G_2. The next step of Algorithm 1 is to build from G_1 the faulty behavior automaton G_{F_1}, whose state transition diagram is depicted in Fig. 4(a). Notice that, as defined in step 1 of Algorithm 1, $\Sigma_{R_1} = R_1(\Sigma) = \{a, b, c, d_{R_1}, e_{R_1}, \sigma_f\}$, and thus, according to Eq. (8), $\Sigma_{db,o} = \cup_{i=1}^{12} \Sigma_{o_i} = \{a, b, c, d, e\}$ and $\Sigma_{F_1} = \{a, b, c, d, e, d_{R_1}, e_{R_1}, \sigma_f\}$. Continuing to run Algorithm 1, we must now obtain from G_2, the normal behavior automaton G_{N_2}, which is shown in Fig. 4(b). Notice that $\Sigma_{N_2} = (\Sigma_2 \cup \Sigma_{db,o}) \setminus \{\sigma_f\} = \{a, b, c, d, e, a_{R_2}, b_{R_2}\}$. Finally, we build verifier automaton $G_{V_{1,2}}$, which is computed by performing the parallel composition of $G_{F_1} \parallel G_{N_2}$. From the state transition diagram of $G_{V_{1,2}}$, shown in Fig. 5, we can see that $G_{V_{1,2}}$ has no cyclic path that satisfies condition (9).

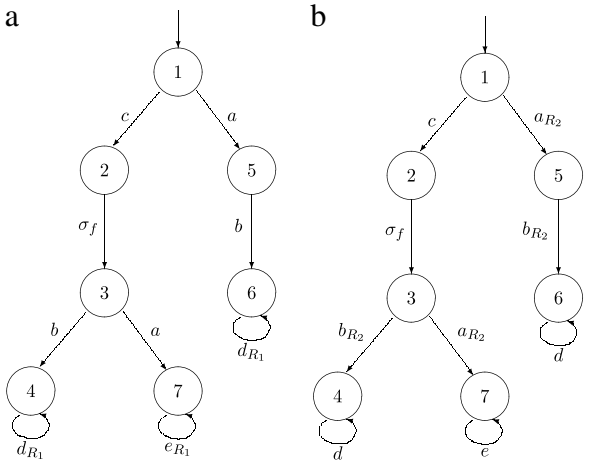

Fig. 3. Automaton G_1 (a); Automaton G_2 (b).

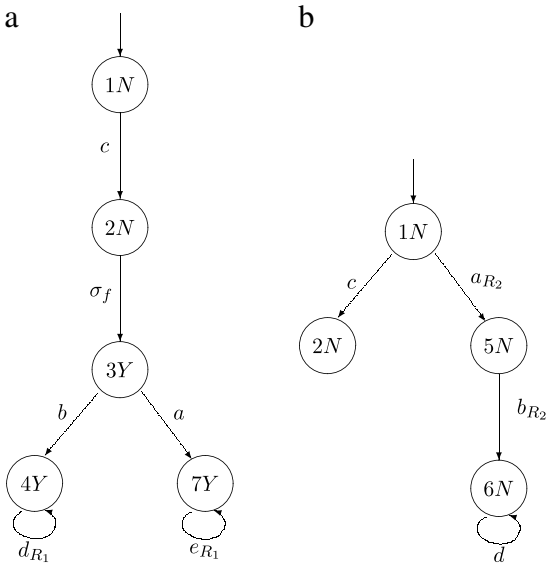

Fig. 4. Automaton G_{F_1} (a) and automaton G_{N_2} (b).

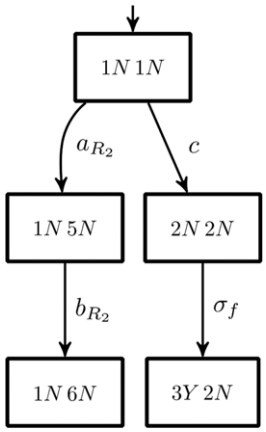

Fig. 5. Verifier automaton $G_{V_{1,2}}$.

