Intermittent fault diagnosability of discrete event systems: an overview of automaton-based approaches

Abderraouf Boussif, Mohamed Ghazel, João Carlos Basilio

To cite this article:
Boussif, A., Ghazel, M. Basilio, J.C. Intermittent fault diagnosability of discrete event systems: an overview of automaton-based approaches. Discrete Event Dyn Syst (2020).
https://doi.org/10.1007/s10626-020-00324-y

Intermittent fault diagnosability of discrete event systems: an overview of automaton-based approaches

Abderraouf Boussif · Mohamed Ghazel · João Carlos Basilio

the date of receipt and acceptance should be inserted later

Abstract Real life experience has shown that intermittent faults are among the most challenging kinds of faults to detect and isolate, being present in the majority of production systems. Such a concern has made intermittent fault an active area of research in both discrete event and continuous-variable dynamic systems. In this paper, we present a review of the state-of-the art of intermittent fault diagnosis of discrete event systems modeled by finite state automata. To this end, we revisit the main definitions of diagnosability of intermittent faults, and present comparisons between them, consider verification and analysis techniques, and discuss available complexity results. Examples are used throughout the paper to illustrate the reviewed concepts and verification algorithms. We also look ahead, by suggesting some perspectives for future research.

Keywords Discrete event systems, automata, intermittent fault, diagnosability, diagnosis

1 Introduction

Fault diagnosis in dynamic systems is a crucial and challenging task to ensure reliability, safety, and correct operation of production systems. In this context, and to fulfill such requirements, the development of effective monitoring techniques

Abderraouf Boussif, E-mail: abderraouf.boussif@railenium.eu
Institut de Recherche Technologique Railenium, F-59300 Famars, France

Mohamed Ghazel*, E-mail: mohamed.ghazel@ifsttar.fr
COSYS-ESTAS, Univ Gustave Eiffel, IFSTTAR, Univ Lille, F-59650 Villeneuve d'Ascq, France

João Carlos Basilio**, E-mail: basilio@dee.ufrj.br
Department of Electrical Engineering, Universidade Federal do Rio de Janeiro, 21949-900, Rio de Janeiro, Brazil.

* The work of M. Ghazel was supported by ELSAT2020 project. ELSAT2020 is co-financed by the European Union with the European Regional development Fund, the French state and the Hauts de France Region Council.
** The work of J. C. Basilio was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Finance Code 001, and the Brazilian Research Council (CNPq), grant number 309652/2017-0.

becomes a concern that must be addressed. In particular, having efficient tools for monitoring and diagnosing fault occurrences is of paramount interest since such actions prevent, or at least mitigate, failure-related disturbances effects.

Fault diagnosis involves the following aspects: *(i)* detection of fault occurrences; *(ii)* isolation of the actual fault from other possible fault candidates, and; *(iii)* identification of the related damage caused to the system. In discrete event systems (DES) (Cassandras and Lafortune, 2008), fault diagnosis is often discussed through two main issues: online diagnosis and diagnosability analysis (Lin, 1994; Sampath et al., 1995, 1996a). Online diagnosis consists in inferring the occurrence of predetermined faults from the observed behavior of the system, while diagnosability is associated with the capacity of the system that performs the fault diagnosis — usually referred to as diagnoser — to provide a precise verdict as far as the fault occurrence is concerned. Thus, system diagnosability analysis consists in determining whether or not every predetermined failure can be detected and identified accurately within a finite delay after its occurrence (Sampath et al., 1995).

A fault is any deviation of the system from its normal or intended behavior. In the DES framework, faults are basically depicted as *unobservable/silent*, *indistinguishable* and *uncontrollable* events (or states). Moreover, faults can be classified on the basis of their individual behavior into three types (Sharma et al., 2015b; Zaytoon and Lafortune, 2013)

1. *Permanent faults:* when the fault occurs and does not disappear (*i.e.*, the system remains in fault states) unless removed by some external intervention. Typically, a permanent fault can be caused by subsystem failures, physical damage or design error. The terminology *failure* is often used to refer to permanent faults;
2. *Drift-like faults:* when the fault varies gradually and slowly develops into a large value. In DES, these faults can be seen as faults which may occur within an incremental frequency, or may evolve into permanent ones. Generally, diagnosing such faults is more involving than the permanent failures since they often evolve slowly and their effects can be confused with noise and model uncertainty;
3. *Intermittent faults:* which correspond to the case when the fault occurs and then suddenly disappears, and this process continues to repeat in either periodic or non-periodic manner, making the system switch between normal and faulty behaviors (Isermann, 2006).

From the diagnosis point of view, it is important to distinguish between these fault types, especially between permanent and intermittent faults (Deng et al., 2014a). Intermittent faults can be spontaneously recovered by the occurrence of uncontrollable and unobservable reset events; therefore, the system oscillates between normal and faulty behavior. Permanent faults, on the other hand, may be associated with recovery events (repair/replacement) which are controllable and observable (Huang, 2003). It is worth remarking that, although in most part of the literature regarding model-based fault diagnosis of DESs, faults are assumed to be permanent, practical evidences have shown that intermittent faults are omnipresent and are among the most challenging kinds of faults to detect and isolate (Fromherz et al., 2004). In this regard, frequent occurrences of intermittent faults may bring serious troubles and result in high safety risk, which may reduce

the competitiveness and damage the reputation of the company. In addition, intermittent faults can induce overhead maintenance costs for companies due to several related problems, such as "Can Not Duplicate (CND)", "No Fault Found (NFF)", "False Alarms (FAs)", *etc.* (Sorensen et al., 1994); in particular, NFF costs could be significantly high due to the need of extra tests to identify such failures (Söderholm, 2007).

In order to show the significance of intermittent faults and their impact in industry, in particular, their financial impacts, we enumerate a set of indicators gathered from the literature and from industrial reports, as follows:

- In the late 1960s, surveys provided by Hardie and Suhocki (1967) and Ball and Hardie (1969) indicate that intermittent faults comprise over 30% of predetermined faults/errors and about 90% of field failures in computer systems.
- Between 80% and 90% of failures in sequential circuits are caused by intermittent faults (Roberts, 1989). A similar conclusion has been reached in wireless sensors networks by Banerjee and Khilar (2010).
- In 2001, over $10 million have been spent by F-16 plane customers in order to replace parts that were tested as NFF at the shop level (Steadman et al., 2002). In addition, NFF observations reported by commercial airlines and military repair depots have been found to be as high as 50-60%.
- The thick film integrated ignition module in Ford cars in the 1980 models led to a lawsuit and a settlement by Ford Motors Company due to intermittent faults, particularly NFFs (Maul et al., 2001).
- Recently, a survey among 80 aerospace organizations (Syed et al., 2013) ranked intermittent faults as the main perceived cause of the NFF problem and the highest cost source in terms of aerospace maintenance;
- In digital electronic cruise control modules (CCM) used in automobiles, intermittent faults were the justification for the fact that 96% of the components that were returned to the vehicle manufacturer due to customer complaints actually operated properly (Kimseng et al., 1999);
- In 1997, the Air Transport Association (ATA) estimated at $20M the annual NFF costs for an airline operating 200 aircrafts; or $100,000 per aircraft every year (Erkoyuncu et al., 2016);
- In 2005, a study by WDS mobile company found that NFFs due to intermittent faults account for about 63% of the mobile phones that were returned to the manufacturer, costing the industry $4.5 billion a year (Overton, 2006).
- The cost of exchange of F-16 avionics boxes due to intermittent faults was estimated at $20,000,000 (Steadman et al., 2005, 2008).

Overall, intermittent faults are a general phenomenon that affects industrial systems ranging from small components to large complex modules (Shen et al., 2016). Therefore, in several domains such as digital and electronic systems (Chang and McCluskey, 1997; Gracia et al., 2008), aerospace industry (Salvatore et al., 2003), aircraft systems (Anderson and Aylward, 1993; Yang et al., 2012), modern industrial and chemical processes (Madden and Nolan, 1999; Yan et al., 2015), transportation systems (Aydin et al., 2013), machine driven systems (Ismaeel and Bhatnagar, 1997; Kim, 2009) and computer systems (Hsu and Hsu, 1991), there is a need to deeply address issues related to intermittent fault diagnosis.

Generally, intermittent faults are characterized by three parameters, as follows:

1. *Duration:* which represents the time during which the fault is active at each occurrence;
2. *Pseudo-period:* which is defined as the mean time between two consecutive fault occurrences (or detections); being therefore, the average delay separating successive faults inside a sliding window. It is worth noticing that it would not be appropriate to define the average interval as a period, due to the asynchronous and random nature of faults;
3. *Number of fault detections:* which represents how many times the fault is detected.

In the literature, we can find several definitions of intermittent faults: Sorensen et al. (1994) defines intermittent faults as "*any temporary deviation from the nominal operating conditions of a circuit or device*". In Syed et al. (2013), intermittent faults are defined as a temporary malfunction of a device. For Pan et al. (2012), an intermittent fault is "*a hardware error which occurs frequently and irregularly for a period of time*". According to the IEEE standard (Sharma et al., 2015a), intermittent faults are defined as "*failures of an item for a limited period of time, following which the item recovers its ability to perform its required function without being subjected to any external corrective action. Moreover, such failures are often recurrent*". In the context of DES, intermittent faults are defined as "*faults which often occur intermittently, and can be seen as fault events followed, later on, by the corresponding reset events, (possibly) followed by new occurrences of fault events, and so forth*" (Contant et al., 2004). Another similar definition is given in Deng et al. (2014a), where intermittent faults are presented as faults that can at some point automatically reset once they have occurred.

This paper aims to provide a comprehensive and general review of the literature regarding intermittent fault diagnosis of DES modeled by finite state automata. We assume that the reader is familiar with DES theory and its classically-related modeling formalism; the reader is referred to Cassandras and Lafortune (2008) for a background on the DES theory. Nevertheless, we also present a brief account of the main contributions on intermittent fault diagnosis using other DES frameworks, such as supervision pattern, temporal logic, discrimination between intermittent and permanent faults, and fault free models, and using different modeling formalism, such as Petri nets and stochastic automata.

The paper is organized as follows: we introduce the main notions and notations regarding DES and the intermittent fault modeling in Section 2; in Section 3, we synthesize the properties of diagnosability reported in the literature and review the main definitions of intermittent fault diagnosability; in Section 4, we review existing techniques to verify the various notions of intermittent fault diagnosability; in Section 5, we briefly discuss how intermittent fault diagnosability has been addressed using other formalisms, such as Petri nets and temporal logic specifications; in Section 6, we point towards the future by discussing some open problems and suggesting some research topics; finally, we draw some conclusions in Section 7.

2 Preliminaries

2.1 Discrete event system modeling

In this paper, the basic concepts and the main contributions in intermittent fault diagnosis of DES are based on finite state automaton (FSA) model; although, as will be seen in Section 5, other approaches to address this problem are possible.

An FSA is defined by the four-tuple $G = (X, \Sigma, \delta, x_0)$, where X is a finite set of states, Σ is a finite set of events, $\delta : X \times \Sigma \to 2^X$ is the partial transition function, and $x_0 \in X$ is the initial state. A triple $(x, \sigma, x') \in X \times \Sigma \times X$ is called a *transition* if $x' \in \delta(x, \sigma)$. The system behavior is described by the prefix-closed language $L \subseteq \Sigma^*$ generated by G, where Σ^* denotes the *Kleene-closure* of Σ. We will partition the set of events Σ as $\Sigma = \Sigma_o \dot{\cup} \Sigma_u$, where Σ_o and Σ_u denote the set of observable and unobservable events, respectively. We say that an *event-sequence* $s = \sigma_1\sigma_2\cdots\sigma_n$, with $\sigma_i \in \Sigma$, is said to be *associated* with a *state-sequence* $\pi = x_1x_2\cdots x_{n+1}$, if $\forall i$ such that $0 < i \leq n$, $x_{i+1} \in \delta(x_i, \sigma_i)$. The partial transition function δ can be extended to event-sequences, *i.e.*, $x_{n+1} \in \delta(x_1, s)$. We denote by σ_i (resp. π_i) the i-th event (resp. state) in s (resp. π). We write $|s|$ to denote the length of s, *i.e.*, the number of events in s. The post-language of L after s is $L/s := \{t \in \Sigma^* \,|st \in L\}$. Notation $s \leq s'$ indicates that s is a prefix of s'.

To capture the observed behavior of the model, we define the *projection mapping* (Lin and Wonham, 1988) $P : \Sigma^* \to \Sigma_o^*$ in the usual manner: $P(\sigma) = \sigma$ for $\sigma \in \Sigma_o$; $P(\sigma) = \epsilon$ for $\sigma \in \Sigma_u$, and $P(s\sigma) = P(s)P(\sigma)$, where $s \in \Sigma^*$, $\sigma \in \Sigma$. When applied to a language L, the projection mapping can be extended as follows: $P : \Sigma^* \to \Sigma_o^*$, where $P(L) = \{t \in \Sigma_o^* | (\exists s \in L)[P(s) = t]\}$. The inverse projection operation P_L^{-1} is defined by $P_L^{-1}(y) = \{s \in L | P(s) = y\}$.

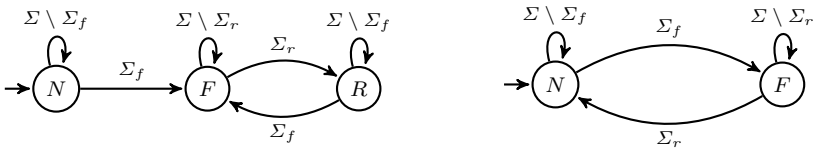
2.2 Modeling intermittent faults

Regarding how the system status evolves after the occurrence of intermittent faults, two modeling settings can be distinguished: recovery and normalization settings.

2.2.1 Recovery setting

In this setting, a distinction is made between the states reached by faulty-free sequences and the states reached by sequences that contain at least one fault event followed later on by its corresponding reset (or recovery) event (Contant et al., 2004; Contant, 2005; Contant et al., 2002; Boussif and Ghazel, 2016a; Carvalho et al., 2010, 2013).

The occurrence of an intermittent fault switches the system from a normal status to a faulty one, after which the system is switched to a recovered status upon the occurrence of the corresponding reset event. Although, such a recovered status is regarded as safe, it is different from the normal status, in the sense that the system never goes back to a normal status once a fault has occurred. In order to capture the changes in the system status, the so-called label automaton $\Omega = (\{N, F, R\}, \Sigma, \delta_\Omega, N)$ (Boussif and Ghazel, 2016a; Jéron et al., 2006; Carvalho et al., 2012; Fabre et al., 2016, 2018) shown in Figure 1a is used. It is clear from the



(a) Recovery modeling. **(b)** Normalization modeling.

Fig. 1 Label automaton Ω for recovery and normalization settings.

figure that Ω translates the system status according to the occurrence of different event types. When the label automaton Ω is in state N (N stands for the normal status), the system is running in its normal behavior, which indicates that no event of Σ_f has occurred yet. However, when a fault event occurs, Ω definitely leaves the normal status and moves to state F (F stands for the faulty status) and remains in this state as long as no reset event occurs. Once the fault is recovered due to the occurrence of a reset event, Ω switches to state R (R stands for the recovered status), where it remains as long as no fault occurs. Since we are dealing with intermittent faults, the system can execute a fault event again. In this case, the label automaton Ω switches back to state F and so on. It is worth noticing that, if a reset event $\sigma_r \in \Sigma_r$ occurs prior to any $\sigma_f \in \Sigma_f$ occurrence, as can be seen from the label automaton Ω, the system will remain in normal status (state N in Ω).

2.2.2 Normalization setting

In the normalization setting, a fault event makes the system status move from normal to faulty, whereas its corresponding reset event moves the system status back to normal, and so, there exist only two possible states for the system since no distinction is made between the normal states reached by fault-free event sequences and those reached by event sequences that have a recovered fault. The reset events in such a context are called normalization events and, thus, the change from faulty to normal status is called normalization. Such a modeling strategy has been used in Biswas (2012), Jiang et al. (2003), Boussif et al. (2016) and Boussif and Ghazel (2019).

Figure 1b shows the label automaton $\Omega = (\{N, F\}, \Sigma, \delta_\Omega, N)$ that is used in the normalization setting. Notice that, when Ω is in state N, the system is running in its normal behavior, indicating that either no fault event of Σ_f has occurred yet or, if some fault event has occurred, it has been normalized, due to the occurrence of a reset (normalizing) event. On the other hand, when a fault event occurs, Ω moves to state F and remains there as long as the system is in its faulty behavior (*i.e.*, no new occurrences of the normalization event). When dealing with intermittent faults, the system may switch between these two state types indefinitely.

3 The different notions of diagnosability of intermittent faults

Broadly speaking, the notions of diagnosability reported in the literature can be divided into two classes: fault counting, and fault detection and identification.

These two categories will be discussed in the following sections. For the sake of clarity and without loss of generality[1], we make the following assumption:
A1. $\Sigma_f = \{\sigma_f\}$ and $\Sigma_r = \{\sigma_r\}$, *i.e.*, there exist exactly one single fault event and one single reset event.

3.1 Fault count-based definitions of intermittent fault diagnosability

The definition of diagnosability of DES introduced in Sampath et al. (1995) is related to "the ability to infer, from the observed behavior of the system, the occurrence of faults". Such a property characterizes single time detection capability, which is suitable for dealing with permanent faults. However, no information regarding multiple occurrences of the same fault can be obtained using Sampath's approach.

Intermittent faults, on the other hand, occur repeatedly. Thus, it may be of interest to have a formalism that not only allows for determining fault occurrences, but also counts its occurrences. Such concepts have been firstly introduced in Jiang et al. (2003), in a state-based scheme within a normalization setting, and then investigated later on in Jiang and Kumar (2006), Yoo and Garcia (2009), Yoo and Garcia (2004), Zhou and Kumar (2009), Yoo and Garcia (2008), and Garcia and Yoo (2005).

Let us denote as $N_s^F \in \mathbb{N}^+$ the number of fault events in a given event-sequence $s \in \Sigma^*$. We now present the different notions of diagnosability for repeated faults.

Definition 1 (κ-diagnosability (Jiang et al., 2003; Yoo and Garcia, 2004)) Given a fixed $\kappa \in \mathbb{N}^\star$, a prefix-closed live language L is said to be κ-diagnosable w.r.t. P and Σ_f if the following holds true:

$$(\exists n_\kappa \in \mathbb{N})(\forall s \in L, N_s^F \geq \kappa)(\forall t \in L/s)\ (|t| \geq n_\kappa) \Rightarrow [(\forall \omega \in P_L^{-1}(P(st))(N_\omega^F \geq \kappa)]$$

Notice that, according to Definition 1, a language is κ-diagnosable if for any event-sequence s containing at least κ faulty events, for every sufficiently long continuation t of s, and for every event-sequence ω indistinguishable from st, *i.e.*, $P(st) = P(\omega)$, then ω must also contain at least κ faulty events.

