Annual Reviews in Control 37 (2013) 308–320

Contents lists available at ScienceDirect

Annual Reviews in Control

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

Overview of fault diagnosis methods for Discrete Event Systems

J. Zaytoon [a,*], S. Lafortune [b]

[a] CReSTIC, University of Reims Champagne-Ardenne, F-51100 Reims, France
[b] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA

ARTICLE INFO

Article history:
Received 28 February 2013
Accepted 26 July 2013
Available online 11 October 2013

ABSTRACT

Fault diagnosis of Discrete Event Systems has become an active research area in recent years. The research activity in this area is driven by the needs of many different application domains such as manufacturing, process control, control systems, transportation, communication networks, software engineering, and others. The aim of this paper is to review the state-of the art of methods and techniques for fault diagnosis of Discrete Event Systems based on models that include faulty behaviour. Theoretical and practical issues related to model description tools, diagnosis processing structure, sensor selection, fault representation and inference are discussed.

Contents

1. Introduction . . . 308
2. Diagnosability and diagnosis of DES . . . 309
3. Classification of diagnosis methods . . . 312
   3.1. Classification of diagnosis methods with respect to fault compilation . . . 312
   3.2. Classification of diagnosis methods with respect to the modelling formalism . . . 313
      3.2.1. Diagnosis of timed and probabilistic automata . . . 313
      3.2.2. Fault diagnosis of Petri nets . . . 313
   3.3. Classification of diagnosis methods with respect to fault representation . . . 313
      3.3.1. Diagnosis using models including faulty behaviour . . . 313
      3.3.2. Diagnosis using fault-free models . . . 314
   3.4. Classification of diagnosis methods with respect to the decision structure . . . 314
      3.4.1. Centralized diagnosis . . . 315
      3.4.2. Decentralized structure with coordinated diagnosis . . . 315
      3.4.3. Distributed diagnosis . . . 315
4. Related problems . . . 316
   4.1. Predicting faults . . . 316
   4.2. Design issues: sensor selection and dynamic activation. . . . 316
   4.3. Sensor reliability . . . 316
   4.4. Robust diagnosis . . . 317
   4.5. Active diagnosis and fault tolerant control . . . 317
5. Conclusion and future directions . . . 317
   Acknowledgements . . . 318
   References . . . 318

1. Introduction

Dynamic systems can be classified into continuous-time systems and Discrete Event Systems (DES). The former capture "physical" system behaviour, typically using differential equation models, while the latter capture the logical and sequential behav-

* Corresponding author.
E-mail addresses: janan.zaytoon@univ-reims.fr (J. Zaytoon), stephane@umich.edu (S. Lafortune).

1367-5788/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.arcontrol.2013.09.009

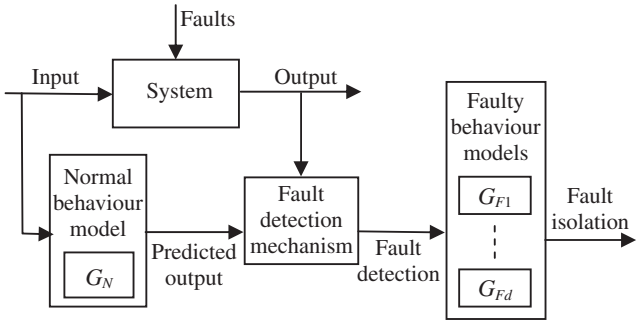

Fig. 1. Classical fault diagnosis configuration.

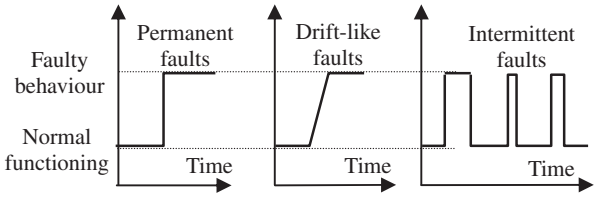

Fig. 2. Fault types.

iour of systems using discrete-state and event-driven models. This paper is focused on DES (Cassandras & Lafortune, 2008) and their diagnosis methods.