Fig. 6(a)–(c) show the state transition diagrams of verifiers $G_{V_{4,2}}$, $G_{V_{5,6}}$ and $G_{V_{9,11}}$, respectively. Notice that they all have cyclic paths that satisfy condition (9), and therefore L is not robust

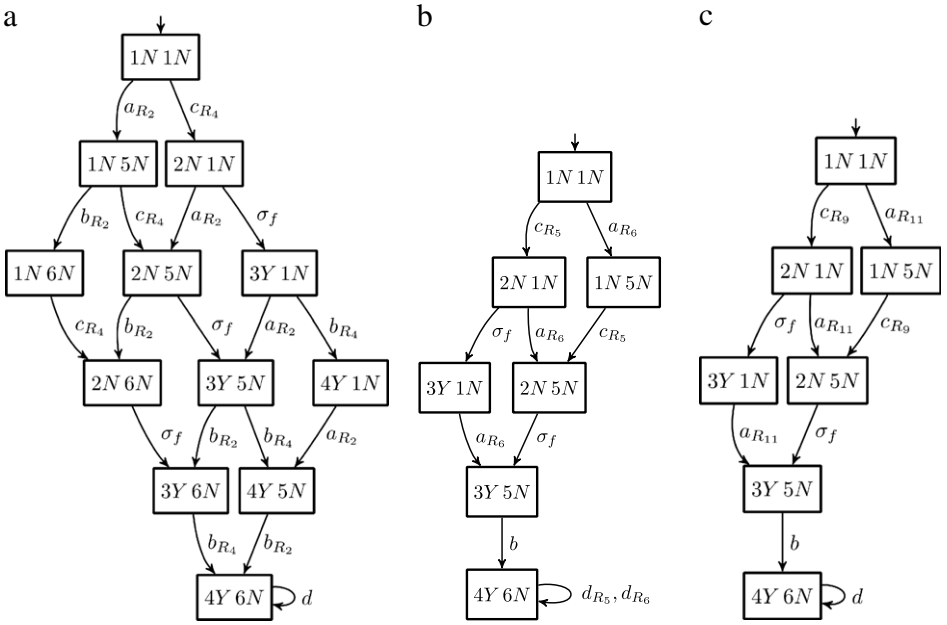

Fig. 6. Verifier automata $G_{V_{4,2}}$ (a), $G_{V_{5,6}}$ (b) and $G_{V_{9,11}}$ (c).

diagnosable with respect to $P_{o_i}, i = 1, 2, \ldots, m$ and Σ_f. As a consequence, there exist two traces, $s_Y = c\sigma_f bd^n$ and $s_N = abd^n$, whose projections over $\{a, b, e\}^*$ and $\{b, c, e\}^*$ are both equal to b, where the faulty trace (the trace that contains event σ_f) is arbitrarily long after the fault occurrence, and the normal trace (the trace that has no faulty event) can be made either finite length (when $n = 0$) or be arbitrarily long (for $n > 1$). In other words, it is not possible to state that either trace s_Y occurred and sensors c and d failed or trace s_N occurred and sensors a and d failed. It is worth remarking that these are the only verifiers that have cyclic paths that satisfy condition (9). Notice that verifier $G_{V_{4,2}}$ has the cyclic path $(4Y6N, d, 4Y6N)$, which implies that we can find two traces, one arbitrarily long faulty trace $s_{F_4} = c_{R_4}\sigma_f b_{R_4} d^n, n \geq 1$, in $\mathcal{L}(G_{F_4})$ and a normal trace $s_{N_2} = a_{R_2} b_{R_2} d^m, m \in \mathbb{N}$, in $\mathcal{L}(G_{N_2})$. After applying the inverse renaming function (7) to these two traces, we obtain traces $s_4 = c\sigma_f bd^n$ and $w_2 = abd^m$, for which $P_{o_4}(s_4) = P_{o_2}(w_2) = d^n$, which violates the robust diagnosability condition given in Definition 3. In addition, an inspection of verifier $G_{V_{5,6}}$ reveals that cyclic paths $(4Y6N; d_{R_5}; 4Y6N)$ and $(4Y6N; d_{R_6}; 4Y6N)$ are formed with renamed events, which is due to the fact that event d is an unobservable event for both diagnosis bases Σ_{o_5} and Σ_{o_6}.