Property 1 (Jiang et al., 2003) *κ-diagnosability is not monotone.* □

According to Property 1, κ-diagnosability does not imply $(\kappa-1)$-diagnosability. In addition, it is straightforward from the definition of κ-diagnosability that $(\kappa-1)$-diagnosability does not ensure κ-diagnosability (for $\kappa \geq 2$).

The lack of monotonicity of κ-diagnosability motivates a stronger notion of diagnosability, which is called $[1,\kappa]$-diagnosability.

Definition 2 ($[1,\kappa]$-diagnosability (Jiang et al., 2003; Yoo and Garcia, 2004)) Given a fixed $\kappa \in \mathbb{N}^\star$, a prefix-closed live language L is said to be $[1,\kappa]$-diagnosable w.r.t. P and Σ_f if the following holds true:

$$\begin{aligned}&(\exists n \in \mathbb{N})(\forall j, 1 \leq j \leq \kappa)(\forall s \in L, N_s^F \geq j)(\forall t \in L/s)(|t| \geq n)\\ &\Rightarrow [(\forall \omega \in P_L^{-1}(P(st))(N_\omega^F \geq j)]\end{aligned}$$

[1] As shown in Santoro et al. (2017), the case of multiple intermittent faults (and consequently recoveries) can be addressed by considering each fault type separately and assuming the other fault types as ordinary unobservable events.

Definition 2 states that a language is $[1,\kappa]$-diagnosable if it is κ-diagnosable for every j, $1 \leq j \leq \kappa$. The delay bound n that ensures $[1,\kappa]$-diagnosability is the maximal delay bound that ensures κ-diagnosability for $1 \leq j \leq \kappa$, *i.e.*, $n = \max_{j=,\ldots,\kappa} n_j$. The following property relates the definitions of κ-diagnosability, $[1,\kappa]$-diagnosability and the definition of diagnosability introduced in Sampath et al. (1995).

Property 2 *If $\kappa = 1$, then $[1,\kappa]$- and κ-diagnosability properties are equivalent to the diagnosability definition introduced in Sampath et al. (1995).* □

Notice that $[1,\kappa]$-diagnosability allows for determining the first κ occurrences of faulty events within a finite delay. Therefore, in order to determine *any* number of fault occurrences, κ should be set to ∞. This has led to the definition of $[1,\infty]$-diagnosability.

Definition 3 ($[1,\infty]$-diagnosability (Jiang et al., 2003; Yoo and Garcia, 2004; Zhou and Kumar, 2009)) A prefix-closed live language L is said to be $[1,\infty]$-diagnosable w.r.t. P and Σ_f if the following holds true:

$$(\exists n \in \mathbb{N})(\forall s \in L)(\forall t \in L/s)\ (|t| \geq n) \Rightarrow [(\forall \omega \in P_L^{-1}(P(st))(N_\omega^F \geq N_s^F)]$$

In the above definition, every sequence estimate $\omega \in P_L^{-1}(P(st))$ of the executed sequence st, with $|t| \geq n$, must have at least the same number of fault occurrences as sequence s.

Property 3 *If a language L is $[1,\infty]$-diagnosable, then $\forall \kappa \geq 1$, L is κ-diagnosable and $[1,\kappa]$-diagnosable. However, the converse does not necessarily hold true.* □

We will now illustrate the different notions of diagnosability introduced so far with an example.

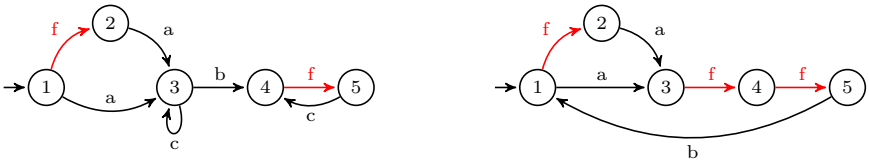
Example 1 *Consider the system models G_1 and G_2, inspired by Jiang et al. (2003) and shown in Figure 2, with $\Sigma_o = \{a,b,c\}$, $\Sigma_u = \Sigma_f = \{f\}$.*

Initially, notice that there are only two different forms of event-sequences generated by G_1, as follows: $s_1^{(m,n)} = ac^m b(fc)^n$ and $s_2^{(m,n)} = fac^m b(fc)^n$, with $P(s_1^{(m,n)}) = P(s_2^{(m,n)}) = ac^m b(c)^n$ $(n,m \in \mathbb{N}^+)$. Let us consider $s_1^{(m,n)} = ac^m b(fc)^n$, with $n \geq 2$. It is clear that $s_1^{(m,n)}$ contains at least 2 occurrences of fault event f. On the other hand, for the same values of m and n, $s_2^{(m,n)}$ contains at least 3 instances of fault event f. Therefore, G_1 is 2-diagnosable.

Let us now consider event sequence $s = fac^p$, which contains only one instance of fault event f. It can be seen from Figure 2a, that there exists in G_1 an event-sequence $s' = ac^p \in L(G_1)$, with the same observation as s, i.e., $P(s) = P(s') = ac^p$, but s' does not contain any occurrence of fault event f. Therefore, G_1 is not 1-diagnosable.

From the above analysis, one can infer that the language generated by G_1 is not $[1,2]$-diagnosable and, obviously, not $[1,\infty]$-diagnosable.

Let us now consider the system modeled by automaton G_2. It can be easily verified that $L(G_2)$ is κ-diagnosable $\forall \kappa \in \mathbb{N}^+$. This is so because every event-sequence $s \in L(G_2)$ that has as observation $P(s) = (ab)^k$, with $k \in \mathbb{N}^+$, has at least $2k$ occurrences of fault event f. So we can choose the delay in Definition 1 as $n_\kappa = 2k$ to satisfy the requirement for κ-diagnosability. Since $L(G_2)$ is κ-diagnosable $\forall \kappa \in \mathbb{N}^+$, then it is also $[1,\kappa]$-diagnosable $\forall \kappa \in \mathbb{N}^+$. On the other hand, $L(G_2)$ is not $[1,\infty]$-diagnosable since the delay bound associated with κ-diagnosability is an increasing function of k, and no 'uniform' delay bound can be found that works for every $\kappa \in \mathbb{N}^+$. □



(a) Model G_1 (b) Model G_2

Fig. 2 Models of Example 1

The above diagnosability properties deal with fault counting assuming uniform bounded delays $n \in \mathbb{N}$, which are independent of the current system execution. In other words, these properties are based on constant bounded delays imposed on the whole faulty behaviors. This notion of uniform delay diagnosability is suitable if a hard deadline for a diagnosis report is required. In several situations, when immediate reaction to fault occurrence is not required, uniformity requirement over diagnosis delays may be too strict. With such a concern in mind, Yoo and Garcia (2009), Yoo and Garcia (2004), Yoo and Garcia (2005), and Yoo and Garcia (2008) proposed a different notion of diagnosability by relaxing the delay uniformity requirement. In this regard, the detection delays are associated with executed sequences and become nonuniform, *i.e.*, they solely depend on the current event-sequence executed by the system.

Definition 4 (Nonuniform $[1, \infty]$-diagnosability (Yoo and Garcia, 2009, 2004)) A prefix-closed live language L is said to be non-uniformly $[1, \infty]$-diagnosable w.r.t. P and Σ_f if the following holds true:

$$(\forall s \in L)(\exists n_s \in \mathbb{N})(\forall t \in L/s)\ (|t| \geq n_s) \Rightarrow [(\forall \omega \in P_L^{-1}(P(st))(N_\omega^F \geq N_s^F)]$$

The above definition means that for every event-sequence s, one can find a finite delay n_s to count the occurrence of faulty events in s. However, one may be able to find a trace s' so that the required delay $n_{s'}$ is larger than n_s. It is worth noticing that similar definitions can be introduced for nonuniform κ- and $[1, \kappa]$-diagnosability properties. From now on, and for the sake of clarity, uniform $[1, \infty]$-diagnosability will be denoted as U$[1, \infty]$-diagnosability whereas nonuniform $[1, \infty]$-diagnosability will be denoted as NU$[1, \infty]$-diagnosability.

The following properties are straightforward from the corresponding definitions of uniform and nonuniform diagnosability properties.

Property 4 *U$[1, \infty]$-diagnosability implies NU$[1, \infty]$-diagnosability (Yoo and Garcia, 2009, 2004).* □

Property 5 *For regular languages, uniform and nonuniform κ- and $[1, \kappa]$-diagnosabilities are equivalent since detection delays can be uniformly bounded by $|X|^2$, the number of states of the automaton that marks the regular language. However, this does not hold true for NU$[1, \infty]$- and U$[1, \infty]$-diagnosability (Yoo and Garcia, 2009).* □

Property 6 *L is NU$[1, \infty]$-diagnosable iff L is κ-diagnosable $\forall \kappa \in \mathbb{N}^+$ (Yoo and Garcia, 2004).* □

Both definitions of uniform and nonuniform $[1, \infty]$-diagnosability requires that each intermittent fault occurrence be detected within a bounded delay that does not depend on the number of fault occurrences.

A variant diagnosability definition, called $\forall\kappa$-diagnosability, which does not require the diagnosis delay bound to be uniform with respect to κ is discussed in Zhou and Kumar (2009). It is based on the fact that it is possible for each fault occurrence to be detectable within a bounded delay which can grow larger as the fault occurrence index κ becomes higher.

Definition 5 ($\forall\kappa$-diagnosability (Zhou and Kumar, 2009) A prefix-closed live language L is said to be $\forall\kappa$-diagnosable ($\kappa \geq 1$) w.r.t. P and Σ_f if the following holds true:

$$(\forall k : 1 \leq k \leq \kappa)(\exists n_k^\kappa \in \mathbb{N})(\forall s \in L, N_s^F \geq k)(\forall t \in L/s)(|t| \geq n_k^\kappa)$$
$$\Rightarrow [(\forall \omega \in P_L^{-1}(P(st))(N_\omega^F \geq k)]$$

Comparing Definitions 1 and 5, we can see that a language is $\forall\kappa$-diagnosable if it is κ-diagnosable $\forall\kappa \geq 1$. Thus, $\forall\kappa$-diagnosability can be seen as a generalization of U$[1, \kappa]$-diagnosability, when κ tends to ∞. Notice that for a $\forall\kappa$-diagnosable system, the diagnosis delay bound may be a function of κ, and, although the diagnosis delay bound must be finite for each κ, the various delays may not be uniformly bounded with respect to κ, *i.e.*, the system may not be U$[1, \infty]$-diagnosable.

Remark 1 *It is worth noticing that $\forall\kappa$-diagnosability is different from NU$[1, \infty]$-diagnosability considered in Yoo and Garcia (2009, 2004), since the non-uniformity in Yoo and Garcia (2009, 2004) comes from the fact that the diagnosis delay bound is a function of the fault event-sequences, namely that the diagnosis delay bound for the κ-th occurrence of a fault can be different for different event-sequences. In contrast, in the case of $\forall\kappa$-diagnosability, the diagnosis delay bound for the κ-th occurrence of a fault is the same for all faulty sequences, namely that, for the detection of the κ-th fault occurrence, the same delay is needed in order for an accurate diagnosability verdict to be issued regardless of the fault-trace executed by the system (Zhou and Kumar, 2009).*

We now illustrate, by means of an example, the definitions of nonuniform and $\forall\kappa$-diagnosability properties.

Example 2 *Let us consider, once again, system models G_1 and G_2 of Example 1 shown in Figure 2. Since L(G_1) is not 1-diagnosable, then it is neither U$[1, \infty]$-diagnosable nor NU$[1, \infty]$-diagnosable. Regarding L(G_2), as shown in Example 1, it is not U$[1, \infty]$-diagnosable. However, L(G_2) is both NU$[1, \infty]$- and $\forall\kappa$-diagnosable. This is so because every event-sequence $s \in L(G_2)$ that has as observation $P(s) = (ab)^k$, with $k \in \mathbb{N}^+$, has at least $2k$ occurrences of fault event f. So, we can choose delays $n_\kappa^k = 2k$ to satisfy the requirement of κ-diagnosability, $\forall\kappa \in \mathbb{N}^+$. The same reasoning can be used to conclude that $L(G_2)$ is $\forall\kappa$-diagnosable.*

We conclude this subsection by summarizing the relationships between the main fault counting diagnosability definitions discussed so far. As shown in Figure 3, U$[1, \infty]$-diagnosability is the strongest diagnosability definition and implies the other three definitions, namely $\forall\kappa$-diagnosability, NU$[1, \infty]$-diagnosability and $[1, \kappa]$-diagnosability (notice that according to the definition

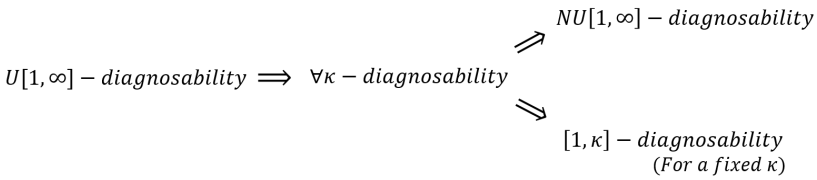


Fig. 3 Relationships between the fault counting-based intermittent fault diagnosability notions.

of $[1,\kappa]$-diagnosability, κ is fixed *a priori*). In addition, $\forall\kappa$-diagnosability implies both NU$[1,\infty]$-diagnosability and $[1,\kappa]$-diagnosability Some further relationships could be established under some restrictions. For instance, $NU[1,\infty]$-diagnosability implies $[1,\kappa]$-diagnosability only if the language describing the system behavior is regular. Moreover if $[1,\kappa]$-diagnosability holds true $\forall\kappa\in\mathbb{N}$, then the system is also $\forall\kappa$-diagnosable and $NU[1,\infty]$-diagnosable.

3.2 Fault detection-based definitions of intermittent fault diagnosability

The classic notion of diagnosability proposed by Sampath et al. (1995) relies on the fact that the faulty status of the system remains fixed after the fault occurrence, *i.e.*, faults are permanent. As a result, fault detection also implies the identification of the faulty status of the system. In the case of intermittent faults, the system status may continuously evolve along with the system evolution. Therefore, detecting the occurrence of such faults does not mean that the current system status has been determined. Consequently, the notion of diagnosability in Sampath et al. (1995) does not take into account all the key issues associated with the diagnosis of intermittent faults.

In this section, we discuss the various notions of diagnosability reported in the literature, as far as intermittent fault detection and system status determination are concerned. For the sake of clarity, we adopt an event-based scheme within the recovery modeling setting.

Let us denote by $\psi(\Sigma_f)$ the set of event-sequences in L that end with faulty event, *i.e.*, $\psi(\Sigma_f)=\{s\sigma_f\in L:\sigma_f\in\Sigma_f\}$. Similarly, $\psi(\Sigma_r)=\{s\sigma_r\in L:\sigma_r\in\Sigma_r\}$ and $\overline{\psi}(\Sigma_r)=\{ss'\in L:s\in\psi(\Sigma_f)\wedge s'\in\psi(\Sigma_r)\}$. Moreover, with a slight abuse of notation, we write $\Sigma_f\in s$ to indicate that a fault event from Σ_f is an event in s, i.e., $\exists\sigma_f\in\Sigma_f$ such that $\sigma_f\in s$. We also recall that the set of system states can be partitioned into three subsets: *Normal*, *Faulty* and *Recovered*, which can be identified using the *labeling function* $\ell:\ L\subseteq\Sigma^*\to\{N,F,R\}$, such that for a given event-sequence $s\in L$, we have:

- $\ell(s)=N \quad if\ (\Sigma_f\notin s)$
- $\ell(s)=F \quad if\ \exists\, s',s'':(s=s's'')\wedge(s'\in\psi(\Sigma_f))\wedge\ (\Sigma_r\notin s'')$
- $\ell(s)=R \quad if\ \exists\, s',s'':(s=s's'')\wedge(\Sigma_f\in s')\wedge(s'\in\psi(\Sigma_r))\wedge(\Sigma_f\notin s'')$

The first two notions of diagnosability to be discussed in the sequel deal only with the detection of fault occurrences and their recovery (Contant et al., 2004, 2002) without necessarily identifying, at any time, the current status of the system.

Such definitions are called "weak diagnosability" (Boussif and Ghazel, 2016a). Then, some restrictive versions of these definitions, which characterize the ability to identify the status of the system after either the occurrence of an intermittent fault or its recovery, will be discussed; they are usually referred to as "strong diagnosability". All of these notions were firstly introduced by Contant et al. (2004), Contant (2005), and Contant et al. (2002), and then revisited in Boussif and Ghazel (2016a), Boussif and Ghazel (2017), and Carvalho et al. (2017), in an event-based scheme within a recovery setting, and in Biswas (2012), Boussif et al. (2016) and Boussif and Ghazel (2019), in a state-based scheme within a normalization setting.

Definition 6 (WF-diagnosability (Contant et al., 2004; Boussif and Ghazel, 2016a)) A prefix-closed live language L is said to be *WF-diagnosable* w.r.t. projection P and Σ_f, if the following holds true:

$$(\exists\, n \in \mathbb{N})(\forall s \in \psi(\Sigma_f))(\forall t \in L/s)(|t| \geq n) \Rightarrow [(\forall \omega \in [P_L^{-1}(P(st))])(\Sigma_f \in \omega)]$$

In Definition 6, W stands for weak and F for fault occurrence. The notion of WF-diagnosability can be interpreted as follows: for every event-sequence s ending with a fault event in Σ_f, and for all sufficiently long continuation t of s ($|t| \geq n$), it is possible to ensure that a fault has occurred based on the captured observation. This implies that all event-sequences that are indistinguishable from st contain at least one fault from Σ_f. It is worth remarking that this definition is exactly that by Sampath et al. (1995) for permanent fault diagnosability since σ_r is not part of the logical expression and, as such, σ_r must be treated as an ordinary subset of the unobservable event set. In other words, the notion of WF-diagnosability accounts only for detecting fault occurrence and is not interested in the system status regarding potential recovery from the fault.

Since we deal with intermittent faults, the fault occurrences are succeeded later on by their corresponding reset event. Therefore, it is worth discussing diagnosability properties that also take into account reset event occurrences.