A fault causes a non-desired deviation of a system or of one of its components from its normal or intended behaviour. The deviation of system performance can either be tolerated or is considered to be critical in the case of a failure or a breakdown. Generally, fault diagnosis aims at achieving three complementary tasks: fault detection, fault isolation, and fault identification. Fault detection is a functionality that decides whether the system works in normal conditions or whether a fault has occurred. If a fault has occurred, fault isolation aims at localizing the system component(s) causing the fault. Fault identification is concerned with identifying the specific nature of the fault (its size, criticality, importance, etc.). Fault detection normally requires a model of the nominal behaviour of the system, while fault isolation and identification also require a model of faulty system behaviour under the considered faults.

In this paper, we lump the above three tasks under the generic terminology of "fault diagnosis", where the objective is to do both fault detection and isolation/identification. Fault diagnosis methods are generally based on the use of a model of the nominal (desired) behaviour of the system and a model for faulty behaviour in response to specified faults. Faults are usually considered as an additional input for the purpose of system modelling, and they are usually partitioned into failure modes, corresponding to types of faults in a system component that have the same effect according to either the configuration or the maintenance procedure. Fig. 1 represents a classical configuration for fault diagnosis.

In DES, faults can either be permanent, incipient (gradual or drift-like), or intermittent such as in the case bad wire contact or vibrations. Fig. 2 shows the time dependencies of these fault types.

Faults are typically classified with respect to the system component where they originate:

- Sensors faults, such as sensor offset or sensor stuck-off/stuck-on, which represent discrepancies between the measured and real values of system variables.
- Actuators faults, such as actuator stuck-off/stuck-on, which represent discrepancies between input commands for actuators and their real output.
- Plant faults, such as tank leakage or obstructed pipes, which induce changes in system dynamics.
- Controller implementation and execution faults, due to hardware or software problems.

The aim of the paper is to review the state-of-the art of fault diagnosis methods for DES based on models that include both nominal and faulty behaviour. It is an extended version of the invited plenary talk that was presented by the first author at the WODES 2012 IFAC Workshop on Discrete Event Systems (Zaytoon & Sayed Mouchaweh, 2012). The paper starts by reviewing in Section 2 the work of Sampath, Sengupta, Lafortune, Sinnamohidden, and Teneketzis (1995), which provided a formal foundation of diagnosability analysis and fault diagnosis of DES that was adopted and further developed by many groups. Section 3 provides a classification of diagnosis methods with respect to a number of criteria such as fault compilation, modelling tools, fault representation, and decision structure and architecture. Some related issues are presented in Section 4, including fault prediction, design problems, sensor selection and reliability, robust diagnosis, active diagnosis, and fault-tolerant control. Finally, Section 5 concludes the paper.

The fault diagnosis methods reviewed in this paper find their background in partial observation and observability approaches for DES that were developed in the 1980s and early 1990s (Caines, Greiner, & Wang, 1991; Cieslak, Desclaux, Fawaz, & Varaiya, 1988; Lin & Wonham, 1988; Ozveren & Willsky, 1990; Ramadge, 1986). These approaches deal with state estimation – current or initial state – and supervisory control. However, they are not directly concerned with the partition of faults and the identification of fault types based on faulty behaviour models. A state-based approach for the diagnosability of DES was proposed by Lin (1994). This approach provides algorithms for computing a sequence of test commands for diagnosing failures. Sensor optimization methods for diagnosis were investigated around the same time by Bavishi and Chong (1994). Several fault detection methods based on Petri net (PN) models of DES were also developed in the 1980s and the 1990s (Velilla & Silva, 1988; Prock, 1991; Sreenivas & Jafari, 1993). This short historical background is related to the work presented in this paper and is not meant to be exhaustive.

2. Diagnosability and diagnosis of DES

The definitions and algorithms proposed in Sampath et al. (1995, 1996) have provided the basic concepts and formal foundations of fault diagnosis and diagnosability analysis of DES for a large body of the literature on this topic. The proposed approach is based on the use of a classical automaton model of the system, denoted by $G = (X, E, \delta, x_0)$, where:

- X is the set of states of the system (including nominal and faulty states);
- $E = E_o \cup E_{uo}$ is the set of events, partitioned into observable events, E_o, and unobservable events, E_{uo}. Faults are usually represented using unobservable events, because their detection and diagnosis would be trivial if they were observable. Fault events are not the only unobservable ones; unobservable events also arise when sensors are unavailable or costly to implement;
- δ is the transition function, $\delta: X \times E \rightarrow X$;
- x_0 is the initial state.