5. Online implementation of robust diagnosers

A robust diagnoser that copes with permanent loss of observation of the events in any of the sets Σ_{uo_i}, or simply, robust diagnoser, is a diagnoser that is able to diagnose a fault and satisfies the conditions imposed by Definition 3. With the view to performing online diagnosis we could deploy the union diagnoser proposed in Lima et al. (2010). However, in order to overcome the potential state space explosion of union diagnosers, we propose here a different scheme in which all partial diagnosers run in parallel. Starting at their initial states and after the occurrence of the first observable event, those partial diagnosers whose current active event sets contain the event that has just occurred move to their respective next state; all the other partial diagnosers are discarded since they are unable to process the current observation. The above process is repeated after each observable event. The definition of robust diagnosability guarantees that after a bounded number of events following the fault, all remaining partial diagnosers that have not been discarded will have reached a certain state, and thus will agree on the diagnosis of the fault.

Example 3. Let us consider again the robust diagnosis problem addressed in Example 2. Notice that, in verifier automata $G_{V_{4,2}}$ (a), $G_{V_{5,6}}$ (b) and $G_{V_{9,11}}$ (c) the paths that lead to the N components in the cyclic path that lead to violation of robust diagnosability are those formed with the following diagnosis bases:

$$\Sigma_N = \{\{b, c, d, e\}, \{c, d, e\}, \{b, c, e\}\}.$$

If we remove the event sets of Σ_N from Σ_{db} given in Eq. (10), then L becomes robustly diagnosable with respect to $P_{o_i} : \Sigma^* \to \Sigma_{o_i}^*$, and $\Sigma_f = \{\sigma_f\}$, where $\Sigma_{o_i} \in \Sigma_{db,rob}$, and

$$\Sigma_{db,rob} = \Sigma_{db} \setminus \Sigma_N = \{\{a, b, c\}, \{a, c, d\}, \{a, d, e\}, \{a, b, e\}, \{a, b, c, d\}, \{a, b, c, e\}, \{a, b, d, e\}, \{a, c, d, e\}, \Sigma_o\},$$

or equivalently, with respect to permanent loss of observation of the events of all sets $\Sigma_{uo_i} \in \Sigma_{rob}, i = 1, 2, \ldots, m$, where

$$\Sigma_{rob} = \{\{d, e\}, \{b, e\}, \{b, c\}, \{c, d\}, \{e\}, \{d\}, \{c\}, \{b\}, \emptyset\}.$$

The robust diagnoser operates by running independently seven partial diagnosers, $G_{d_i}, i = 1, 2, \ldots, 7$, whose corresponding observable event sets $\Sigma_{o_i}, i = 1, 2, \ldots, 7$, are the first seven event sets of $\Sigma_{db,rob}$ and the last one is the centralized diagnoser, i.e., the one whose observable event set is Σ_o. To illustrate the robust diagnnoser operation, assume that trace $s_Y = c\sigma_f ae^n, n \in \mathbb{N}$, occurs and that event c fails to be permanently observed prior to its first occurrence. Since $\{c\} \in \Sigma_{rob}$, then the robust diagnoser must be able to diagnose the occurrence of σ_f in spite of the loss of observation of c. Notice that the first event occurrence to be recognized by the robust diagnoser is a. As seen in the first row of Table 2, all partial diagnosers move to another state; notice that $G_{d_1}, G_{d_5}, G_{d_6}$ and G_d give wrong information regarding the fault occurrence whereas the others are in uncertain states. After the first occurrence of event e only diagnosers G_{d_3} and G_{d_4} move to their next states; all the others get stuck and should be discarded,

Table 2
Operation of the robust diagnoser after the occurrence of trace $s_Y = c\sigma_f a e^n (n \in \mathbb{N})$ assuming permanent loss of observations of event c.