Definition 7 (WR-diagnosability (Contant et al., 2004; Boussif and Ghazel, 2016a)) A prefix-closed live language L is said to be WR-diagnosable w.r.t. P, Σ_f and Σ_r, if the following holds true:

$$(\exists\, n \in \mathbb{N})(\forall s \in \overline{\psi}(\Sigma_r))(\forall t \in L/s)(|t| \geq n) \Rightarrow [(\forall \omega \in [P_L^{-1}(P(st))])(\Sigma_r \in \omega)]$$

In Definition 7, W stands for weak and R stands for reset occurrence. The notion of WR-diagnosability has the following meaning: there always exists a delay $n \in \mathbb{N}$, such that for every event-sequence s ending with a reset event in Σ_r (which means that at least one fault has occurred and was reset) and for all sufficiently long continuation t of s ($|t| \geq n$), it is possible to detect that the fault has been reset; although it is not possible to infer that either the fault or reset events have occurred more than once. This implies that all of the event-sequences that are indistinguishable from st have necessarily experienced a fault occurrence and recovery. Notice that, similar to WF-diagnosability, there is no constraint regarding the determination of the system status when the recovery is diagnosed. It is worth noticing that, from Definition 7, if the system is WR-diagnosable, we are only capable to infer whether or not the system has recovered from the fault at least once, and so, it is not possible to state if the current status of the system is faulty or recovered.

The previous notions of intermittent fault diagnosability serve only to detect the occurrence of the fault (or its reset) but they provide no information regarding the system status at any time. In order to take the system status into account, strong versions have been introduced.

Definition 8 (SF-diagnosability (Contant et al., 2004)) A prefix-closed live language L is said to be *SF-diagnosable* w.r.t. P, Σ_f and Σ_r, if the following holds true:

$$(\exists n \in \mathbb{N})(\forall s \in \psi(\Sigma_f))(\forall t \in L/s)(|t| \geq n) \Rightarrow$$
$$\left[\exists t' \leq t : \forall \omega \in [P_L^{-1}(P(st'))] \Rightarrow \ell(\omega) = F\right]$$

In Definition 8, S stands for strong. SF-diagnosability states that for every event-sequence s that ends with a fault event in Σ_f, and for all sufficiently long continuation t of s, one can detect the fault occurrence and determine, with certainty, the faulty status of the system after the occurrence of at most n events, based on the captured observations. This implies that all event-sequences that are indistinguishable from st lead the system to fault states at the same observation point, within a finite delay after the occurrence of the fault.

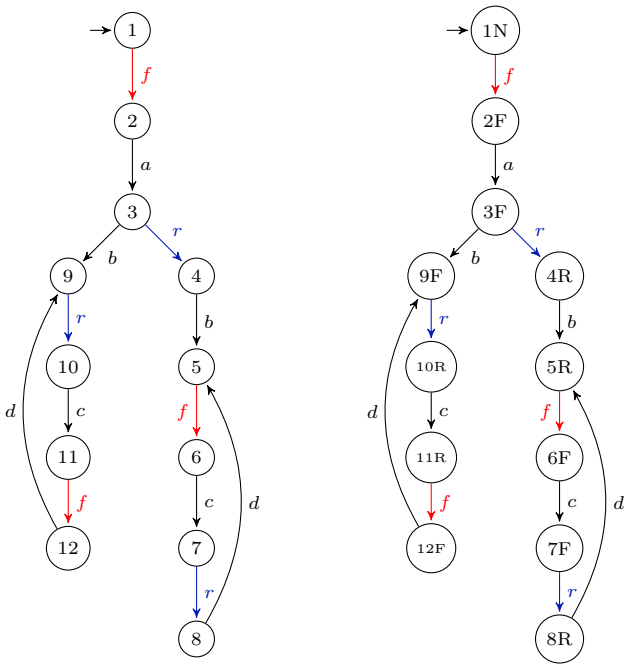
Similarly, SR-diagnosability, the dual notion of SF-diagnosability, can be introduced as follows.

Definition 9 (SR-diagnosability (Contant et al., 2004)) A prefix-closed live language L is said to be *SR-diagnosable* w.r.t. P, Σ_f and Σ_r, if the following holds true:

$$(\exists n \in \mathbb{N})(\forall s \in \overline{\psi}(\Sigma_r))(\forall t \in L/s)(|t| \geq n) \Rightarrow$$
$$\left[\exists t' \leq t : \forall \omega \in [P_L^{-1}(P(st'))] \Rightarrow \ell(\omega) = R\right]$$

According to Definition 9, SR-diagnosability ensures that for every event-sequence s ending with a reset event in Σ_r, and for every sufficiently long continuation t of s, one can detect the reset of the fault and determine, with certainty, the recovery status of the system based on the captured observations. This implies that all of the event sequences that are indistinguishable from st lead to recover states at the same observation point, within a finite delay after the fault recovery.

Example 3 *Consider the system model G (taken from Contant et al. (2002)), shown in Figure 4(a), with $\Sigma_o = \{a, b, c, d\}$ and $\Sigma_u = \{f, r\}$, and assume that $\Sigma_f = \{f\}$ and $\Sigma_r = \{r\}$. The corresponding label automaton $G_\ell = G \parallel \Omega$ is depicted in Figure 4b. The faulty event-sequences in G are $\rho_1 = farb\,(fcrd)^*$ and $\rho_2 = fab\,(rcfd)^*$, with $P(\rho_1) = P(\rho_2) = ab(cd)^*$. Thus, it is not difficult to see that $L(G)$ is both WF- and WR-diagnosable, since it is possible to infer the occurrence of both fault and recover events (events f and r, respectively). However, $L(G)$ is not SF-diagnosable, since no bound delay n exists, such that after this limit, both ρ_1 and ρ_2 lead to fault states at the same time, which means that it is not possible to determine the faulty status of the system. Similar reasoning leads to the conclusion that $L(G)$ is not SR-diagnosable. These two points will be made clear in the next section when we introduce the notion of diagnoser.*



(a) Model G (b) Model G_ℓ

Fig. 4 Automata G and G_ℓ for Example 3

Diagnosability notions WF, WR, SF and SR consider the detection/ identification of the fault/reset occurrences within finite delays. However, they do not take into account the multiplicity of fault/reset occurrences. In other words, a fault can occur and reset several times before its detection/identification. In Boussif and Ghazel (2018), Fabre et al. (2016), and Fabre et al. (2018), a stronger notion of intermittent diagnosability was introduced. It consists in not only detecting each fault occurrence within a finite delay, but also before its reset. Hereafter, we refer to this new property as F_r-diagnosability.

Definition 10 (F_r-diagnosability (Boussif and Ghazel, 2018; Fabre et al., 2016, 2018)) A prefix-closed live language L is said to be *F_r-diagnosable* w.r.t. P, Σ_f and Σ_r, if the following holds true:

$$(\forall s \in \psi(\Sigma_f))(\forall t \in L/s : t \in \bar{\bar{\psi}}(\Sigma_r)) \Rightarrow \\ \left[\exists t' < t : \forall \omega \in [P_L^{-1}(P(st'))] \Rightarrow \ell(\omega) = F\right],$$

where $\bar{\bar{\psi}}(\Sigma_r) = \{s = \sigma_1\sigma_2\ldots\sigma_n \in \Sigma^* : (\sigma_i \notin \Sigma_r, i = 1, 2, \ldots, n-1) \wedge (\sigma_n \in \Sigma_r)\}$ is the set of finite event-traces in L/s whose unique event in Σ_r is the last one.

Definition 10 can be interpreted as follows: let s be a finite event-sequence in L that ends with a faulty event, and t be every finite continuation of s that ends with a reset event but does not have any reset event before its last event. Then, all

the finite event-sequences that share the same observation with st, must take the system to a faulty status between the moment when the fault has occurred and its recovery. In other words, when a fault event occurs, one needs to be able to detect it and identify the faulty status of the system before it resets. Such a feature can be of interest for maintenance operation, for instance. Analogously, one may require to detect the recovered status of the system before a new occurrence of the fault, for this reason the dual version of F_r-diagnosability, the so-called R_f-diagnosability that consists in detecting and identifying that the system reaches a recovered status after each occurrence of a reset event and before every new occurrence of the corresponding fault event, is defined.

Property 7 *As expected:*

(a) F_r*-diagnosability* $\Rightarrow$ SF*-diagnosability* $\Rightarrow$ WF*-diagnosability*
(b) R_f*-diagnosability* $\Rightarrow$ SR*-diagnosability* $\Rightarrow$ WR*-diagnosability* □

Property 7 summarizes the existing relationships between the main fault detection definitions discussed in this section. As it can be seen, F_r-diagnosability (resp. R_f-diagnosability) is the strongest property regarding fault detection (resp. recovery detection) since it implies SF-diagnosability (resp. SR-diagnosability), which in turn implies WF-diagnosability (resp. WR-diagnosability). Notice that no relationship can be established between F_r- and R_f-diagnosability definitions on one side, nor between SF- and SR-diagnosability on the other side. Regarding WF- and WR-diagnosability, an equivalence relation may exist under some assumptions, as will be discussed later on in the paper.

4 Diagnosability analysis and diagnoser synthesis

In this section, we summarize the main approaches developed for verifying the various notions of intermittent fault diagnosability for DES modeled by FSA. These approaches can be divided in three classes, as follows: (i) diagnoser-based approaches; (ii) twin-plant-based approaches, and; (iii) verifier-based approaches. They have been firstly introduced to deal with permanent failures, and, recently, they have been adapted to deal with intermittent faults.

(i) Diagnoser-based approaches. Diagnoser-based approaches are based on the construction of a deterministic automaton, called diagnoser, which keeps track of all possible state estimation of the system based on the observed event-sequences (Cassandras and Lafortune, 2008; Sampath et al., 1995; Hashtrudi Zad et al., 2003; Viana and Basilio, 2019). The diagnoser can be thought of as an extended observer that provides (i) an estimate of the current state of the system after the occurrence of an observable event and (ii) information on potential past failure occurrences in the form of labels, and so, each state of the diagnoser is composed of a set of the system state estimations which indicate that the system is in its normal or faulty behavior. They are used to check diagnosability by verifying the existence of particular ambiguous cycles, called indeterminate cycles (Sampath et al., 1995), and, once the system language is checked to be diagnosable, the diagnoser can also be used to perform online diagnosis.

(ii) Twin-plant-based approaches. A twin-plant is a non-deterministic automaton whose states are composed of a pair of system states (which can be normal or faulty), and whose paths correspond to a pair of event-sequences in the system model that share the same observation. Its structure is exploited to analyse diagnosability by searching for 'bad' cycles (called F-confused cycles, or infinite critical pair) (Cimatti et al., 2003; Boussif and Ghazel, 2015). An F-confused cycle is composed exclusively of ambiguous states, *i.e.*, states in the twin-plant containing one normal and one faulty state. Using the twin-plant, diagnosability of permanent faults can be checked using polynomial-time algorithm(s) (Jiang et al., 2001; Schumann and Pencole, 2007).

(iii) Verifier-based approaches. Verifier-based approaches (Yoo and Lafortune, 2002; Moreira et al., 2011; Grastien, 2009; Qiu and Kumar, 2006) usually rely on the construction of a non-deterministic automaton[2] V_{Σ_f} called Σ_f-verifier, where Σ_f is the fault class. Verifiers are built by performing a parallel composition of the system model with itself (regarding the observable events) augmented with a tagging function. Differently from twin-plants, verifiers are built directly from the system model, and, for this reason, possess both observable and unobservable events. Like twin-plants, verifier states are composed of a pair of system states (which can be normal or faulty). The diagnosability analysis is based on the search for F-confused cycles.

In the following two sections, we discuss the existing works that rely on these techniques for checking the various notions of intermittent fault diagnosability.

4.1 Verification of fault counting-based diagnosability notions

The verification of fault counting properties are mostly based on some variant of the verification automaton used to carry it out. We, start by making the following assumption[3]:

A2. Language L is live, *i.e.*, for all $x \in X$, there exist $\sigma \in \Sigma$ and $x' \in X$ such that $\delta(x, \sigma) = x'$.

4.1.1 Verification of uniform diagnosability

Jiang et al. (2003) proposed a verifier-like automaton to check κ- and $[1, \kappa]$-diagnosability properties, which is a transition graph, *i.e.*, no event labels are associated with transitions. For an automaton $G = (X, \Sigma, \delta, x_0)$, the verifier-like automaton is denoted as $\mathcal{V} = (X', \mathcal{R}, x'_{00})$, where $X' = X \times X$ is the set of states, $x'_{00} = (x_0, x_0)$ is the initial state, and $\mathcal{R} \subseteq X' \times X'$ is the state transition defined as follows: two states $x_{12} = (x_1, x_2)$ and $x'_{12} = (x'_1, x'_2)$, $x_{12}, x'_{12} \in X'$, are such that $x_{12} \to x'_{12} \in R$, if, and only if, one of the two following conditions holds true:

1. $x_1 = x'_1$ (resp. $x_2 = x'_2$), and $\exists \sigma \in \Sigma_u$ such that $(x_2, \sigma, x'_2) \in \delta$ (resp. $(x_1, \sigma, x'_1) \in \delta$);

[2] The only exception is the verifier proposed by Moreira et al. (2011). See also Kumar and Takai (2014) and Moreira et al. (2016).

[3] Every made assumption is to be applied to the remainder of the text unless explicitly indicated. Nevertheless, for the sake of clarity, we will indicate in all results which assumptions are being required.

2. $\exists \sigma \in \Sigma_o$ such that $(x_1, \sigma, x_1') \in \delta$ and $(x_2, \sigma, x_2') \in \delta$;

In order to keep tracking of the number of fault occurrences, transition graph $\mathcal{V}$ is augmented with some counters and Boolean variables, being renamed as $\mathcal{V}_1$, as follows:

- a tagging value-pair $(min\{N^F_{s_1}, \kappa\}, min\{N^F_{s_2}, \kappa\})$ associated with each state (x_1, x_2) in $\mathcal{V}$, with $x_1 \in \delta(x_0, s_1)$ and $x_2 \in \delta(x_0, s_2)$, where, as defined before, $N^F_{s_i}$ is the number of fault events in a given event-sequence s_i;
- a Boolean variable $v \in \{0, 1\}$ that indicates whether or not x_1 or x_2 is reachable by a state-trace ρ, sufficiently long, which has a higher number of faults.

The general structure of a state-pair x_{12} in the augmented transition graph $\mathcal{V}_1$ is then $x_{1,2} = ((x_1, x_2), (min\{N^F_{s_1}, \kappa\}, min\{N^F_{s_2}, \kappa\}), v)$.

Theorem 1 *(Jiang et al., 2003) — Under assumption **A2**:*

- *a system model G is κ-diagnosable w.r.t. P and Σ_f iff no cycle containing a state-pair $((x_1, x_2), (\kappa, i), 1)$ with $i < \kappa$, exists in $\mathcal{V}_1$.*
- *a system model G is $[1, \kappa]$-diagnosable w.r.t. P and Σ_f iff no cycle containing a state-pair $((x_1, x_2), (j, i), 1)$ with $i < j$, exists in $\mathcal{V}_1$, $\forall 1 \leq j \leq \kappa$.*

We will now illustrate the verification method proposed by Jiang et al. (2003) with the following example.

Example 4 *Consider, again, system model G_1 of Figure 2a. A relevant part of the augmented transition graph $\mathcal{V}_1$ associated with G_1 is shown in Figure 5, from which we can notice that $\mathcal{V}_1$ contains a cycle (self-loop) composed of state $((3, 3), (1, 0), 1)$, which implies that G_1 is not 1-diagnosable. In fact, this result is expected since, as discussed in Example 2, the fault counting process fails when it comes to handle sequences ac^*. To establish its relationship with verifier $\mathcal{V}_1$, first notice that trace ac^* is an observed sequence, which corresponds in model G_1 (Figure 5) to two sequences, one faulty and another non-faulty, both leading to state 3 of the system model G_1; hence the pair $(3, 3)$ in the verifier state $((3, 3), (1, 0), 1)$, which corresponds to the states reached by these two sequences. In addition, the second component $((1, 0))$ indicates that one sequence has a fault event, while the second one does not. Finally, since the faulty sequence is infinite (i.e., it leads to the self-loop), then the Boolean variable in the verifier state is equal to '1'. Such a configuration ensures that fault counting process fails in detecting the number of faults, when dealing with observable sequence ac^*.*

The above approach consists in counting the number of fault occurrences in each event-sequence. Such an approach cannot be used directly for the analysis of U$[1, \infty]$-diagnosability, since automaton $\mathcal{V}_1$ may be *unbounded*, i.e., if one keeps track of the number of faults with each state-trace in each state-trace pair, then one may get a transition graph with an infinite number of states. In order to overcome this drawback, instead of keeping track of the number of faults with each state-trace in each trace-pair, we only keep tracking of the difference between the number of faults with the state-pair traces. Although the difference in terms of the number of faults in the state pairs may still be unbounded, it can be shown that if it goes above an upper bound, say $|X|^2$, the system model is not U$[1, \infty]$-diagnosable (Jiang et al., 2003). Hence, in order to verify U$[1, \infty]$-diagnosability, the label of the vertices of transition graph $\mathcal{V}$ must be augmented, with the following parameters:

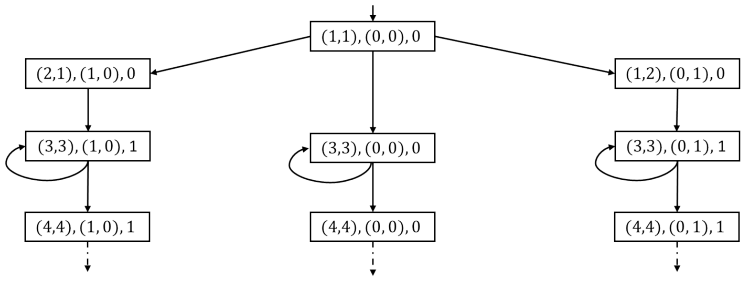


Fig. 5 Augmented transition graph $\mathcal{V}_1$ associated with model G_1 of Example 1.

- a fault-difference counter $d_f = N^F_{s_1} - N^F_{s_2}$ associated with each state pair (x_1, x_2) in $\mathcal{V}$, with $x_1 \in \delta(x_0, s_1)$ and $x_2 \in \delta(x_0, s_2)$;
- a Boolean variable $b \in \{0, 1\}$ that indicates whether or not the current fault occurrence must be reported, *i.e.*, whether or not both event-sequences associated with the trace-pair (in $\mathcal{V}$) contain, after the fault occurrence, at least one fault event each.
- a Boolean variable $v \in \{0, 1\}$ that indicates whether or not the reached state pair (x_1, x_2) results from an extension of an state-trace with higher number of faults.