The language generated by this automaton represents the set of all possible executions or sequences of events of the system in nominal and faulty operation (for the considered faults). Hence, this methodology is appropriate for classes of sensor, actuator, plant, or controller faults that can be modelled by unobservable

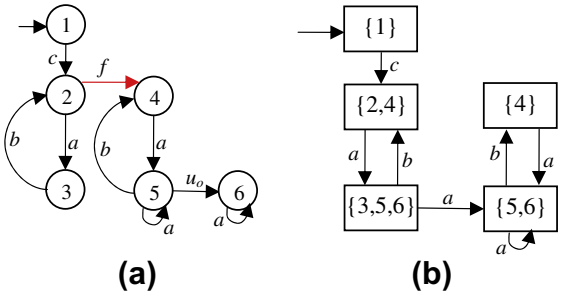

Fig. 3. An automaton (a) and its observer (b).

events with associated subsequent event-driven behaviour. (Note that no set of marked states needs to be specified for G since the entire, prefix-closed, system behaviour is being considered.)

The diagnosis problem is basically concerned with determining which faults (unobservable events), if any, explain a given observed sequence of events, based on the model of the system. Fault diagnosis is therefore closely related to the problem of state observability, which consists in building a deterministic automaton, called the *observer*, whose transitions are due to the observable events of the system and whose states are estimates of the true system state.

For example, Fig. 3b represents an observer for the 6-state automaton depicted in Fig. 3a in the case where a, b and c are observable events, and u_o and the fault f are unobservable events. The observer is initialized to state {1}, since 1 is the initial state of the automaton. The occurrence of event c activates state 2, which is indistinguishable to the observer from state 4 because of the unobservable fault event, f, associated with the transition between these two states. Hence, the observer moves to the state labelled with the state estimate {2,4}, which indicates that the automaton is either in state 2 or in state 4, following the occurrence of event c and prior to the occurrence of the next observable event. The only observable event that may occur here is a, which leads the automaton to state 3 if it was in state 2, or to state 5 if it was in state 4. Moreover, unobservable event u_o implies that state 6 is indistinguishable from state 5. Therefore, event a leads the observer to a new state labelled with the state estimate {3,5,6}, where the next observable events are either a or b. Event b will be observed if the system was in state 3 or 5 and it causes a transition to state estimate {2,4} in the observer. On the other hand, a second occurrence of event a corresponds to one of the two self-looping transitions and leads the observer to the new state estimate {5,6}. The observation of event b at this stage corresponds to the transition from state 5 to state 4, and leads the observer to conclude with certainty that the system is in state {4}.

In general, the construction the observer can lead to the problem of state explosion because the size of the observer is exponential in the size of the automaton, in the worst case. This is because observer states are subsets of automaton states.

A more challenging problem than state estimation is how to exploit the observable events to detect the occurrence of unobservable events. For example, the question for the automaton in Fig. 3a is how to detect the occurrence of event f using symptoms related to the observation of events a and b. It is easy in this case to see that an observation sequence starting with c and involving two consecutive occurrences of a is a symptom of f, because such a sequence can only be observed after the occurrence of f, and it implies that the automaton is either in state 5 or in state 6. On the other hand, a repetitive execution of the cycle given by the two consecutive events a and b does not allow an observer to conclude whether event f has occurred or not, because such an observation

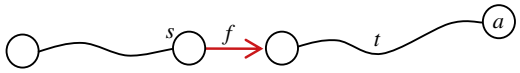

Fig. 4. Illustration of diagnosability.

cycle can either correspond to the two transitions between states 2 and 3, meaning that f has not yet occurred, or to the two transitions between states 4 and 5, implying that f has occurred.

The above model-based reasoning led Sampath et al. (1995) to introduce the following definition of diagnosability.