Event	G_{d_1}	G_{d_2}	G_{d_3}	G_{d_4}	G_{d_5}	G_{d_6}	G_{d_7}	G_d
a	✓ $5N$	✓ $5N, 6N$	✓ $7Y, 5N, 6N$	✓ $7Y, 5N$	✓ $5N$	✓ $5N$	✓ $5N, 6N$	✓ $5N$
e	×	×	✓ $7Y$	✓ $7Y$	×	×	×	×
e	×	×	✓ $7Y$	×	×	×	×	×

as shown in the second line of Table 2. After the next occurrences of event e, the only diagnoser that continues to run is G_{d_3}, and, since it loops in a certain state, we may say that the robust diagnoser succeeded in diagnosing the fault occurrence.

6. Conclusion

We have considered in this paper the problem of making fault diagnosis systems resilient to potential loss of observations due to unknown and permanent sensor failures when a partially-observed discrete-event system is turned on. Robustness to potential loss of observations is achieved by exploiting the redundancy that may exist in the sensors, as captured by the notions of diagnosis bases and redundant event sets. We present a methodology for testing robust diagnosability based on the use of verifiers. We also propose the use of a set of partial diagnosers to perform online diagnosis, where each partial diagnoser is built assuming the loss of a set of redundant events. The concepts and techniques introduced in this paper contribute to the development of fault-tolerant diagnostic and control architectures.

References

Athanasopoulou, E., Lingxi, L., & Hadjicostis, C. (2010). Maximum likelihood failure diagnosis in finite state machines under unreliable observations. *IEEE Transactions on Automatic Control*, *55*(3), 579–593.

Basilio, J.C., & Lafortune, S. (2009). Robust codiagnosability of discrete event systems. In *Proc. of the American control conference*. St. Louis, Missouri (pp. 2202–2209).

Basilio, J. C., Lima, S. T. S., Lafortune, S., & Moreira, M. V. (2012). Computation of minimal event bases that ensure diagnosability. *Discrete Event Dynamic Systems: Theory and Applications*, *22*(3), 249–292.

Boel, R.K., & van Schuppen, J.H. (2002). Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers. In *Proc. of the 2002 international workshop on discrete event systems*. Zaragoza, Spain (pp. 175–181).

Carvalho, L.K., Basilio, J.C., & Moreira, M.V. (2010). Robust diagnosability of discrete event systems subject to intermittent sensor failures. In *Proc. of the 10th international workshop on discrete event systems*. Berlin, Germany (pp. 94–99).

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

Carvalho, L.K., Moreira, M.V., & Basilio, J.C. (2011). Generalized robust diagnosability of discrete event systems. In *Proc. of 18th IFAC world congress*. Milan, Italy (pp. 8737–8742).

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

Contant, O., Lafortune, S., & Teneketzis, D. (2006). Diagnosability of discrete event systems with modular structure. *Discrete Event Dynamic Systems: Theory And Applications*, *16*(1), 9–37.

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

Genc, S. (2008). Formal methods for intrusion detection of windows NT attacks. In *Proc. of 3rd annual symposium on information assurance & 11th annual NYS cyber security conference, vol. 1* (pp. 71–79).

Jéron, T., Marchand, H., Pinchinat, S., & Cordier, M.-O. (2006). Supervision patterns in discrete event systems diagnosis. In *Proc. of 8th international workshop on discrete event systems*. Ann Arbor, MI (pp. 262–268).

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

Kumar, R., & Takai, S. (2009). Inference-based ambiguity management in decentralized decision-making: Decentralized diagnosis of discrete-event systems. *IEEE Transactions on Automation Science and Engineering*, *6*(3), 479–491.

Lima, S.T.S., Basilio, J.C., Lafortune, S., & Moreira, M.V. (2010). Robust diagnosability of discrete event systems subject to permanent sensor failures. In *Proc. of the 10th international workshop on discrete event systems*. Berlin, Germany (pp. 100–107).

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

Lunze, J., & Schröder, J. (2004). Sensor and actuator fault diagnosis of systems with discrete inputs and outputs. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, *34*(2), 1096–1107.