The general structure of a state-pair in the augmented transition graph, denoted as $\mathcal{V}_2$, will, therefore, be $x_{12} = ((x_1, x_2), d_f, b, v)$, starting from the initial vertex: $x_{12} = ((x_1, x_2), 0, 0, 0)$. Given a vertex $x_{12} = ((x_1, x_2), d_f, b, v)$, a new vertex $x'_{12} = ((x'_1, x'_2), d'_f, b', v')$ is defined accordance with the following rule:

$$\text{If } \exists\, \sigma \in \Sigma_o \text{ s.t. } \delta(x_1, \sigma) = y_1 \text{ and } \delta(x_2, \sigma) = y_2, \text{ then } (x'_1, x'_2) = (y_1, y_2),$$
$$d'_f = d_f, \qquad b' = 0, \qquad v' = \begin{cases} 1 & \text{if } d_f \neq 0 \\ 0 & \text{otherwise} \end{cases};$$
$$\text{If } \exists\, \sigma_1, \sigma_2 \in \Sigma_f \text{ s.t. } \delta(x_1, \sigma_1) = y_1 \text{ and } \delta(x_2, \sigma_2) = y_2, \text{ then } (x'_1, x'_2) = (y_1, y_2),$$
$$d'_f = d_f, \qquad b' = 1, \qquad v' = \begin{cases} 1 & \text{if } d_f \neq 0 \\ 0 & \text{otherwise} \end{cases};$$
$$\text{If } \exists\, \sigma \in \Sigma_u \text{ s.t. } \delta(x_1, \sigma) = y_1, \text{ then } (x'_1, x'_2) = (y_1, x_2),$$
$$d'_f = \begin{cases} d_f + 1, & \text{if } \sigma \in \Sigma_f \\ d_f, & \text{otherwise} \end{cases}, \quad b' = \begin{cases} 1, & \text{if } |d'_f| < |d_f| \\ 0, & \text{otherwise} \end{cases}, \quad v' = \begin{cases} 1, & \text{if } d_f > 0 \\ 0, & \text{otherwise} \end{cases};$$
$$\text{If } \exists\, \sigma \in \Sigma_u \text{ s.t. } \delta(x_2, \sigma) = y_2, \text{ then } (x'_1, x'_2) = (x_1, y_2),$$
$$d'_f = \begin{cases} d_f - 1, & \text{if } \sigma \in \Sigma_f \\ d_f, & \text{otherwise} \end{cases}, \quad b' = \begin{cases} 1, & \text{if } |d'_f| < |d_f| \\ 0, & \text{otherwise} \end{cases}, \quad v' = \begin{cases} 1, & \text{if } d_f < 0 \\ 0, & \text{otherwise} \end{cases}.$$

Theorem 2 *(Jiang et al., 2003) — Under assumption **A2**, the language generated by a system model G is $U[1, \infty]$-diagnosable w.r.t. P and Σ_f, iff:*

(a) the augmented transition graph $\mathcal{V}_2$ corresponding to G is finite, and
(b) no cycle containing a state-pair $((x_1, x_2), d_f, 0, 1)$, with $d_f \neq 0$, exists in $\mathcal{V}_2$.

Let us now illustrate the verification of $U[1, \infty]$-diagnosability using $\mathcal{V}_2$.

Example 5 *Let us consider once again automaton G_1 of Figure 2a. The augmented transition graph $\mathcal{V}_2$ associated with G_1 is shown in Figure 6, from where we can see*

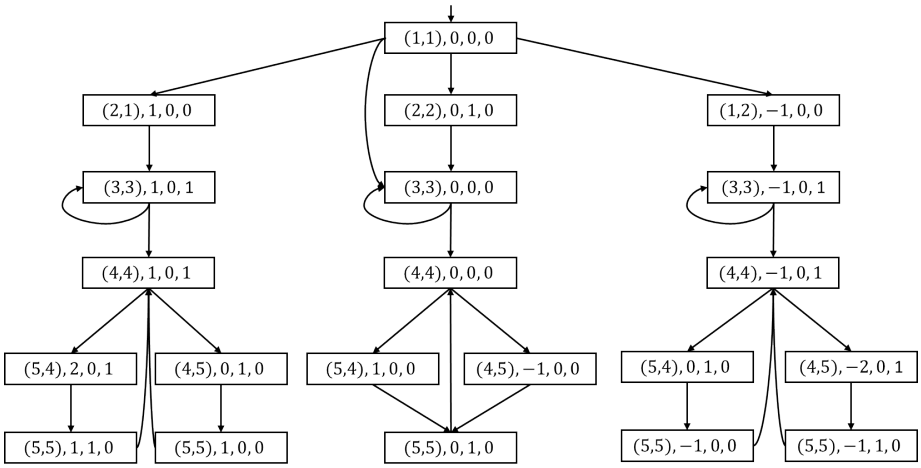


Fig. 6 Augmented transition graph $\mathcal{V}_2$ associated with model G_1 of Example 1.

that $\mathcal{V}_2$ contains cycles that have some states $((x_1, x_2), d_f, 0, 1)$ where $d_f \neq 0$; e.g., state $((4,4), -1, 0, 1)$. Thus, $L(G_1)$ is not $U[1,\infty]$-diagnosable.

Remark 2 (Complexity analysis (Jiang et al., 2003)) *The number of states and transitions of $\mathcal{V}_1$ (resp. $\mathcal{V}_2$) for checking κ-/$[1,\kappa]$-diagnosability (resp. $U[1,\infty]$-diagnosability) using the above approach are, in the worst case, $2|X|^2 \times (\kappa+1)^2$ and $8|X|^4 \times +4|X|^2$ (resp. $2|X|^4 \times (\kappa+1)^2$ and $8|X|^6 + 4|X|^4$), where $|X|$ is the number of states in the plant automaton. For the verification, Jiang et al. (2003) proceed as follows: first, particular states in the augmented transition graphs are searched, and then, it is checked if such states belong to some cycles or not; e.g., for $U[1,\infty]$-diagnosability, states of form $((x_1, x_2), d_f, 0, 1)$ are investigated. For deterministic systems, the overall complexity for analyzing κ/$-[1,\kappa]$-diagnosability (resp. $U[1,\infty]$-diagnosability) is $\mathcal{O}(|X|^2|\Sigma|^2)$ (resp. $\mathcal{O}(|X|^4|\Sigma|^2)$).*

Remark 3 (Online diagnosis) *Jiang et al. (2003) developed a systematic procedure for online diagnosis of κ-, $[1,\kappa]$- and $U[1,\infty]$-diagnosable systems, whose idea behind its construction is that it must determine the potential states of the system after each observation. The procedure consists in maintaining a state estimator $(Q_d, I_d) \in 2^{X \times \mathbb{N}} \times \mathbb{N}$, where Q_d is a set of state estimations that can be reached following an observed event-sequence, and I_d is a count indicator used to store either the total number of detected faults (for κ- and $[1,\kappa]$-diagnosis) or the total number of newly detected faults (for $U[1,\infty]$-diagnosis). According to Jiang et al. (2003), the size of Q_d is bounded by $|X| \times (\kappa+1)$ (for κ- and $[1,\kappa]$-diagnosis) and $|X| \times (|X|^2+1)$ (for $U[1,\infty]$-diagnosis).*

4.1.2 Verification of nonuniform intermittent fault diagnosabilities

Regarding NU$[1,\infty]$-diagnosability, Yoo and Garcia (2009) and Yoo and Garcia (2004) proposed another verifier variant that is based on the construction of a weighted graph which can be leveraged to compute shortest-paths (Goldberg, 1995; Cherkassky et al., 1996).

The weighted graph, denoted as $\mathcal{W} = (V, E, w, v_0)$,where $V \subseteq X \times X$ is a finite set of vertices, $E \subseteq V \times V$ is a finite set of edges, $v_0 = (x_0, x_0)$ is the initial vertex, and $w : E \to S$ is the edge weighting function, with $S = \{-1, 0^+, 0^-, 0, \hat{0}, +1\}$ being the set of weight symbols. In this regard, for two vertices $v_1, v_2 \in V$, there can be associated an edge $(v_1, v_2) \in E$ possessing weight $w[v_1, v_2] \in S$. We write $v_1 \xrightarrow{i} v_2$ to denote that $i \in w[v_1, v_2]$. Regarding the edges of $\mathcal{W}$, they are defined as follows: let $x_1, x_1', x_2, x_2' \in X$ and $\sigma_1, \sigma_2 \in \Sigma$, with $\delta(x_1, \sigma_1) = x_1'$ and $\delta(x_2, \sigma_2) = x_2'$. Then,

(i) If $\sigma_1 = \sigma_2 \in \Sigma_o$, then $(x_1, x_2) \xrightarrow{0} (x_1', x_2')$ if $(\delta(x_1, \sigma_1) = x_1' \wedge \delta(x_2, \sigma_2) = x_2')$
(ii) If $\sigma_1, \sigma_2 \in \Sigma_u$, then

$$(x_1, x_2) \begin{cases} \xrightarrow{-1} (x_1', x_2) \text{ if } (\sigma_1 \in \Sigma_f) \wedge (\delta(x_1, \sigma_1) = x_1'); \\ \xrightarrow{+1} (x_1, x_2') \text{ if } (\sigma_2 \in \Sigma_f) \wedge (\delta(x_2, \sigma_2) = x_2'); \\ \xrightarrow{0^-} (x_1', x_2) \text{ if } (\sigma_1 \notin \Sigma_f) \wedge (\delta(x_1, \sigma_1) = x_1'); \\ \xrightarrow{0^+} (x_1, x_2') \text{ if } (\sigma_2 \notin \Sigma_f) \wedge (\delta(x_2, \sigma_2) = x_2'); \\ \xrightarrow{\hat{0}} (x_1', x_2') \text{ if } (\sigma_1, \sigma_2 \in \Sigma_f) \wedge (\delta(x_1, \sigma_1) = x_1' \wedge \delta(x_2, \sigma_2) = x_2'); \\ \xrightarrow{0} (x_1', x_2') \text{ if } (\sigma_1, \sigma_2 \notin \Sigma_f) \wedge (\delta(x_1, \sigma_1) = x_1' \wedge \delta(x_2, \sigma_2) = x_2') \end{cases}.$$

Essentially, $\mathcal{W}$ is obtained by performing a synchronized product of the FSA G_1 with itself, in accordance with the above rules.

For a path $\rho = v_0, v_1, \ldots, v_k$ in $\mathcal{W}$ of length k, the weight of ρ is the sum of the minimum weights of its constituent edges, that is:

$$w[\rho] = \sum_{i=1}^{k} w[v_{i-1}, v_i],$$

where the zero weights $0, 0^+, 0^-$, and $\hat{0}$ are all considered as 0 when the above operations over the weights is performed.

The shortest-path weight from v_0 to v_k is:

$$\text{short}[v_0, v_k] = \begin{cases} \min(w[\rho]) & \text{if } v_k \text{ is reachable from } v_0 \text{ via } \rho \\ \infty & \text{otherwise} \end{cases}$$

In order to formulate a necessary and sufficient condition for NU$[1, \infty]$-diagnosability, Yoo and Garcia (2004) define a T-cycle ($T \subseteq S$) in $\mathcal{W}$ as follows. Let $V_{cl} = \{v_1, v_2, \ldots, v_n\}$ be a set of vertices that forms a cycle in $\mathcal{W}$. Then the T-cycle of $\mathcal{W}$ associated with V_{cl} is a set of edges formed as follows: $T = \{t \in S : (\exists v_i, v_j \in V_{cl})[v_i \xrightarrow{t} v_j]\}$.

Theorem 3 (Yoo and Garcia, 2004) — *Under assumption **A2**, the language $L(G)$ generated by an automaton G is NU$[1, \infty]$-diagnosable w.r.t. P and Σ_f, if, and only if, the following three conditions hold true:*

(a) $\forall$ *T-cycle in $\mathcal{W}$, if $-1 \in T$ then either $\hat{0} \in T$ or $+1 \in T$;*
(b) $\forall\, v \in \{0^-\}$*-cycle in $\mathcal{W}$ then* $\text{short}[v_0, v] \geq 0$.
(c) $\forall\, v \in T$*-cycle in $\mathcal{W}$, such that $T \in \{\{0\}, \{0^-, 0\}, \{0^+, 0\}, \{0^-, 0, 0^+\}\}$ then* $\text{short}[v_0, v] = 0$.

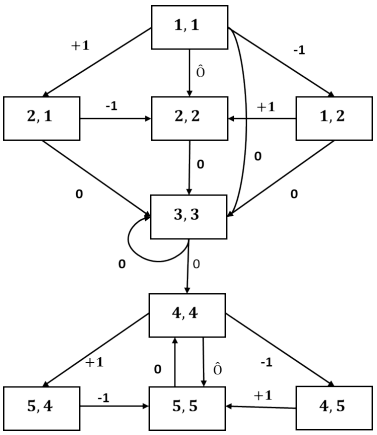


Fig. 7 The weighted graph $\mathcal{W}$ associated with model G_1 of Example 1.

Example 6 *Figure 7 shows the weighted graph $\mathcal{W}$ that corresponds to model G_1 in Example 1* (cf. Figure 2a)*, from where, it is possible to see that $\mathcal{W}$ contains four T-cycles: $T_1 = \{+1, -1, 0\}$, $T_2 = T_1$, $T_3 = \{0, \hat{0}\}$ and $T_4 = \{0\}$, associated with the sets of vertices $V_1 = \{(4,4),(4,5),(5,5)\}$, $V_2 = \{(4,4),(5,4),(5,5)\}$, $V_3 = \{(4,4),(5,5)\}$ and $V_4 = \{(3,3)\}$ respectively. Notice that T-cycles T_1, T_2 and T_3 satisfy condition (a), and trivially satisfy conditions (b) and (c) of Theorem 3. Regarding T_4-cycle, it can be seen that it trivially satisfies conditions (a) and (b). Let us check condition (c). There are various paths that connect the initial vertex to the edge that forms the $\{0\}$-cycle in vertex $(3,3)$, for example, paths $\rho_1 = (1,1) \xrightarrow{+1} (2,1) \xrightarrow{0} (3,3)$, and $\rho_2 = (1,1) \xrightarrow{-1} (1,2) \xrightarrow{0} (3,3)$. It is not difficult to check that* $\text{short}[(1,1),(3,3)] = -1$*, which violates condition (c) of Theorem 3. Thus, model G_1 is not $NU[1,\infty]$-diagnosable, which is consistent with the analysis carried out in Example 2.*

Remark 4 (Complexity analysis (Yoo and Garcia, 2009, 2004)) *Since the number of states and transitions in the weighted graph $\mathcal{W}$ are $|X|^2$ and $|X|^2 \times |\Sigma|^2$, respectively, and the verification algorithm is based on the search for strongly connected components,the verification algorithm proposed by Yoo and Garcia (2009) has lower computation complexity when compared to that of Jiang et al. (2003). Thus, the overall computational complexity for checking $NU[1,\infty]$- and $U[1,\infty]$-diagnosabilities of a deterministic system model according to the above procedure is $\mathcal{O}(min(|X|^3 \times |\Sigma|^2, |X|^5))$ time and $\mathcal{O}(min(|X|^2 \times |\Sigma|^2, |X|^4))$ space, where $|X|$ and $|\Sigma|$ are the numbers of states and events in the system model, respectively.*

Remark 5 (Online diagnosis) *Yoo and Garcia (2009) proposed an online algorithm for counting, in an efficient way, the number of fault occurrences, which is based on a recursive computation of $min\{N_s^F : s \in P_L^{-1}(\omega)\}$, with $\omega \in \Sigma_o^*$ being an observable event-sequence. The idea behind this recursive computation is to record the model state estimations and the corresponding minimum number of fault occurrences in the diagnoser state, and so, $Q_d = \{(q_1, i_1), \ldots (q_n, i_n)\} \in 2^{X \times \mathbb{N}}$, where $\min_{j=1\ldots n}(i_j)$ will be the current fault count. Such an online algorithm can be conducted with $\mathcal{O}(|\Sigma + log|X|) \times |X|)$ complexity.*

4.1.3 Verification of $\forall\kappa$-diagnosability

In order to check $\forall\kappa$-diagnosability, Zhou and Kumar (2009) proposed an approach based on the construction of the so-called indistinguishable-pair automaton (IPA). The IPA is a variant of the verifier automaton where the set of states contains pairs of the form $((x_+, x_-), (w(x_+), w(x_-))) \in X^2 \times \{0,1\}^2$, with $(w(x_+), w(x_-)) \in \{0,1\}^2$ is a vector-weight associated with state-pair $(x_+, x_-) \in X^2$ and used to count the fault occurrences. The set of events contains pairs of the form $(\sigma_+, \sigma_-) \in (\Sigma_\epsilon^2 \setminus \Sigma_u^2)$, with $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$.

Formally, the IPA is defined as $G_I = (Y, \Sigma_I, \delta_I, y_0)$, where $Y \in X^2 \times \{0,1\}^2$ is the set of states, $\Sigma_I = \Sigma_\epsilon^2 \setminus \Sigma_u^2$ is the set of events, $y_0 = ((x_0, x_0), (0,0))$ is the initial state, and $\delta_I : Y \times \Sigma_I \to Y$ is the transition function, being each transition of the form $(y, (\sigma_+, \sigma_-), y')$, with $y = ((x_+, x_-), (w(x_+), w(x_-))$ and $y' = ((x'_+, x'_-), (w(x'_+), w(x'_-))$, where:

$$(x'_+, x'_-) = \begin{cases} (\delta(x_+, \sigma_+), \delta(x_-, \sigma_-)), & \text{if } \sigma_+ = \sigma_- \in \Sigma_o \ \wedge \delta(x_+, \sigma_+)! \wedge \delta(x_-, \sigma_-)!, \\ (\delta(x_+, \sigma_+), x_-), & \text{if } \sigma_+ \in \Sigma_u \wedge \sigma_- = \epsilon \wedge \delta(x_+, \sigma_+)!, \\ (x_+, \delta(x_-, \sigma_-)), & \text{if } \sigma_- \in \Sigma_u \wedge \sigma_+ = \epsilon \wedge \delta(x_-, \sigma_-)!, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

and

$$w(x'_+) = \begin{cases} 1, & \text{if } x'_+ = \delta(x_+, \sigma_+) \wedge (\sigma_+ \in \Sigma_f), \\ 0, & \text{otherwise,} \end{cases}$$

and similarly for $w(x'_-)$. In the equations above, $\delta(x, \sigma)!$ is defined, i.e., there exists a state $x' \in X$ such that $\delta(x, \sigma) = x'$.

The vector-weight components $w(x'_+)$ and $w(x'_-)$ are called the positive and negative weight of y, respectively, and are denoted by $+ve$ and $-ve$. The weight of a state y is given by $w(y) = w(x_+) - w(x_-)$, and the weight of a sequence of states $\pi = y_1, \ldots, y_n$ of G_I is given by $w(\pi) = \sum_{i=1}^{n} w(y_i) = w_+(\pi) - w_-(\pi)$, where $w_+(\pi) = \sum_{i=1}^{n} w(x_{i_+})$ and $w_-(\pi) = \sum_{i=1}^{n} w(x_{i_-})$.