Definition 1. A fault is diagnosable if it can be detected with certainty within a finite number of observable events after its occurrence. This means that fault f is diagnosable if for every execution trace s of events ending with f, there exists a sufficiently long continuation trace t such that any other execution trace indistinguishable from $s \cdot t$ – that is, that produces the same record of observable events as $s \cdot t$ – also contains f.

Fig. 4 shows an example of a trace $s \cdot t$ of events with prefix s ending with f and suffix t ending in state a. The resulting observation is given by $P(s \cdot t)$, where the projection function $P(u)$ retains the observable events and filters out the unobservable ones in the trace u. The fault f is diagnosable if all traces of the system that produce the observation $P(s \cdot t)$ also contain f.

The definition of diagnosability can be extended to n-diagnosability as follows.

Definition 2. A fault is n-diagnosable if it can be detected with certainty within a specified number, n, of observable events after its occurrence.

This idea is conceptually sketched in Fig. 5 for a 6-diagnosable fault, f. Fig. 5a shows trace $s \cdot t$ of events with prefix s ending with f and suffix t containing 5 observable events. This figure indicates that f is not diagnosable within 5 observable events after its occurrence because one of the 5 other trajectories that are indistinguishable from $s \cdot t$ – the one that ends in state b – does not contain f. This means that the observation trace $P(s \cdot t)$ does not allow one to conclude with certainty whether event f has occurred or not.

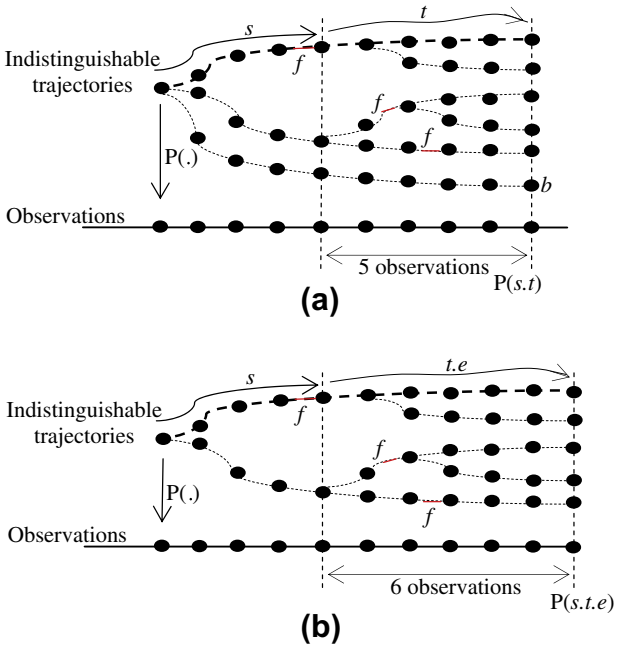

Fig. 5. Illustration of 6-diagnosability.

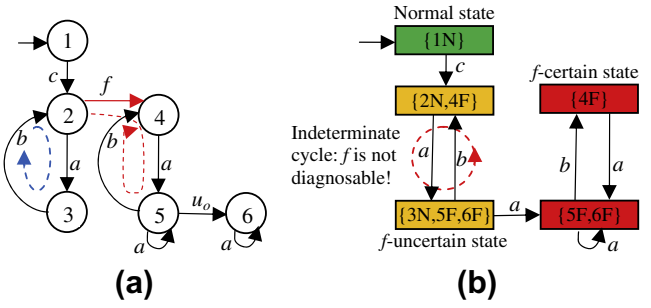

Fig. 6. An automaton (a) and its diagnoser (b).

Fig. 5b represents an extension of t with an observable event e and shows that only four trajectories remain indistinguishable from $s \cdot t \cdot e$. The occurrence of fault event f, the last event of trace s, can be detected with certainty in this case because each of the 4 indistinguishable trajectories contains f. Fault f will be 6-diagnosable if each possible occurrence of f in the entire system language can be detected with certainty, in the same way, after 6 observations.

Definition 3. A system is diagnosable if it is possible to detect within a finite delay occurrences of faults of any type using the record of observed events. Alternatively speaking, diagnosability requires that every occurrence of every fault event leads to observations distinct enough to enable unique identification of the fault event within a finite delay.