Moreira, M.V., Jesus, T.C., & Basilio, J.C. (2010). Polynomial time verification of decentralized diagnosability of discrete event systems. In *Proc. of the 2010 American control conference* (pp. 3353–3358).

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

Pencolé, Y., & Cordier, M. O. (2005). A formal framework for the decentralized diagnosis of large scale discrete event systems and its applications to telecommunication networks. *Artificial Intelligence*, *164*(1-2), 121–170.

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

Rohloff, K.R. (2005). Sensor failure tolerant supervisory control. In *Proc. of joint 2005 European control conference and 44th IEEE conference on decision and control*. Seville, Spain (pp. 3493–3498).

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

Sanchez, A. M., & Montoya, F. J. (2006). Safe supervisory control under observability failure. *Discrete Event Dynamic Systems: Theory and Applications*, *16*(4), 493–525.

Takai, S. (2010). Robust failure diagnosis of partially observed discrete event systems. In *Preprints of 10th international workshop on discrete event systems*. Berlin, Germany (pp. 215–220).

Takai, S. (2012). Verification of robust diagnosability for partially observed discrete event systems. *Automatica*, *48*(8), 1913–1919.

Thorsley, D., & Teneketzis, D. (2005). Diagnosability of stochastic discrete-event systems. *IEEE Transactions on Automatic Control*, *50*(4), 476–492.

Thorsley, D., Yoo, T.-S., & Garcia, H. (2008). Diagnosability of stochastic discrete-event systems under unreliable observations. In *Proc. of the 2008 American control conference*. Seattle, WA (pp. 1158–1365).

Tripakis, S. (2002). Fault diagnosis for timed automata. In W. Damm, & E.-R. Olderog (Eds.), *Lecture notes in computer sciences*: *vol. 2469. Formal techniques in real time and fault tolerant systems* (pp. 205–221). Springer-Verlag.

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

Ye, L., Dague, P., & Yan, Y. (2009). An incremental approach for pattern diagnosability in distributed discrete event systems. In *Proc. of 21st international conference on tools with artificial intelligence*. Newark, NJ (pp. 123–130).

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

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

Lilian Kawakami Carvalho was born on March, 11, 1979 in São Paulo, Brazil. She received the Electronic Engineer degree, the M.Sc. degree and the D. Sc. degree in Control from the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, in 2003, 2005 and 2011, respectively. Since 2011, she has been an Associate Professor at the Department of Electrical Engineering at the Federal University of Rio de Janeiro. Her main interests are fault diagnosis of discrete-event systems, supervisory control applied to mobile robotics, and the development of control laboratory techniques.

Marcos Vicente Moreira was born on May, 11, 1976 in Rio de Janeiro, Brazil. He received the Electrical Engineer degree, the M.Sc. degree and the D. Sc. degree in Control from the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, in 2000, 2002 and 2006, respectively. Since 2007, he has been an Associate Professor at the Department of Electrical Engineering at the Federal University of Rio de Janeiro. His main interests are multivariable control, robust control, discrete-event systems and the development of control laboratory techniques.

João Carlos Basilio was born on March 15, 1962 in Juiz de Fora, Brazil. He received the Electrical Engineering degree in 1986 from the Federal University of Juiz de Fora, Juiz de Fora, Brazil, the M.Sc. degree in Control from the Military Institute of Engineering, Rio de Janeiro, Brazil, in 1989, and the Ph.D. degree in Control from Oxford University, Oxford, UK, in 1995. He began his career in 1990 as an Assistant Lecturer at the Department of Electrical Engineering of the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, and, since 2007, has been a Senior Associate Professor in Control at the same department. He served as the Academic Chair for the graduation course in Control and Automation from January, 2005, to December, 2006, and as the Chair for the Electrical Engineering Post-graduation Program from January, 2008, to February, 2009. From September, 2007, to December, 2008, he spent a sabbatical leave at the University of Michigan, Ann Arbor. His is currently interested in discrete-event systems and in the development of control and automation laboratories and new teaching techniques. Dr. Basilio is the recipient of the Correia Lima Medal.

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