Several types of state-sequences (possibly cycles) can be distinguished with respect to the sequence weight. For a given state-sequence π in G_I, we say that π is:

- fault-free, if $w_+(\pi) = w_-(\pi) = 0$;
- a $+ve$-path (resp. a $-ve$-path), if $w(\pi) > 0$ (resp. $w(\pi) < 0$);
- a $+ve$-part fault-free (resp. *$-ve$-part fault-free*) path, if $w_+(\pi) = 0$ (resp. $w_-(\pi) = 0$);
- a $+ve$-vocal (resp. $-ve$-vocal) path, if π contains a transition (σ_+, σ_-) with $\sigma_+ \in \Sigma_o$ (resp. $\sigma_- \in \Sigma_o$).

Notice that vocality ensures the execution of at least one observable event along the path. In addition, based on these path definitions, it is possible to identify cyclic state-sequences; for example, a $+ve$-vocal cycle cl is a cycle that contains a transition (σ_+, σ_-) with $\sigma_+ \in \Sigma_o$.

Recall that, according to Definition 5, a system is not $\forall\kappa$-diagnosable if, and only if, there exists a pair of indistinguishable sequences π and π' infinitely-long, such that $N_\pi^F < \kappa$ while $N_{\pi'}^F \geq \kappa$ (or, symmetrically, the converse) and the path with the larger number of faults is infinitely vocal, i.e., the path continues to execute observable events regularly.

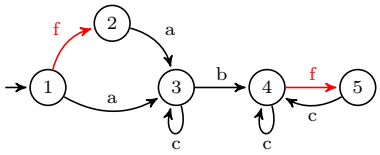


Fig. 8 System model G_3 of Example 7

Based on automaton G_I, the following necessary and sufficient condition for $\forall\kappa$-diagnosability can be stated.

Theorem 4 (Zhou and Kumar, 2009) — *Under assumption **A2**, language $L(G)$ generated by system model G is not $\forall\kappa$-diagnosable w.r.t. P and Σ_f, if, and only if, either one of the following two conditions holds true:*

(a) there exists a cycle cl in G_I that is both $+$ve-part fault-free and $-$ve-vocal;
(b) there exists a $-$ve-path that begins at the initial state of G_I and is connected to a cycle that is simultaneously fault-free and $-$ve-vocal.

The next example illustrates both the construction of IPA G_I and Theorem 4.

Example 7 *Consider the system model G_3 in Figure 8, where $\Sigma_o = \{a, b, c\}$ and $\Sigma_u = \Sigma_f = \{f\}$. Figure 9 depicts the IPA G_{3_I} corresponding to model G_3. From G_{3_I}, we can see that both conditions of Theorem 4 hold true, as follows:*

(a) Cyclic path $\pi_{c\ell} = ((4,4),(0,0)) \xrightarrow{f\epsilon} ((5,4),(1,0)) \xrightarrow{cc} ((4,4),(0,0))$ is both $+$ve-part fault-free and $-$ve-vocal, since $\pi = ((4,4),(0,0)),((5,4),(1,0)),((4,4),(0,0))$, $w_+(\pi_{c\ell}) = 1$ and for $(x_+, x_-) = (5,4), \delta(4,c) = 4$, with $c \in \Sigma_o$;

(b) Path $\pi' = ((1,1),(0,0)) \xrightarrow{\epsilon f} ((1,2),(0,1)) \xrightarrow{aa} ((3,3),(0,0))$ is a $-$ve-path since $w(\pi') = 0 - 1 = -1$, that starts at the initial state of G_{3_I} and is connected to cyclic path $\pi'_{c\ell} = ((3,3),(0,0)) \xrightarrow{cc} ((3,3),(0,0))$ that is both fault-free and $-$ve-vocal, since $\delta(3,f)$ is not defined and $\delta(3,c) = 3$, with $c \in \Sigma_o$.

Therefore, the language generated by G_3 is not $\forall\kappa$-diagnosable.

Remark 6

- *According to Zhou and Kumar (2009), the indistinguishable-pair automaton can also be used to check $U[1,\infty]$-diagnosability. The verification algorithm proposed in Zhou and Kumar (2009) has proved to have lower complexity than those proposed in Jiang et al. (2003), Yoo and Garcia (2009), and Yoo and Garcia (2004).*
- *For system models that fail the $\forall\kappa$-diagnosability test, Zhou and Kumar (2009) also presented an algorithm to compute the set of fault occurrence indices κ for which the system is not κ-diagnosable.*

Remark 7 (Complexity analysis (Zhou and Kumar, 2009)) *The number of states and transitions in the indistinguishable-pair automaton I are, in the worst case, $2|X|^2$ and $2|X|^2 \times |\Sigma|^2$. The verification algorithm is based on the search of strongly connected components and the computation of the shortest paths, which can be performed in $\mathcal{O}(|X|^2)$ and $\mathcal{O}(|X|^3 \times |\Sigma|^2)$ respectively. Thus, the overall complexity to check $\forall\kappa$-diagnosability is $\mathcal{O}(|X|^3 \times |\Sigma|^2)$, with $|X|$ and $|\Sigma|$ being respectively the numbers of states and events in the system model to diagnose.*

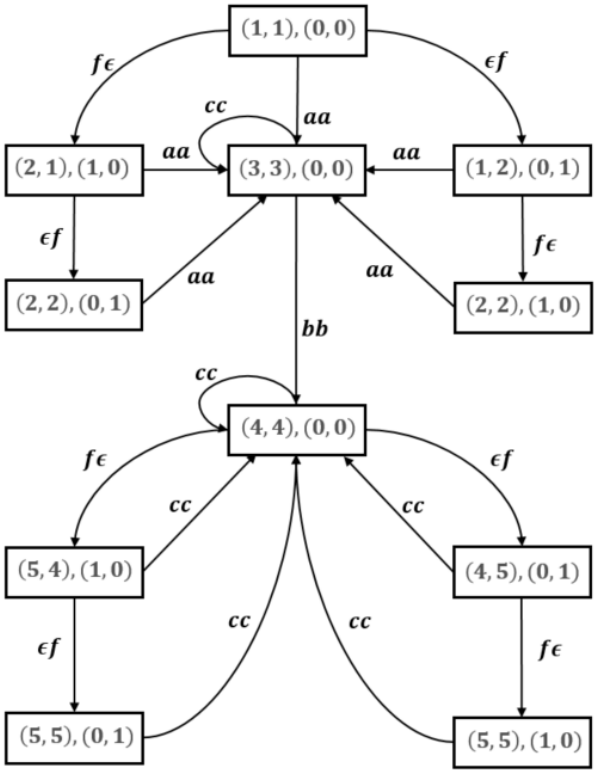


Fig. 9 The IPA G_{3_I} associated with system model G_3 of Example 7.

4.2 Verification of fault detection-based diagnosability notions

The verification of fault detection properties is carried out by using variants of diagnoser, twin-plant, and/or verifier automata. Thus, besides Assumption **A2**, the following assumptions are also required[4]:

A3. There is no cycle of unobservable events in the system model G;
A4. Each fault event f has its corresponding reset event r.
A5. There exists at least one observable event between the occurrence of a fault event f and its corresponding reset event r and also between the occurrence of a reset event r and a new occurrence of its corresponding fault event f.
A6. Each occurrence of fault event f is followed later on by the occurrence of its corresponding reset event r within a finite delay, and vice-versa, *i.e.*, each occurrence of a reset event r is followed later on by a new occurrence of the corresponding fault event f within a finite delay.

It is worth noticing that Assumption **A3** is the usual assumption in fault diagnosis of permanent faults, and Assumption **A6** implies that the fault and reset events

[4] Some assumptions can be relaxed for some approaches. When it is the case, it will be indicated explicitly.

occur with some regularity (pseudo-periodicity). These notions are called Σ_f-recurrence and Σ_r-recurrence respectively (Contant et al., 2004).

Property 8 *Under assumption **A6**, WF-diagnosability and WR-diagnosability are equivalent.*

Regarding Property 8, it is worth noticing that Assumptions **A1**–**A5** are not required.

4.2.1 Verification of fault detection-based diagnosability notions using diagnosers

- *Verification of WF-, WR-, SF-, and SR-diagnosability notions*

Contant et al. (2004), Contant (2005) and Contant et al. (2002) proposed an extension of the diagnoser approach firstly introduced in Sampath et al. (1995) by modifying its structure to deal with intermittent faults. The proposed diagnoser for a model $G = (X, \Sigma, \delta, x_0)$ is a deterministic FSA $G_d = (\mathcal{Q}, \Sigma_d, \delta_d, q_0)$ associated with a tagging function $Diag : X_o \to 2^{\Delta}$, where $\Delta = \{N, F, R\}$, with N standing for normal, F for faulty, and R for recovered, and $X_o = \{x_0\} \cup \{x \in X : \exists((x', \sigma) \in X \times \Sigma_o)[x \in \delta(x', \sigma)]\}$ is the finite set of reached state after the occurrence of an observable event. In addition, $\mathcal{Q} \subseteq 2^{(X \times \Delta)}$, $\Sigma_d = \Sigma_o$, $q_0 = (x_0, N)$, and $\delta_d : \mathcal{Q} \times \Sigma_d \to \mathcal{Q}$, where δ_d is defined as follows: given two states $q_1, q_2 \in \mathcal{Q}$, then $q_2 = \delta_d(q_1, \sigma) \Leftrightarrow \forall(x_2, l_2) \in q_2, (\exists u\sigma \in \Sigma_u^* \Sigma_o)(\exists(x_1, l_1) \in q_1) : x_2 = \delta(x_1, \sigma)$, with $l_1, l_2 \in \{N, F, R\}$ being the labels associated with each state according to the tagging function $Diag$. Notice that each state $q \in \mathcal{Q}$ has the form $q = \{(x_1, \ell_1), \ldots, (x_n, \ell_n)\}$, with $x_i \in X_o$ and $\ell_i \in \Delta$. If $\forall i = 1, \ldots, n$, $\ell_i = N$ (resp. $\ell_i = F$, $\ell_i = R$) then, diagnoser state q is said to be N-certain (resp. F-certain, R-certain), otherwise, it is an *uncertain* state.

Based on G_d, Contant et al. (2004) proposed necessary and sufficient conditions for WF-, WR-, SF-, and SR-diagnosabilities, defined according to Definitions 6, 7, 8, and 9, using the notion of indeterminate cycles. In this regard, a cycle cl in diagnoser G_d is a WF-indeterminate cycle if the following two conditions are satisfied: *(i)* no state in cl is F-certain, and; *(ii)* there exist, at least, two cycles cl_1 and cl_2 in the system model G consistent[5] with cycle cl, such that one cycle is formed with normal states only, and the other one contains states with labels F and/or R.

The notion of WF-indeterminate cycle is crucial, and leads to the following necessary and sufficient condition for WF-diagnosability.

Theorem 5 (Contant et al., 2004) *Under assumptions **A2**–**A6**, a system model G is WF-diagnosable w.r.t. P, Σ_f, Σ_r if, and only if, diagnoser G_d has no WF-indeterminate cycle.*

As discussed in Section 3.2, SF-diagnosability implies WF-diagnosability, and, thus, the necessary and sufficient condition for WF-diagnosability is a necessary condition for SF-diagnosability. Nevertheless, Contant et al. (2004) has presented

[5] Given a cycle cl in the diagnoser, we say that two cycles cl_1 and cl_2 in the system model are consistent with cl if event-sequences s_1 and s_2 associated with cl_1 and cl_2 respectively, and the event-sequence s associated with cl, share the same observation, i.e., $P(s_1) = P(s_2) = s_{cl}$.

a necessary and sufficient condition for SF-diagnosability based on the notion of SF-indeterminate cycles. According to Contant et al. (2004), an SF-indeterminate cycle is a cycle cl in diagnoser G_d (with s_{cl} its corresponding event-sequence), composed of non F-certain states only, and for which there exists a corresponding cycle cl_1 in G (with s_1 its corresponding event-sequence, and $P(s_1) = s_{cl}$) such that cl_1 has states with labels F and/or R.

Theorem 6 *(Contant et al., 2004) Under assumptions **A2**–**A6**, the language generated by a system model G is SF-diagnosable w.r.t. P, Σ_f, Σ_r if, and only if, diagnoser G_d has no SF-indeterminate cycle.*

Remark 8 *Since WF- and WR-diagnosability (resp. SF- and SR-diagnosability) are dual properties, necessary and sufficient conditions such as those stated in Theorems 5 and 6 can be developed for WR- and SR-diagnosability, respectively, in the same manner.*

Remark 9 *(Computational Complexity (Contant et al., 2004; Rintanen et al., 2007)) The number of states and transitions in the diagnoser are at most $2^{|X|}$ and $2^{|X|} \times |\Sigma_o|$ respectively. Thus, the computational complexity for building the diagnoser is $\mathcal{O}(2^{|X|} \times |\Sigma_o|)$, with $|X|$ and $|\Sigma_o|$ being respectively the numbers of states and observable events in the system model. Generally, verification algorithms for checking diagnosability properties using diagnoser approaches are based on the search for cycles, which has a factorial computation complexity (Johnson, 1975), being therefore exponential. Recently, Viana et al. (2015) and Viana and Basilio (2019) proposed a verification technique (which has a linear complexity) based on the search for strongly connected components in a diagnoser-like automaton.*

Example 8 *Figure 10 depicts diagnoser G_d, built in accordance with Contant et al. (2004) that corresponds to system model G shown in Figure 4a and considered in Example 3. Notice that diagnoser G_d has only one cycle, which is composed of states $\{5R, 9F\}$ and $\{7F, 11R\}$. Although this cycle is composed of non F-certain states, it is not a WF-indeterminate cycle, since there does not exist a corresponding normal cycle in the system model G (as shown in Example 3). Thus, according to Theorem 5, G is WF-diagnosable and thus, WR-diagnosable (cf. Property 8). In contrast, the cycle in G_d is SF-indeterminate, since it contains no F-certain states and, in addition, there exists a corresponding cycle in G which has experienced as least one fault occurrence (cf. Example 3). Consequently, G is not SF-diagnosable. Using the same reasoning, we can show that G is not SR-diagnosable either. Finally, since G is not SF-diagnosable (resp. not SR-diagnosable), then according to the relationships between the fault detection properties summarized in Property 7, G is not F_r-diagnosable (resp. not R_f-diagnosable).*

We will now make a brief account of related works to Contant et al. (2004). We start with the work by Correcher et al. (2003), where it is proposed a strategy to diagnose intermittent faults in industrial processes. The approach is applied to a classic pump/valve case-study (Sampath et al., 1996b), whose simulation is performed with Matlab, employing Simulink, for modeling the continuous behavior, and Stateflow, for the DES diagnoser. Biswas (2012) considers the verification of intermittent fault diagnosability notions in a state-based diagnosis framework, using the normalization setting. Two diagnosability

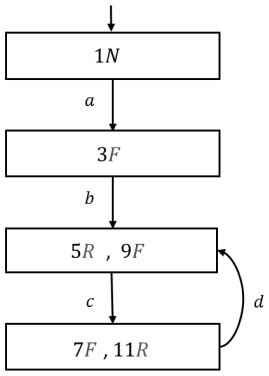


Fig. 10 Diagnoser G_d corresponding to the model of Example 3.

notions, which correspond to SF- and SR-diagnosabilities, have been addressed, being the approach proposed to analyze such notions an extension of the state-based diagnoser introduced in Hashtrudi Zad et al. (2003). For each notion of diagnosability, a necessary and sufficient conditions were established by Biswas (2012). Carvalho et al. (2012) investigate the problem of robust diagnosability against intermittent sensor faults assuming that either some sensors do not operate properly all the time or some observed events may not reach the diagnoser, *i.e.*, temporary loss of event observation may take place. Such a problem is formalized as an intermittent fault diagnosis one and a necessary and sufficient condition for robust diagnosability is presented. More recently, Carvalho et al. (2017) addressed the problem of diagnosing intermittent sensor faults in an event-based diagnosis framework within the recovery setting. They have modified the model of intermittent loss of observation to account for sensor malfunction only. Then, the problem of detecting intermittent sensor faults is transformed into a problem of diagnosing intermittent faults, in the same sense as in Contant et al. (2004). A diagnoser similar to the one developed in Cassandras and Lafortune (2008) is used in Carvalho et al. (2017) to check some diagnosability notions, equivalent (under some modeling restrictions) to WF- and WR-diagnosabilities. Using such a diagnoser, the assumption that no cycle involving only unobservable events exists in the system model can be relaxed. It is worth remarking that the necessary and sufficient conditions developed in Carvalho et al. (2017) consider both indeterminate observed cycles (equivalent to those in Contant et al. (2004)) and indeterminate hidden cycles, *i.e.*, cycles of states connected by unobservable events only. Finally, Boussif and Ghazel (2017) proposed a variant of the diagnoser presented in Cassandras and Lafortune (2008) in order to perform the verification of the above-mentioned diagnosability notions, using an event-based diagnosis framework within a recovery setting. The main idea behind the variant diagnoser is to separate normal, faulty and recovered states in each diagnoser node. By exploiting some features of the new diagnoser structure, necessary and sufficient conditions for checking WF-, WR-, SF- and SR-diagnosability notions are presented.

- *Verification of F_r-diagnosability*

Regarding the verification of F_r-diagnosability, Fabre et al. (2018) approach this problem in a normalization setting within state-based framework. Applying the same idea as in Viana et al. (2015), the approach proposed in Fabre et al. (2018) is based on the construction of an augmented diagnoser, G_a which is a parallel composition between the labeled system model and its diagnoser.

Formally, given a system model $G = (X, \Sigma, \delta, x_0)$ and its corresponding diagnoser $G_d = (\mathcal{Q}, \Sigma_d, \delta_d, q_0)$[6], we first compute the labeled system model G_ℓ as $G_\ell = G \parallel \Omega = (X_\ell, \Sigma, \delta_\ell, x_{\ell 0})$, with $X_\ell = X \times \{N, F, R\}$, $X_{\ell 0} = (x_0, N)$, and $\delta_\ell : X_\ell \times \Sigma \to X_\ell$. We then compute augmented diagnoser $G_a = G_\ell \parallel G_d = (X_a, \Sigma, \delta_a, x_{a0})$, with $X_a = X_\ell \times \mathcal{Q}$, $x_{a0} = (x_{\ell 0}, q_0)$, and $\delta_a : X_a \times \Sigma \to X_a$.