Based on these definitions, the observer automaton can be refined to build a *diagnoser automaton* by associating the labels F and/or N to the states within the state estimate observer states. Label F is used for states that can be reached by a trace containing event f, whereas label N is used for states that can be reached by a trace that does not contain event f. The resulting diagnoser is an automaton that is based on off-line compilation of observed trajectories. This automaton can be used either off-line to check diagnosability or online (on-the-fly) by connecting it to the system to provide on-line diagnosis upon the occurrence of observable events.

For example, Fig. 6b depicts the diagnoser for the automaton in Fig. 3a. Compared with the observer of Fig. 3b, the states are labelled with F or N in Fig. 6b. The diagnoser states can either correspond to:

- a *normal* state – such as {1N} – if all the corresponding states in the estimate set are labelled with N;
- an f-uncertain state – such as {2N,4F} or {3N,5F,6F} – if some of the corresponding states in the estimate state are labelled with N and others are labelled with F. In such a state, the diagnoser concludes that a fault event f may have occurred but it is not possible to ascertain from the observed event sequence up to that point whether the fault has indeed occurred.
- an f-certain state – such as {5F,6F} or {4F} – if all the corresponding states in the estimate have label F. In such a state, the diagnoser can conclude that fault event f has occurred with certainty.

In general, the relationship between the observer and the diagnoser is not as simple as going from Fig. 3b to Fig. 6b. For instance, if an unobservable transition labelled u_o were to be added between states 3 and 5 in the automaton in Fig. 3a, then the diagnoser state reached upon observable sequence ca would be {3N,5N,5F,6N,6F}, since states 5 and 6 could be reached with or without executing event f. Moreover, in general, the diagnoser state space need not be isomorphic to the observer state space; it could contain more states. The reader is referred to Cassandras and Lafortune (2008) for the description of two algorithms for building diagnosers. If there are multiple fault types, one can build one diagnoser for each fault type, as described above, or a single diagnoser for all fault types, using different F labels for the different fault types.

We define an f-indeterminate cycle in a diagnoser to be a cycle composed exclusively of f-uncertain states and corresponding to the presence of two cycling traces in the system with the same observable projection, such that f occurs in the 1st trace but not in the 2nd. The result that follows recalls, in non-mathematical terms, the main result of Sampath et al. (1995) regarding testing the property of diagnosability using diagnoser automata.

Theorem 1. *The system is diagnosable if and only if there are no f-indeterminate cycles in the diagnoser for any fault type f.*

Fig. 6b shows an f-indeterminate cycle between the diagnoser states {2N,4F} and {3N,5F,6F}. This cycle corresponds to the presence of two cycling traces in the automaton starting in state 2 and having the same observable projection, $a \cdot b$, such that the fault does not occur in the first trace, which is given by the cycle between states 2 and 3, but the fault occurs in the second trace, which starts with f followed with the cycle between states 4 and 5.

It should be noted that the above definitions and results related to diagnosability are based on the assumption that the system G under investigation is live (i.e., there are no terminating traces) and does not contain any cycle of unobservable events. The first assumption can be relaxed with some technical changes to deal with the diagnosis of terminating faulty traces; regarding the second assumption, the presence of a cycle of unobservable events after fault event f immediately implies a violation of diagnosability.

Diagnosing fault event f in the automaton of Fig. 3a is intuitively straightforward. Definition 3 and Theorem 1 provide a formal framework and algorithmic procedures for such model-based inferencing that can be applied to any system modelled by an automaton with corresponding sets of observable and unobservable events. The diagnoser is an efficient structure because it provides:

- a complete characterization of the diagnosis problem under the considered model: every state of the diagnoser is a possible diagnosis and every possible diagnosis is a state of the diagnoser;
- an efficient diagnosis algorithm: updating the diagnosis after a new observation only requires the firing of a single transition.