In order to simplify notation, let us consider that labels N and R are indistinguishable and denoted by $\bar{F}$. As a result, each state of the augmented diagnoser will have components in $\{F, \bar{F}\} \times \{F, \bar{F}, U\}$, where U represents an uncertain diagnoser state, *i.e.*, a diagnoser state that is neither F-certain nor $\bar{F}$-certain. The necessary and sufficient condition for F_r-diagnosability is based on the notion of minimally faulty states: a state $x_a = (x_\ell, q)$ in the augmented diagnoser G_a is said to be minimally faulty if $x_\ell = F$ is directly reachable from a state $x'_a = (x'_\ell, q')$ where $x'_\ell = \bar{F}$.

Theorem 7 (Fabre et al., 2018) *Under assumptions **A2**–**A5**, the language generated by a deterministic automaton G is not F_r-diagnosable if, and only if, there exists in the augmented diagnoser a reachable minimally faulty state (x, q) of type $(F, (\bar{F}))$ or $(F, (U))$ and either:*

1. *there exists a state (x', q') of type $(\bar{F}, (\bar{F}))$ or $(\bar{F}, (U))$, or*
2. *there exists a cycle composed exclusively of $(F, (U))$ states*

that is reachable from (x, q) through a (possibly empty) sequence of $(F, (\bar{F}))$ states followed by a sequence of $(F, (U))$ states.

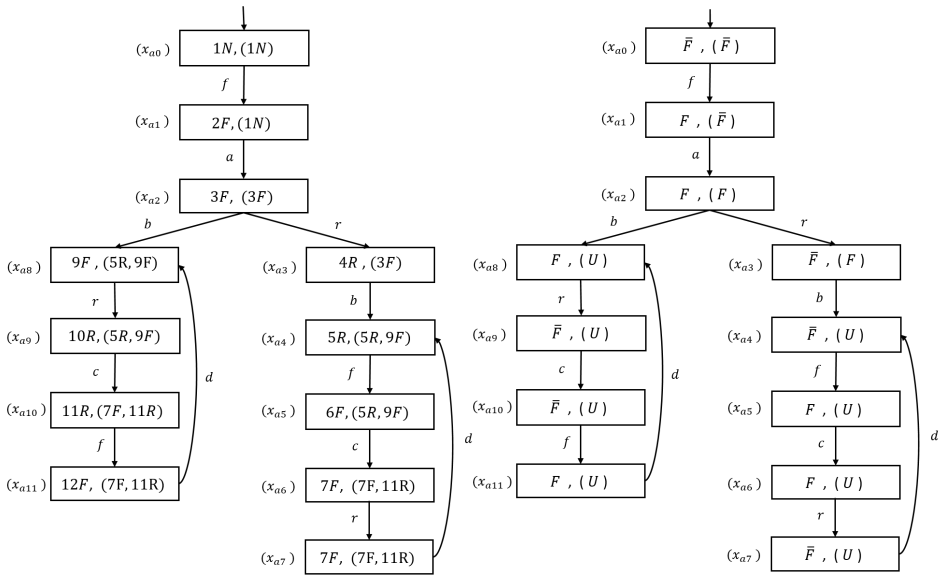
Theorem 7 provides the two conditions that leads to violation of F_r-diagnosability. The first condition accounts for the existence of two finite event-sequences s_1 and s_2 in the system model such that s_1 exhibits a faulty behavior and s_2 exhibits either a normal or a recovered behavior, meaning that the diagnoser was not able to identify that the system has recovered from the fault, whereas the second condition considers cyclic event-sequences.

Remark 10 *As shown in Viana and Basilio (2019), where a similar structure is used to perform diagnosability of permanent fault verification, there is no need to search for cycles that satisfy Condition (2) of Theorem 7. In this regard, Condition (2) could be replaced by the search of non-trivial strongly connected components that have states whose second elements are labeled by U and the first elements have both F and $\bar{F}$ labels in different states.*

Example 9 *Let us consider again system model G introduced in Example 3 and depicted in Figure 4(a). The labeled system model G_ℓ is depicted in Figure 4(b), and the augmented diagnoser G_a is depicted in Figure 11(a). An abstracted version of G_a*

[6] The diagnoser computation is presented at the beginning of this section (cf. Section 4.2.1).

showing only the label elements associated with each state is depicted in Figure 11(b). Notice that the augmented diagnoser state $x_{a_{11}} = (12F, (7F, 11R))$ *is a minimally faulty state since it is of type* $(F, (U))$*, and it is directly reachable from state* $x_{a_{10}} = (11R, (7F, 11R)))$ *which is of type* $(\bar{F}, (U))$*. In addition, notice that there exists state* $x_{a_9} = (10R, (4R, 9F)$ *of type* $(\bar{F}, (U))$ *which is reachable from state* $x_{a_{11}}$ *through state* $x_{a_8} = (9F, (5R, 9F))$*, which is of type* $(F, (U))$*. Hence, state sequence* $x_{a_{11}}, x_{a_8}, x_{a_9}$ *satisfies condition (1) of Theorem 7, and, thus, system model* G *is non* F_r*-diagnosable.*



(a) The augmented diagnoser G_a **(b)** An abstraction version of G_a

Fig. 11 Augmented diagnoser G_a of Example 9 and its abstraction version.

Remark 11 (Computational Complexity (Fabre et al., 2018)) *The number of states and transitions of the augmented diagnoser are at most* $2^{|X|} \times |X|$ *and* $2^{|X|} \times |X| \times \Sigma$ *transitions, respectively. In addition, as proved in Fabre et al. (2018), analyzing* F_r*-diagnosability of a system model* G *is a PSPACE-complete problem, while deciding the* SF*-diagnosability is PSPACE-hard problem.*

Fabre et al. (2018) have also addressed the run-time fault counting issue, *i.e.*, for a given event-sequence, determine how many times the faults have occurred. Firstly, they show that F_r-diagnosability notion is not strong enough to correctly count fault occurrences. Indeed, associating a fault counter to the computed diagnoser does not ensure a correct counting of faults. Also, Fabre et al. prove that the problem of deciding if an automaton is fault countable is an NLOGSPACE problem. This result is stated without providing the fault counter construction; although they provide a run-time function that can be used to count the number of faults the diagnoser is able to detect.

Remark 12 *(Online diagnosis) It is worth noticing that advantage of using the diagnoser-based approaches comes from the fact that for diagnosable languages, the constructed diagnoser can also be used to perform online diagnosis, as detailed in Sampath et al. (1995).*

4.2.2 Verification of intermittent fault diagnosability notions using the twin-plants

The verification of weak and strong diagnosability notions have also been addressed using the twin-plant approach (Boussif and Ghazel, 2016a, 2019), where an extension of the twin-plant developed in Jiang et al. (2001) has been presented (by modifying its structure) in order to deal with the intermittent faults.

The twin-plant of a model G is a non-deterministic automaton $\mathcal{P} = (\mathcal{Q}, \Sigma_o, \gamma, q_0)$, where $\mathcal{Q} \subseteq X_o \times X_o$, with $X_o = \{x_0\} \cup \{x \in X : (\exists (x', \sigma) \in X \times \Sigma_o)[x \in \delta(x', \sigma)\}$, γ is the transition function defined as follows: $\gamma : \mathcal{Q} \times \Sigma_o \to 2^{\mathcal{Q}}$, for $q = (x_1, x_2)$, $q' = (x'_1, x'_2) \in \mathcal{Q}$, $q' \in \gamma(q, \sigma)$ if, and only if, $x'_1 = \delta(x_1, u_1\sigma)$ and $x'_2 = \delta(x_2, u_2\sigma)$, for some $u_1, u_2 \in \Sigma_u^*$, and $q_0 = (x_0, x_0) \in \mathcal{Q}$. Notice that X is the finite set of states of the system model (as defined in Section 2), and $X_o \subseteq X$ is the finite set of states in X which are (directly) reached by the occurrence of an observable event. For example, $X_o = \{1, 3, 9, 11, 5, 7\}$ for model G in Figure 4a.

The fault propagation is preserved in the twin-plant using the *fault-assignment* function $\Psi : \mathcal{Q} \to \{N, F, R\} \times \{N, F, R\}$, allowing different types of states to be distinguished in the twin-plant, as follows:

- N-state (resp. F-state, R-state) is a state $q = (x, x') \in \mathcal{Q}$, such that $\Psi(q) = (N, N)$ (resp. $\Psi(q) = (F, F)$, $\Psi(q) = (R, R)$);
- NF-state (resp. NR-state) is a state $q = (x, x') \in \mathcal{Q}$, such that $\Psi(q) = (N, F)$ (resp. $\Psi(q) = (N, R)$). FN and RN-state are defined similarly;
- $N1$-state is a state $q = (x, x') \in \mathcal{Q}$, such that $\Psi(q) = (N, \triangle)$ with $\triangle \in \{N, F, R\}$;
- non-N-state (resp. non-F-state, non-R-state) is a state which is not an N-state (resp. F-state, R-state).

The twin-plant structure is symmetric in the sense that a path containing FN-states has its symmetric path which contains the symmetric NF-states, and vice versa.

Checking WF-diagnosability using the twin-plant consists in seeking for F-confused cycles. An F-confused cycle in the twin-plant corresponds to two cycles in the original model G, whose corresponding event-sequence have the same observable projection, such that one event-sequence has no fault event (a fault-free cycle) while the second one contains at least one fault event (which is ensured by the existence of an NF-state in the cycle). Formally, an F-confused cycle is defined as follows.

Definition 11 *(F-confused cycles)* An F-confused cycle is a cycle $cl = (q_1, q_2, \ldots, q_n, q_{n+1} = q_1)$ of the twin-plant, such that $\forall\, 1 \leq i \leq n$, q_i is an $N1$-state, and $\exists\, 1 \leq j \leq n$, such that q_j is an NF-state.

Theorem 8 (Boussif and Ghazel, 2019) Under assumptions **A2**–**A6**, *a language $L(G)$ generated by an automaton G is WF-diagnosable w.r.t. P, Σ_f, and Σ_r if, and only if, there exists no F-confused cycle in twin-plant $\mathcal{P}$ associated with G.*

It is worth noticing that since WF-diagnosability does not ensure SF-diagnosability, the necessary and sufficient condition for WF-diagnosability given in Theorem 8 represents only a necessary condition for the SF-diagnosability. Indeed, SF-diagnosability cannot be checked by only seeking some F-confused cycles, as in the case of permanent faults and WF-diagnosability since it cannot be characterized by paths of the twin-plant taken individually (Fabre et al., 2016, 2018).

The necessary and sufficient conditions for SF- and SR-diagnosabilities are based on the notion of generated prime state-path in the twin-plant, which has been inspired by Zhou and Kumar (2009) and Basilio et al. (2012)). We start by defining state-path in the twin plant $\mathcal{P}$ as a sequence of states $(q_1, q_2, \ldots, q_n)$ such that $\forall q_i$, $i = 1, 2, \ldots, n-1$, $\exists \sigma_i \in \Sigma_o$ such that $q_{i+1} = \gamma(q_i, \sigma_i)$. A state-path that starts at the initial state q_0 is called a generated state-path, a state-path that does not include any state that is visited twice, *i.e.*, $\forall i, j \in \{1, \cdots, n\}$ and $i \neq j$, we have $q_i \neq q_j$, is called elementary state-path, and an elementary cyclic state-path is a state cycle $(q_1, q_2, \ldots, q_n)$ such that $q_i \neq q_j$, $\forall i, j \in \{1, \cdots, n-1\}$ with $i \neq j$; and $q_1 = q_n$. Finally, a generated prime state-path is a generated state path $\wp = \wp' cl$ that is composed of an elementary state-path $\wp'$ and an elementary cyclic state-path cl. Given observable event-sequence $s = \sigma_0 \sigma_1 \ldots \sigma_n \in \Sigma_o^*$, we can define an associated set $\Pi(s)$ of all generated prime state-paths corresponding to s, being formally defined as follows:

$$\Pi(s) = \{\wp = (q_0, q_1, \ldots, q_n) \in \mathcal{P} : (\wp \text{ is a generated prime state-path}) \wedge (q_{i+1} \in \gamma(q_i, \sigma_i), 0 \leq i < n,) \wedge ((q_j \in \gamma(q_n, \sigma_n), \text{for some } 0 \leq j \leq n)\}.$$

A twin-plant based necessary and sufficient condition for SF-diagnosability (resp. SR-diagnosability) has been proposed in Boussif and Ghazel (2019), which is based on the concept of F-Interception condition, defined formally as follows.

Definition 12 (F-*Interception* condition) Let $s \in \Sigma_o^*$ be an observable event-trace in $\mathcal{P}$. Then, we say that s satisfies the *F-Interception* if $\exists k \in \mathbb{N}$, $\forall \wp = \wp' cl = (q_0, q_1, \ldots, q_n) \in \Pi(s)$: $q_k \in cl$, and q_k is an F-state.

According to Definition 12, the F-Interception condition ensures that, after a finite delay, all generated prime state-paths that correspond to s reach F-states, in their corresponding elementary cycles at the same time (*i.e.*, after $k-1$ observations). The necessary and sufficient condition for SF-diagnosability is given by the following theorem.

Theorem 9 *(Boussif and Ghazel, 2019)— Under assumptions* ***A2****–****A6****, a language $L(G)$ is SF-diagnosable w.r.t. P, Σ_f, and Σ_r iff the F-Interception condition is satisfied by each event-trace s obtained from twin-plant $\mathcal{P}$ constructed from model G.*

Remark 13 *(Computational Complexity (Jiang et al., 2001; Boussif, 2016)) The number of states and transitions of the twin-plant are at most $4 \times |X|^2$ and $8 \times |X|^4 \times |\Sigma_o|$, respectively. The verification of WF-diagnosability can be performed with linear complexity with respect to the twin-plant size (using the procedure presented in Jiang et al. (2001). Thus, the overall complexity of checking WF-diagnosability using the twin-plant approach is $\mathcal{O}(|X|^4 \times |\Sigma_o|)$, which is polynomial in the number of states of the model. The verification of SF-diagnosability involves, in the worst case, the search of all elementary cycles in the twin-plant. As pointed out before, such a procedure has a factorial computation complexity in the worst case (Johnson, 1975).*

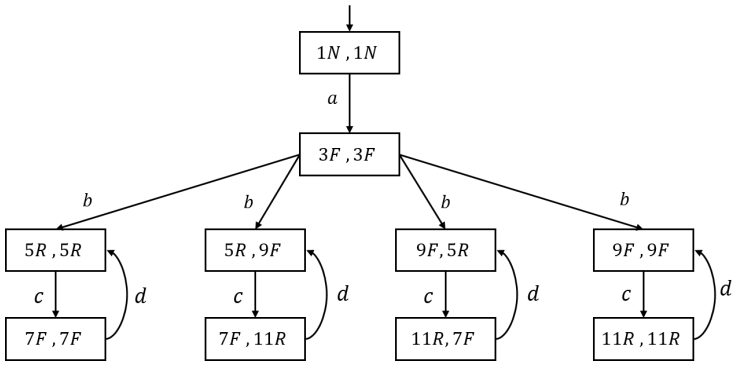


Fig. 12 The twin-plant $\mathcal{P}$ corresponding to model G of Example 3 (Figure 4).

Example 10 *Figure 12 depicts the twin-plant $\mathcal{P}$ corresponding to model G shown in figure 4a and considered in Example 3. An inspection of Figure 12 reviews that G is WF-diagnosable since there is no F-confused cycle in $\mathcal{P}$. However, the model is non SF-diagnosable since, as discussed in Example 3, event-sequence $ab(cd)^*$ does not satisfy the F-Interception condition.*

4.2.3 Analysis of fault detection properties using the verifier-based approach

Carvalho et al. (2017) discussed the intermittent sensor fault detection problem, addressing three cases regarding sensor faults, as follows: (i) when the sensor under consideration never recovers after it fails, (ii) when the sensor never fails again after the last time it recovers from the failure, and (iii) when the sensor fails at some point and can or cannot recover from the fail. Such configurations can be viewed as weak diagnosability properties (under some restrictions). The authors discuss the analysis of such properties using a verifer-based approach.

The verification process is based on the construction of three verifier automata V_{NF}, V_{NR}, and V_{FR} whose constructions are carried out according to Algorithm 1 of Carvalho et al. (2017). In this context, the analysis of weak diagnosability properties can be performed using verifiers V_{NF} and V_{NR}. To this end, three sub-automata need to be computed from the augmented system model $G_\ell = G||\Omega$ (with Ω being the label automaton illustrated in Figure 1a), as follows: (1) G_N, the sub-automaton that depicts only the normal behavior of G_ℓ; (2) G_F, the sub-automaton that depicts the faulty behavior of G_ℓ, and; (3) G_R, the sub-automaton that depicts the recovered behavior of G_ℓ. Verifier $V_{NF} = G_N^{\mathcal{R}}||G_F$ (resp. $V_{NR} = G_N^{\mathcal{R}}||G_R$), where $G_N^{\mathcal{R}}$ is identical to G_N except that the unobservable events of G_N are renamed in order to make them private in the parallel composition, is then computed. Notice that by considering Assumptions **A1**–**A4**, we have $V_{NF} = V_{NR}$. Checking WF-diagnosability using verifier V_{NF} consists in seeking bad cycles, which are equivalent to the F-confused cycles in the twin-plant, and correspond to cycles $cl = (q_1, q_2, \ldots, q_n, q_{n+1} = q_1)$ in verifier V_{NF}, such that q_i, $i = 1, 2, \ldots, n$ is an $N1$-state, and q_j, $j = 1, 2, \ldots, n$, is an NF-state.

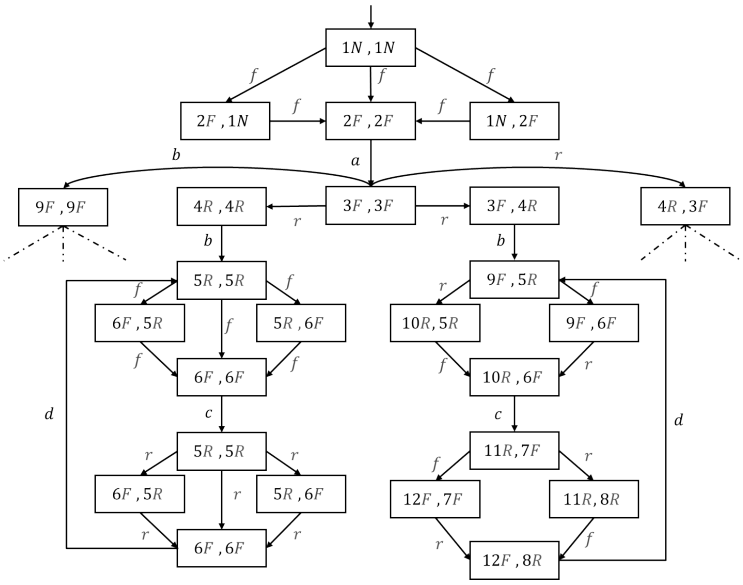


Fig. 13 Verifier V_{NF} corresponding to model G of Example 3 (Figure 4).

Theorem 10 *Under assumptions **A2**–**A6**, a language $L(G)$ is WF-diagnosable (resp. WR-diagnosable) w.r.t. P, Σ_f, and Σ_r iff no F-confused (resp. R-confused) cycle exists in verifier V_{NF}.*

Example 11 *Figure 13 depicts a relevant part of verifier V_{NF} corresponding to model G depicted in Figure 4a and considered in Example 3. We can infer that V_{NF} is WF- and WR-diagnosable since no F-confused cycle exists in V_{NF}.*

Remark 14 *(Complexity analysis (Moreira et al., 2011; Carvalho et al., 2017)) The number of states and transitions in the verifier automaton of Moreira et al. (2011) are at most $2|X|^2$ and $2|X|^2 \times |\Sigma|$, respectively. The verification of weak diagnosability is based on the search for strongly connected components, which is performed with linear complexity. Thus, the overall complexity for checking the weak diagnosability properties using the verifier approaches is $\mathcal{O}(|X|^2 \times |\Sigma|)$, which is polynomial in the number of states in the model.*

To conclude this section, we summarize in Table 1 the general results in the literature up to date regarding the verification of intermittent fault diagnosability properties and their corresponding complexity.

Table 1 A summary regarding the verification of intermittent fault diagnosability properties

Problem	**Technique**	**Complexity order**	**References**
κ-, $[1,\kappa]$-diagnosability	Transition graph	$\mathcal{O}(\|X\|^4)$	(Jiang et al., 2003)
U$[1,\infty]$-diagnosability	Transition graph	$\mathcal{O}(\|X\|^6)$	(Jiang et al., 2003)
	Weighting graph	$\mathcal{O}(min(\|X\|^3 \times \|\Sigma\|^2, \|X\|^5))$	(Yoo and Garcia, 2009, 2004)
	IPA	$\mathcal{O}(\|X\|^3 \times \|\Sigma\|^2)$	(Zhou and Kumar, 2009)
NU$[1,\infty]$-diagnosability	Weighting graph	$\mathcal{O}(min(\|X\|^3 \times \|\Sigma\|^2, \|X\|^5))$	(Yoo and Garcia, 2009, 2004)
$\forall\kappa$-diagnosability	IPA	$\mathcal{O}(\|X\|^3 \times \|\Sigma\|^2)$	(Zhou and Kumar, 2009)
WF-diagnosability	Diagnoser	$\mathcal{O}(2^{\|X\|} \times \|\Sigma_o\|)$	(Contant et al., 2004; Carvalho et al., 2013), (Boussif and Ghazel, 2017; Carvalho et al., 2017)
	Twin-plant	$\mathcal{O}(\|X\|^4 \times \|\Sigma_o\|)$	(Jiang et al., 2001; Boussif and Ghazel, 2019)
	Verifier	$\mathcal{O}(\|X\|^2 \times \|\Sigma\|)$	(Moreira et al., 2011; Carvalho et al., 2017)
SF-diagnosability	Diagnoser	$\mathcal{O}(2^{\|X\|} \times \|\Sigma_o\|)$	(Contant et al., 2004; Boussif and Ghazel, 2017)
F_r-diagnosability	Augmented diagnoser	$\mathcal{O}(2^{\|X\|} \times \|X\| \times \Sigma)$	(Fabre et al., 2016, 2018)

5 Other approaches that deal with intermittent fault diagnosis

Besides the works on intermittent fault diagnosis described in the previous sections, where automaton formalism is used, the intermittent fault diagnosis problem has also been addressed using different frameworks, such as supervision pattern, temporal logic, discrimination between intermittent and permanent faults, and fault free models, and different modeling formalism, such as Petri nets and stochastic automata. In this section, we review the main contributions within these approaches.

5.1 Intermittent fault diagnosis as supervision pattern diagnosis

A supervision pattern is a formal model (automaton, Petri net, etc.) whose language is the set of trajectories to be diagnosed (Jéron et al., 2006; Gougam et al., 2013a). It is general enough to cover a broad class of diagnosis objectives found in the literature, *e.g.*, diagnosis of multiple and repeated faults, sequences of significant events, repair of faults, etc. For example, the label automaton of Figure 1a, used to determine the system status w.r.t. the occurrence of faults and their recovery (normal, faulty, or recovered), can be seen as a supervision pattern, in the sense that it is a formal model that characterizes a specific behavior of the system as a partial order of observable events or states.

Supervision patterns extend the expressiveness of faulty models by introducing more complex faulty behaviors (Gougam et al., 2013b), being useful in the generalization of the diagnosis definitions and to clarify the separation between the diagnosis objectives and the system specifications. In this regard, the results obtained from the diagnosis task can be simply re-utilized to deal with similar diagnosis issues, due to their generic nature (Lamperti and Zanella, 2004).

The supervision pattern diagnosis problem is generally achieved based on the matching between the real behavior of the system and the compiled faulty behavior (Zaytoon and Sayed-Mouchaweh, 2012). In the context of fault diagnosability, it can be formulated as follow: given a DES model and a supervision pattern, is the supervisor (or the diagnoser) always able to determine with certainty that some pattern has occurred or not in the system after observing a finite sequence of events?

Supervision pattern diagnosis of DES has been first addressed by Jéron et al. (2006). After that, further works followed. Ye et al. (2009), Yan et al. (2010) and Ye and Dague (2012) deal with the diagnosis of patterns in distributed DESs modeled by finite state automata. Gougam et al. (2014) discuss the discriminability[7] of supervision patterns in a Petri net framework (in that work, both the system model and patterns are Petri net models). The diagnosability of Petri net patterns has also been discussed in Gougam et al. (2013b) and Gougam et al. (2017). In Pencolé and Subias (2018), the diagnosis of patterns is formulated as a pattern matching problem, and to this end, they use bounded and labeled prioritized Petri nets and tackle this problem using model checking techniques.

It is worth remarking that the work by Jéron et al. (2006) remains the unique work that deals with intermittent fault diagnosability as a supervision pattern

[7] Differently from diagnosability, discriminability is the possibility to detect the exclusive occurrence of a particular behavior of interest.

diagnosability problem. Indeed, Jéron et al. (2006) have proposed two patterns that model the faulty behaviours corresponding to k occurrences of a fault and the intermittent fault occurrence with a repair. In order to verify such patterns, Jéron et al. (2006) have used the twin-plant approach of Jiang et al. (2003), in which the synchronous product between the system model and the supervision pattern is used as the input of the twin-plant algorithm. It is worth remarking that, with slight modifications, the diagnoser and verifier approaches can also be used to check the diagnosability of supervision patterns.

5.2 Intermittent fault diagnosis using temporal logic specifications

Due to their expressiveness, temporal logics (Emerson et al., 1990) have been used for a long time in supervisory control of DES (Thistle and Wonham, 1986; Lin, 1991). Regarding fault diagnosis, temporal logics provide another way to specify fault properties. In addition to expressing fault event occurrences and reachability of faulty states, temporal logics can also be used to express complex types of fault properties such as the violation of liveness, safety, invariance, recurrence and stability properties (Jiang and Kumar, 2004).

In the context of temporal logic-based diagnosis, the occurrence of intermittent faults can be expressed using linear (LTL) or branching (CTL) temporal logic formulae. Therefore, analyzing intermittent fault related properties can be translated as model-checking problems, and tackled using the associated verification engines (Jiang and Kumar, 2006; Boussif and Ghazel, 2016b).

In Jiang and Kumar (2006), fault counting of repeated failures is discussed in a temporal logic framework. Notions of diagnosability and prediagnosability[8] for intermittent faults are formulated in a temporal logic setting. A polynomial test algorithm for prediagnosability verification is provided. The authors also discuss the various notions of diagnosability related to the multiplicity of fault occurrences, adapted from Jiang et al. (2003), in a linear-time temporal logic (LTL) setting.

In Boussif and Ghazel (2016a), a model-checking framework to deal with intermittent fault diagnosis, which is an extension of the practical verification approaches for analyzing diagnosability of permanent faults using model-checking (Cimatti et al., 2003; Boussif and Ghazel, 2015) is proposed. Firstly, the authors revise the weak intermittent fault diagnosability properties, i.e., WF- and WR-diagnosability, and then, necessary and sufficient conditions based on the twin-plant proposed in Boussif et al. (2016) are expressed as linear temporal logic (LTL) model-checking problems (Boussif and Ghazel, 2016b). A benchmark is used to illustrate the various steps and to assess the efficiency and the scalability of the approach. This technique has then been extended in Boussif and Ghazel (2018) to deal with F_r-diagnosability.

5.3 Discriminating intermittent faults from permanent faults

Almost all the works that approach the fault diagnosis problem in DES consider only one type of faults, namely, permanent, intermittent, or transient faults.

[8] Prediagnosability consists in detecting the occurrence of an indicator trace which ensures that the fault occurrence is inevitable.

However, real-life systems often exhibit more than one type of faults. Therefore, it is common for diagnosis systems to misjudge some types of faults or presume that all faults are of the same type.

A more general framework is proposed in Deng et al. (2014a), Deng et al. (2014b), and Deng et al. (2013), where fault models that include both permanent and intermittent faults are considered. In order to diagnose faults in such a setting, the authors propose an approach that first discriminates between the fault classes. The approach is based on an extension of the diagnoser proposed in Contant et al. (2004), where the label propagation function is modified in order to account for each fault class dynamic. The authors show that the fault types can be diagnosed (and discriminated) within bounded delay if the system is diagnosable with respect to each fault type. The approach is firstly discussed within ordinary automata (Deng et al., 2014a) and then extended to stochastic ones (Deng et al., 2014b, 2013).

5.4 Intermittent fault diagnosis using fault-free models

Fault diagnosis using fault-free models is based on the comparison between the actual system output and the model nominal output. The fault is detected if the observed behavior of the system cannot be reproduced by its model. In a series of works, Soldani et al. (2006, 2007a,b) discuss the detection and isolation of intermittent faults in a fault-free DES modeling setting in both automata (Soldani et al., 2007a) and Petri net (Soldani et al., 2007b). In those works, failures may imply either the occurrence (insertion) of spurious events or the lack of foreseen events. The proposed approach consists of three steps:

1. System modeling (offline). A model that expresses the nominal behavior of the system to be diagnosed is firstly constructed based on the design data whereby only the observable actions are represented (Soldani et al., 2006). The built models can be either automata or Petri nets.
2. Intermittent and fugitive fault detection (online). The fault detection process consists of comparing the observable event sequences issued by the system with the expected event sequences from the model. A faulty behavior is detected if there exists some inconsistency between the received event and the expected one (Soldani et al., 2006).
3. Online fault localization. Fault localization succeeds the detection process and consists of determining the events that are prospectively responsible for the fault. The technique is based on the construction of two diagnosers, one for localizing the missing events, and the other one for localizing the inserted events.

5.5 Intermittent fault diagnosis using Petri net as modeling formalism

Fault diagnosis of intermittent faults has been also investigated using Petri net as the modeling formalism. García et al. (2008b) proposed a new methodology to deal with fault diagnosis of both permanent and intermittent faults using colored Petri nets (CPNs), in which, in order to model the faults, the set of colored tokens

is divided in two subsets: the subset of normal tokens, representing the nominal dynamic behavior of subnets system, and the subset of tokens, representing the faulty behaviors to be diagnosed. Then, the so-called Fault Latent Nestling (FLN) Method (see García et al. (2008a)) is used to nestle faults into each place of the initial PN using a folding technique with CPNs and the characteristics of sensor readings to isolate faults in a specific place or detect them. Such a combination of CPN modeling and FLN method allows for efficiently tackling the combinatorial explosion that often arises when it comes to diagnose large systems. The proposed approach has been firstly applied on an academic example in García et al. (2008b), and then on a real application in Rodriguez et al. (2008), namely a wind turbine system. This approach has been recently extended to deal with intermittent fault diagnosis of hybrid colored Petri nets in Rodriguez-Urrego et al. (2015), and applied to a digital electronic module, namely an insulated-gate bipolar transistor (IGBT).

Recently, and with the same objective as García et al. (2008b), Trigos et al. (2016) proposed a PN fault diagnosis approach to deal with both permanent and intermittent faults, where Petri nets are used for depicting the system behavior and building the PN diagnoser using a data acquisition system. In practice, a fault diagnosis algorithm is designed to perform this task. The proposed approach has been firstly applied to a liquid packaging process (Martínez and Moreno, 2008) to deal with permanent faults, and afterwards to an unmanned aerial vehicle (Trigos et al., 2016) to deal with intermittent faults.

5.6 Intermittent fault diagnosis of stochastic models

A few works have been devoted to intermittent fault diagnosis of stochastic models. In Yoo and Garcia (2005), a counting strategy that accommodates stochastic automata was presented, which, strictly speaking, is deterministic in the sense that the discussed counting algorithm seeks the minimum count of the associated state estimate rather than using the probabilistic distribution of the state estimate of the stochastic automaton; essentially, it deals with possibility rather than probability. The work by Yoo and Garcia (2008) is an extension of previous works (Yoo and Garcia, 2009, 2004; Zhou and Kumar, 2009) in the framework of deterministic FSA, and attempts to fully utilize the probabilistic aspects of stochastic FSA in developing algorithms for event counting. Such an approach is based on updating the active counter information state sequentially with available observations. Deng et al. (2014b) and Deng et al. (2013) extend their previous work in Deng et al. (2014a) regarding the discrimination of intermittent faults from permanent ones to stochastic FSA, by enlarging the model presented in Sampath et al. (1995) to consider both permanent and intermittent faults. By assuming that environmental stress is the main cause of faults, the authors treat it as a fault event. Therefore, a stress level evaluation algorithm based on interval grey relational degree is developed to identify the fault events by computing the level of the correlative environmental stress. In Deng et al. (2014b), the notions of A- and AA-diagnosability of permanent faults for stochastic FSA (Thorsley and Teneketzis, 2005) were extended to deal with intermittent faults. As far as diagnosability analysis is concerned, the authors propose a diagnoser-approach with a probability matrix appended to each transition, which can be used to update the probability

distribution on the state estimate. With the knowledge of each state transition probability, it is possible to distinguish event-sequences or states that are more likely from those that are less probable to occur or achieve, respectively.

Using a different model formalism, Lefebvre and Leclercq (2011) presented an approach to deal with intermittent fault based on stochastic Petri net identification techniques. The authors proposed two learning algorithms to design and identify stochastic Petri net models (that are used as reference models for FDI) according to causal and temporal specifications. Then, an FDI algorithm to perform the online detection and isolation of intermittent faults (which are considered as behaviors that do not satisfy the causal or temporal specifications) is established.

6 Looking backwards to point towards the future

This section provides some remarks regarding the existing works that approach the intermittent fault diagnosis problem in DES and suggests some perspectives for further research.

According to the discussed literature review, it can be inferred that most of the contributions focus on the analysis of some diagnosability definition and the synthesis of diagnosers to perform online diagnosis. However, no works have discussed the complementary issues directly related to fault diagnosis, such as fault prediction, decentralized/modular diagnosis, sensor selection and dynamic sensor activation, robust diagnosis, active diagnosis and fault tolerant control, which are widely discussed in the case of permanent fault diagnosis (Zaytoon and Lafortune, 2013). Those are open problems that still need further investigation in the context of intermittent faults.

Deng et al. (2014a), Deng et al. (2014b), and Deng et al. (2013) have discussed the issue of discriminating intermittent faults from permanent ones, which is an interesting issue from system maintenance point of view. Indeed, maintenance actions could greatly differ according to the nature of failure (permanent or intermittent), and, consequently, maintenance costs could be reduced by avoiding unnecessary shutdown and repair operations (Deng et al., 2014a). In this regard, it would be interesting to also consider a correlated phenomenon, which is the evolution of intermittent faults to become permanent ones. In this case, intermittent faults can be viewed as a symptom of the degradation of some physical aspects in the system. As degradation increases, the rate and severity of intermittent symptoms may increase until the intermittent faults eventually become permanent (Syed et al., 2013). An intuitive way to deal with such a setting would be to consider the persistence feature of intermittent faults, *i.e.*, the frequency of occurrence. In this regard, a frequency threshold can be proposed to determine when an intermittent fault can be assimilated to a permanent one. Nevertheless, this issue still needs more investigation from both the theoretical and practical points of view.

As reported in Section 3, intermittent fault diagnosis of DES has been discussed according to two distinct points of view: fault counting and fault detection. Various definitions and different verification techniques to deal with these two problems have been proposed. However, no effective connection can be perceived between these two perspectives. We believe that these issues should be addressed more jointly. The relationships between the various properties need to be discussed,

and more expressive diagnosability definitions that address both fault counting and fault detection could be proposed. Practitioners would appreciate diagnosers that address both fault counting and fault detection requirements for monitoring and maintenance purposes. The works in Boussif and Ghazel (2018), Fabre et al. (2016), and Fabre et al. (2018) can be seen as first steps in this direction.

To conclude this section, notice that no useful connection has yet been made between the various diagnosability definitions discussed in the paper and the three indicators (duration, pseudo-periodicity, and the number of fault occurrences) presented in the introduction. This is so because, unlike in the permanent failure problem, where some works investigated fault diagnosis by also considering temporal aspects (see Zaytoon and Lafortune (2013) and, more recently, Viana and Basilio (2019) and Viana et al. (2019)), the majority of works that deal with intermittent fault diagnosis consider untimed DES models. Up to now, as far as intermittent fault is concerned, only the logical features of the system and fault dynamics are taken into account, *i.e.*, the logical order and the number of event occurrences, which is inadequate when it is necessary to consider real time aspects. However, as shown in Ghazel et al. (2009), temporal information has shown to be determinant in several diagnosis problems. Therefore, diagnosis of intermittent faults of discrete event systems with timing structure also appears as a promising research topic.

7 Conclusion

In this paper, we have provided a detailed review of the literature on intermittent fault diagnosis in DES modeled by automata. We also highlighted the main contributions to intermittent fault diagnosis by applying other frameworks, such as supervision pattern, temporal logic, discrimination between intermittent and permanent faults, and fault free models, and different modeling formalism, such as Petri nets and stochastic automata. As far as future research on intermittent fault diagnosis is concerned, we listed several open problems and proposed new research topics. We hope this review will serve as a helpful guide for future studies in the field and as a background for those who want to pursue some research in intermittent fault diagnosis.