However, having an exact diagnoser is an ideal situation because its construction implies the availability of an exhaustive and correct faulty model, which is unrealistic in real complex systems. Moreover, the construction of the entire diagnoser may be unwieldy as in the worst case its size is exponential in the number of states of G, as well as in the number of faults if a single diagnoser is desired. The second limitation can be addressed by building separate diagnosers for each fault type. In this case, when building a diagnoser for a given fault type, the events corresponding to the other fault types are treated as other unobservable events; thus, the total complexity is linear in the number of fault types. The first limitation can be addressed in two ways: (i) development of a computationally simpler test (in the worst case) than Theorem 1 for the property of diagnosability, as explained in the next paragraph and (ii) on-the-fly construction of the diagnoser state, based on G, at run-time, a topic discussed further in Section 3.1.

To overcome the potential state explosion problem for off-line testing of diagnosability using diagnosers, the so-called "twin machine" technique (Jiang, Huang, Chandra, & Kumar, 2001; Yoo &

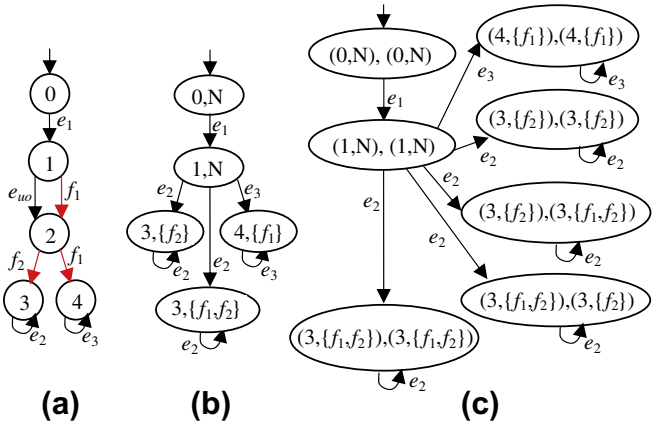

Fig. 7. A system G (a), its nondeterministic observation automata G_0 (b), and the composition of G_0 with itself (c).

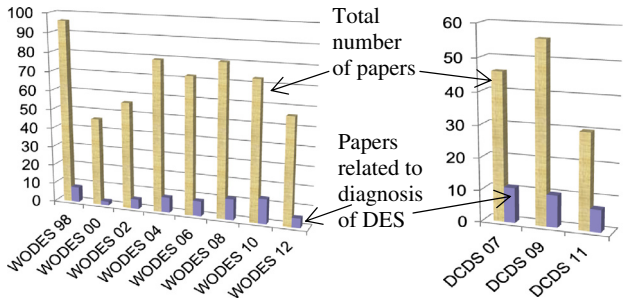

Fig. 8. Publications statistics from the WODES and DCDS Workshops.

Lafortune, 2002a) was introduced to provide a worst-case polynomial test in the number of states of the system for diagnosability, without constructing diagnosers. The idea here is that a fault f is diagnosable if and only if there is no pair of arbitrarily long traces having the same observable projection, such that f occurs in the first trace but not in the second.

For the sake of illustration, an example presented by (Jiang et al., 2001) is recalled. Consider the system G given in Fig. 7a, with observable events e_1, e_2 and e_3, unobservable event e_{uo}, and faults f_1 and f_2, assumed to be of different types. The first step in this approach is to generate the nondeterministic observer in Fig. 7b, denoted by G_0, whose states can be reached by taking the observable transitions. In this automaton, a label f_i in a state indicates that fault f_i occurs along a certain path from the initial state to this state; otherwise, the state label is "N". If multiple faults occur along a path, the label becomes a set. The next step is to compute the parallel composition of G_0 with itself. The resulting automaton, denoted by G_d, is depicted in Fig. 7c. Its states are now pairs of states with associated labels. The system is not diagnosable if G_d contains a cycle where the left labels and the right labels differ by fault type f_i in every state along the cycle (they may differ in other ways as well). Specifically, such as cycle means that fault type f_i is not diagnosable in G. For example, in Fig. 7c, there is a self-loop at the state $((3,\{f_2\}), (3,\{f_1,f_2\}))$ or at the state $((3,\{f_1,f_2\}), (3,\{f_2\}))$. In each case, the cycle indicates that fault f_1 is not diagnosable in G, as the presence of each cycle implies the existence of two arbitrarily long traces with the same projection, where $f_1$is contained in one trace but not in the other.