References

Anderson RJ, Aylward SR (1993) Lab testing of neural networks for improved aircraft onboard-diagnostics on flight-ready hardware. Annual Reliability and Maintainability Symposium pp 404–410

Aydin I, Karaköse E, Karaköse M, Gençoğlu MT, Akın E (2013) A new computer vision approach for active pantograph control. IEEE International Symposium on Innovations in Intelligent Systems and Applications pp 1–5

Ball M, Hardie F (1969) Effects and detection of intermittent failures in digital systems. Proceedings of Computer Conference pp 329–335

Banerjee N, Khilar P (2010) Performance analysis of distributed intermittent fault diagnosis in wireless sensor networks using clustering. International Conference on Industrial and Information Systems pp 13–18

Basilio JC, Lima STS, Lafortune S, Moreira MV (2012) Computation of minimal event bases that ensure diagnosability. Discrete Event Dynamic Systems 22(3):249–292

Biswas S (2012) Diagnosability of discrete event systems for temporary failures. Computers & Electrical Engineering 38(6):1534–1549

Boussif A (2016) Contributions to fault diagnosis of discrete-event systems. PhD thesis, University of Lille - Sciences & Technologies

Boussif A, Ghazel M (2015) Diagnosability analysis of input/output discrete event system using model checking. 5^{th} IFAC International Workshop on Dependable Control of Discrete Systems 48(7):71–78

Boussif A, Ghazel M (2016a) Intermittent fault diagnosis of industrial systems in a model-checking framework. IEEE International Conference on Prognostics and Health Management pp 1–6

Boussif A, Ghazel M (2016b) Using model-checking techniques for diagnosability analysis of intermittent faults-a railway case study. Verification and Evaluation of Computer and Communication Systems pp 93–104

Boussif A, Ghazel M (2017) A diagnoser-based approach for intermittent fault diagnosis of discrete-event systems. American Control Conference pp 3860–3867

Boussif A, Ghazel M (2018) Formal verification of intermittent fault diagnosability of discrete-event systems using model-checking. International Journal of Critical Computer-Based Systems 8(2):193–213

Boussif A, Ghazel M (2019) Diagnosability analysis of intermittent faults in discrete event systems using a twin-plant structure. International Journal of Control, Automation and Systems (IJCAS) pp 1–14, DOI 10.1007/s12555-018-0682-9

Boussif A, Liu B, Ghazel M (2016) A twin-plant based approach for diagnosability analysis of intermittent failures. 13th International Workshop on Discrete Event Systems pp 237–244

Carvalho LK, Basilio JC, Moreira MV (2010) Robust diagnosability of discrete event systems subject to intermittent sensor failures. International Workshop on Discrete Event Systems pp 84–89

Carvalho LK, Basilio JC, Moreira MV (2012) Robust diagnosis of discrete event systems against intermittent loss of observations. Automatica 48(9):2068–2078

Carvalho LK, Basilio JC, Moreira MV, Clavijo LB (2013) Diagnosability of intermittent sensor faults in discrete event systems. American Control Conference pp 929–934

Carvalho LK, Moreira MV, Basilio JC (2017) Diagnosability of intermittent sensor faults in discrete event systems. Automatica 79:315–325

Cassandras C, Lafortune S (2008) Introduction to discrete event systems, 2^{nd} Edition. Springer Science

Chang JTY, McCluskey EJ (1997) Detecting bridging faults in dynamic CMOS circuits. IEEE International Workshop on IDDQ Testing pp 106–109

Cherkassky BV, Goldberg AV, Radzik T (1996) Shortest paths algorithms: Theory and experimental evaluation. Mathematical Programming 73(2):129–174

Cimatti A, Pecheur C, Cavada R (2003) Formal verification of diagnosability via symbolic model checking. Proceedings of the 18th international joint conference on Artificial intelligence pp 363–369

Contant O (2005) On monitoring and diagnosing classes of discrete event systems. PhD thesis, University of Michigan

Contant O, Lafortune S, Teneketzis D (2002) Failure diagnosis of discrete event system: the case of intermittent faults. International Conference on Decision and Control pp 4006–4017

Contant O, Lafortune S, Teneketzis D (2004) Diagnosis of Intermittent Faults. Discrete Event Dynamic Systems 14(2):171–202

Correcher A, Garcia E, Morant F, Quiles E (2003) Intermittent failure diagnosis in industrial processes. IEEE International Symposium on Industrial Electronics 2:723–728

Deng G, Qiu J, Liu G, Lyu K (2013) A novel fault diagnosis approach based on environmental stress level evaluation. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 227(5):816–826

Deng G, Qiu J, Liu G, Lyu K (2014a) A discrete event systems approach to discriminating intermittent from permanent faults. Chinese Journal of Aeronautics 27(2):390–396

Deng G, Qiu J, Liu G, Lyu K (2014b) A stochastic automaton approach to discriminate intermittent from permanent faults. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 228(6):880–888

Emerson EA, et al. (1990) Temporal and modal logic. Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B) 995(1072):1–5

Erkoyuncu JA, Khan S, Hussain SMF, Roy R (2016) A framework to estimate the cost of no-fault found events. International Journal of Production Economics 173:207–222

Fabre E, Hélouet L, Lefaucheux E, Marchand H (2016) Diagnosability of repairable faults. 13^{th} International Workshop on Discrete Event Systems pp 230–236

Fabre E, Hélouët L, Lefaucheux E, Marchand H (2018) Diagnosability of repairable faults. Discrete Event Dynamic Systems 28(2):183–213

Fromherz MPJ, Bobrow DG, de Kleer J (2004) Model-based computing for design and control of reconfigurable systems. AI Magazine 24(4):120–130

García E, Rodriguez L, Morant F, Correcher A, Quiles E (2008a) Latent nestling method: A new fault diagnosis methodology for complex systems. In: 34th Annual Conference of Industrial Electronics, IEEE, pp 253–258

García E, Rodríguez L, Morant F, Correcher A, Quiles E, Blasco R (2008b) Fault diagnosis with coloured Petri nets using latent nestling method. In: IEEE International Symposium on Industrial Electronics, IEEE, pp 986–991

Garcia HE, Yoo TS (2005) Model-based detection of routing events in discrete flow networks. Automatica 41(4):583–594

Ghazel M, Toguyéni A, Yim P (2009) State observer for DES under partial observation with time Petri nets. Discrete Event Dynamic Systems 19(2):137–165

Goldberg AV (1995) Scaling algorithms for the shortest paths problem. SIAM Journal on Computing 24(3):494–504

Gougam HE, Subias A, Pencolé Y (2013a) Supervision patterns: diagnosability checking by Petri net unfolding. 4th IFAC Workshop on Dependable Control of Discrete Systems pp 73–78

Gougam HE, Subias A, Pencolé Y (2013b) Supervision patterns: formal diagnosability checking by Petri net unfolding. IFAC Workshop on Dependable Control of Discrete Systems pp 73 – 78

Gougam HE, Subias A, Pencolé Y (2014) Discriminability analysis of supervision patterns by net unfoldings. IFAC Proceedings Volumes 47(2):459–464

Gougam HE, Pencolé Y, Subias A (2017) Diagnosability analysis of patterns on bounded labeled prioritized petri nets. Discrete Event Dynamic Systems 27(1):143–180

Gracia J, Saiz LJ, Baraza JC, Gil D, Gil PJ (2008) Analysis of the influence of intermittent faults in a microcontroller. In: 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems, IEEE, pp 1–6

Grastien A (2009) Symbolic testing of diagnosability. 20^{th} International Workshop on Principles of Diagnosis

Hardie FH, Suhocki RJ (1967) Design and use of fault simulation for saturn computer design. IEEE Transactions on Electronic Computers EC-16(4):412–429

Hashtrudi Zad S, Kwong RH, Wonham WM (2003) Fault diagnosis in discrete-event systems: framework and model reduction. IEEE Transactions on Automatic Control 48(7):1199–1212

Hsu YT, Hsu CF (1991) Novel model of intermittent faults for reliability and safety measures in long-life computer systems. International journal of electronics 71(6):917–937

Huang Z (2003) Rules based modeling of discrete event systems with faults and their diagnosis. PhD thesis, University of Kentucky

Isermann R (2006) Fault-diagnosis systems: an introduction from fault detection to fault tolerance. Springer Science & Business Media

Ismaeel AA, Bhatnagar R (1997) Test for detection and location of intermittent faults in combinational circuits. IEEE Transactions on Reliability 46(2):269–274

Jéron T, Marchand H, Pinchinat S, Cordier MO (2006) Supervision patterns in discrete event systems diagnosis. Discrete Event Systems, 2006 8th International Workshop on pp 262–268

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

Jiang S, Kumar R (2006) Diagnosis of repeated failures for discrete event systems with linear-time temporal-logic specifications. IEEE Transactions on Automation Science and Engineering 3(1):47–59

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

Jiang S, Kumar R, Garcia HE (2003) Diagnosis of repeated/intermittent failures in discrete event systems. IEEE Transactions on Robotic and Automatic 19(2):310–323

Johnson DB (1975) Finding all the elementary circuits of a directed graph. SIAM Journal on Computing 4(1):77–84

Kim CJ (2009) Electromagnetic radiation behavior of low-voltage arcing fault. IEEE Transactions on Power Delivery 24(1):416–423

Kimseng K, Hoit M, Tiwari N, Pecht M (1999) Physics-of-failure assessment of a cruise control module. Microelectronics Reliability 39(10):1423–1444

Kumar R, Takai S (2014) Comments on "polynomial time verification of decentralized diagnosability of discrete event systems" versus "decentralized failure diagnosis of discrete event systems": Complexity clarification. IEEE

Transactions on Automatic Control 59(5):1391–1392
Lamperti G, Zanella M (2004) Diagnosis of discrete-event systems by separation of concerns, knowledge compilation, and reuse. In: Proceedings of the 16th European Conference on Artificial Intelligence, IOS Press, pp 838–842
Lefebvre D, Leclercq E (2011) Stochastic petri net identification for the fault detection and isolation of discrete event systems. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 41(2):213–225
Lin F (1991) Analysis and synthesis of discrete event systems using temporal logic. IEEE International Symposium on Intelligent Control pp 140–145
Lin F (1994) Diagnosability of discrete event systems and its applications. Discrete Event Dynamic Systems 4(2):197–212
Lin F, Wonham WM (1988) On observability of discrete-event systems. Information Sciences 44(3):173–198
Madden MGM, Nolan PJ (1999) Monitoring and diagnosis of multiple incipient faults using fault tree induction. IEE Proceedings - Control Theory and Applications 146(2):204–212
Martínez MAT, Moreno EG (2008) Fault diagnosis and modeling of the liquids packaging process. a research based on Petri nets. In: 10th International Conference on Control, Automation, Robotics and Vision, IEEE, pp 1620–1624
Maul C, McBride JW, Swingler J (2001) Intermittency phenomena in electrical connectors. IEEE Transactions on Components and Packaging Technologies 24(3):370–377
Moreira MV, Jesus TC, Basilio JC (2011) Polynomial time verification of decentralized diagnosability of discrete event systems. IEEE Transactions on Automatic Control 56(7):1679–1684
Moreira MV, Basilio JC, Cabral FG (2016) "Polynomial time verification of decentralized diagnosability of discrete event systems" versus "Decentralized failure diagnosis of discrete event systems": A critical appraisal. IEEE Transactions on Automatic Control 61(1):178–181
Overton D (2006) No fault found returns cost the mobile industry $4.5 billion per year. WDSGlobal, juillet
Pan S, Hu Y, Li X (2012) Ivf: Characterizing the vulnerability of microprocessor structures to intermittent faults. IEEE Transactions on Very Large Scale Integration Systems 20(5):777–790
Pencolé Y, Subias A (2018) Diagnosis of supervision patterns on bounded labeled petri nets by model checking. 28th International Workshop on Principles of Diagnosis pp 184–199
Qiu W, Kumar R (2006) Decentralized failure diagnosis of discrete event systems. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 36(2):384–395
Rintanen J, et al. (2007) Diagnosers and diagnosability of succinct transition systems. International Joint Conference on Artificial Intelligence pp 538–544
Roberts M (1989) A fault-tolerant scheme that copes with intermittent and transient faults in sequential circuits. Proceedings of the 32^{nd} Midwest Symposium on Circuits and Systems pp 36–39
Rodriguez L, Garcia E, Morant F, Correcher A, Quiles E (2008) Application of latent nestling method using coloured Petri nets for the fault diagnosis in the wind turbine subsets. In: Proceedings of 2008 IEEE International Conference Emerging Technologies and Factory Automation, pp 767–773

Rodriguez-Urrego L, García E, Quiles E, Correcher A, Morant F, Pizá R (2015) Diagnosis of intermittent faults in IGBTs using the latent nestling method with hybrid coloured Petri nets. Mathematical Problems in Engineering 2015

Salvatore JB, Elizabeth R, Joanne Bechta D, Kishor S T, Nitin M, Robert M G, Mark D S (2003) Hybrid Automated Reliability Predictor Integrated Reliability Tool System HARP (Version 7.0). NASA Langley Technical Report Server

Sampath M, Sengupta R, Lafortune S (1995) Diagnosability of discrete-event systems. IEEE Transactions on Automatic Control 40(9):1555–1575

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis DC (1996a) Failure diagnosis using discrete-event models. IEEE Transactions on Control Systems Technology 4(2):105–124

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis DC (1996b) Failure diagnosis using discrete-event models. IEEE Transactions on Control Systems Technology 4(2):105–124

Santoro LP, Moreira MV, Basilio JC (2017) Computation of minimal diagnosis bases of discrete-event systems using verifiers. Automatica 77:93–102

Schumann A, Pencole Y (2007) Scalable diagnosability checking of event-driven system. International Joint Conference on Artificial Intelligence pp 575–580

Sharma R, Dewan L, Chatterji S (2015a) Computer networks reliability evaluations and intermittent faults. International Journal of Electronics and Electrical Engineering 3(6):465–471

Sharma R, Dewan L, Chatterji S (2015b) Fault diagnosis methods in dynamic systems: a review. International Journal of Electronics and Electrical Engineering 3(6):465–471

Shen Q, Qiu J, Liu G, Lyu K (2016) Intermittent faults parameter framework and stochastic Petri net based formalization model. Eksploatacja I neizawodnosc 18(2):1–210

Söderholm P (2007) A system view of the no fault found (NFF) phenomenon. Reliability Engineering & System Safety 92(1):1–14

Soldani S, Combacau M, Subias A, Thomas J (2006) Intermittent fault detection through message exchanges: a coherence based approach. International Workshop Principles Diagnosis pp 251–257

Soldani S, Combacau M, Subias A, Thomas J (2007a) Intermittent fault diagnosis: a diagnoser derived from the normal behavior. International Workshop Principles Diagnosis pp 391–399

Soldani S, Combacau M, Subias A, Thomas J (2007b) On-board diagnosis system for intermittent fault: Application in automotive industry. IFAC Proceedings Volumes 40(22):151–158

Sorensen B, Kelly G, Sajecki A, Sorensen P (1994) An analyzer for detecting intermittent faults in electronic devices. IEEE Proceedings of Systems Readiness Technology Conference pp 417–421

Steadman B, Pombo T, Madison I, Shively J, Kirkland L (2002) Reducing no fault found using statistical processing and an expert system. In: AUTOTESTCON Proceedings, IEEE, pp 872–878

Steadman B, Sievert S, Sorensen B, Berghout F (2005) Attacking bad actor and no fault found electronic boxes. IEEE AUTOTESTCON pp 821–824

Steadman B, Berghout F, Olsen N, Sorensen B (2008) Intermittent fault detection and isolation system. IEEE AUTOTESTCON pp 37–40

Syed WA, Khan S, Phillips P, Perinpanayagam S (2013) Intermittent fault finding strategies. Procedia CIRP 11:74–79

Thistle J, Wonham W (1986) Control problems in a temporal logic framework. International Journal of Control 44(4):943–976

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

Trigos M, Barrientos A, Del-Cerro J (2016) Unmanned helicopter faults diagnosis based on Petri nets. ID Revista de Investigaciones, 08(2):91–103

Viana G, Moreira MV, Basilio JC (2019) Codiagnosability analysis of discrete-event systems modeled by weighted automata. IEEE Transactions on Automatic Control 64(10):4361–4368

Viana GS, Basilio JC (2019) Codiagnosability of discrete event systems revisited: A new necessary and sufficient condition and its applications. Automatica 101:354–364

Viana GS, Basilio JC, Moreira MV (2015) Computation of the maximum time for failure diagnosis of discrete-event systems. American Control Conference pp 396–401

Yan R, He X, Zhou D (2015) Robust detection of intermittent faults for linear discrete-time stochastic systems with parametric perturbations. 34^{th} Chinese Control Conference pp 6308–6313

Yan Y, Ye L, Dague P (2010) Diagnosability for patterns in distributed discrete event systems. In: 21st International Workshop on Principles of Diagnosis DX'10

Yang H, Jiang B, Zhang Y (2012) Tolerance of intermittent faults in spacecraft attitude control: switched system approach. IET Control Theory & Applications 6(13):2049–2056

Ye L, Dague P (2012) A general algorithm for pattern diagnosability of distributed discrete event systems. In: Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on, IEEE, vol 1, pp 130–137

Ye L, Dague P, Yan Y (2009) An incremental approach for pattern diagnosability in distributed discrete event systems. In: 2009 21st IEEE International Conference on Tools with Artificial Intelligence, IEEE, pp 123–130

Yoo TS, Garcia HE (2004) Event diagnosis of discrete-event systems with uniformly and nonuniformly bounded diagnosis delays. Proceedings of the American Control Conference, 2004 6:5102–5107

Yoo TS, Garcia HE (2005) New results on discrete-event counting under reliable and unreliable observation information. IEEE Proceedings on Networking, Sensing and Control pp 688–693

Yoo TS, Garcia HE (2008) Stochastic event counter for discrete-event systems under unreliable observations. American Control Conference, 2008 pp 1145–1152

Yoo TS, Garcia HE (2009) Event counting of partially-observed discrete-event systems with uniformly and nonuniformly bounded diagnosis delays. Discrete Event Dynamic Systems 19(2):167–187

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

Zaytoon J, Lafortune S (2013) Overview of fault diagnosis methods for discrete event systems. Annual Reviews in Control 37(2):308–320

Zaytoon J, Sayed-Mouchaweh M (2012) Discussion on fault diagnosis methods of discrete event systems. 11th IFAC Workshop on Discrete Event Systems pp 9 –

12
Zhou C, Kumar R (2009) Computation of diagnosable fault-occurrence indices for systems with repeatable faults. IEEE Transactions on Automatic Control 54(7):1477–1489