Next, suppose it is not necessary to distinguish the fault type f_1 from the type f_2. Then by letting $f_2 = f_1$ in Fig. 7 and deleting the redundant states, the modified G_d is obtained (not shown here). In the modified G_d there does not exist any cycle as stated above, so the system is diagnosable if there is a single fault type.

3. Classification of diagnosis methods

The early contributions presented in the previous section established the basic definitions and the formal and algorithmic foundations of fault diagnosis and diagnosability analysis of DES. Subsequent contributions from many research teams have been concerned with the development of new models, new properties, new algorithms, and efficient solutions for fault diagnosis of DES. This line of work is still on-going today and has produced significant results and publications. For example, Fig. 8 presents some statistics from the WODES (Workshop on Discrete Event Systems) and DCDS (Workshop on Dependable Control of Discrete Systems) series of international conferences dedicated to DES. These statistics shows that about 12% of WODES papers and 22% of DCDS papers are related to fault diagnosis problems. One to three papers on fault diagnosis of DES have also been published every year since 1998 in the Journal of Discrete Event Dynamic Systems. This represents about 12% of the papers of this international journal of the DES community. Other contributions to fault diagnosis of DES have also been published in control journals with broad scope, such as Automatica, IEEE Transactions on Automatic Control, Control Engineering Practice, and others. These figures show that the research domain of fault diagnosis of DES is vibrant, producing interesting results and well-known diagnosis methods that are well recognized by the community.

These contributions have been accompanied with various diagnosability notions and ad hoc algorithms to construct diagnosers and verify diagnosability, and this makes it sometimes difficult to choose a suitable diagnoser/diagnosability approach for a given application. An important effort is therefore still needed to apply these results to real applications. In this view, many application examples have been considered to provide proof of the established concepts in many areas, including: manufacturing (Philippot, Sayed-Mouchaweh, & Carré-Ménétrier, 2009; Viswanadham & Johnson, 1998); heating, ventilation, and air conditioning systems (Sampath et al., 1996); transportation (Şimşek, Sengupta, Yovine, & Eskafi, 1999); document processing systems (Sampath, Godambe, Jackson, & Mallow, 2000); telecommunication networks (Fabre & Benveniste, 2007; Rozé & Cordier, 2002); mixing batch processes (Garcia, Correcher, Morant, Quiles, & Blasco, 2005); and computer security (Genc, 2008).

This section provides a classification of diagnosis methods with respect to a number of criteria such as fault compilation, modelling tools, fault representation, and decision structure and architecture. The usage of the term "diagnoser" hereafter will be generic and includes the diagnoser reviewed in Section 2 as well as more general notions of diagnosers for fault detection, isolation and identification in DES.

3.1. Classification of diagnosis methods with respect to fault compilation

Fault diagnosis can either be achieved using an off-line compiled diagnoser or computed on-line.

In the off-line case, the system to be diagnosed is considered to be in a test-bed, i.e., not in normal functioning condition. The desired diagnoser is compiled based on testing a set of inputs (commands sequences) and observing the resulting outputs. Off-line compilation of a diagnoser provides a complete off-line characterization of the diagnosis problem and an efficient on-line solution in terms of diagnosis response time. This is because every state of the diagnoser provides a possible diagnosis of the system and updating this diagnosis only requires the firing of a transition of the diagnoser as a consequence of the observed system events. However, as

mentioned in the previous section, off-line compilation of a diagnoser requires the availability of an exhaustive and correct faulty model – which is unrealistic in real complex systems – and is computationally challenging.

Another approach consists in on-line detection and isolation/identification of the set of faults that may have occurred after each new observation acquired from the system during its operation. Complex computations may therefore be required to achieve the on-line diagnosis. Such an approach is more computationally demanding in terms of diagnosis response time, but results in a substantial gain in memory space because there is no need to store the complete diagnoser.

3.2. Classification of diagnosis methods with respect to the modelling formalism

Many modelling formalisms have been used to build diagnosers, including:

(i) Automata (Sampath et al., 1995) and their timed and probabilistic extensions.
(ii) Petri nets (Basile, Chiacchio, & De Tommasi, 2008, 2009; Cabasino, Giua, & Seatzu, 2010; Dotoli, Fanti, & Mangini, 2009; Fanti, Mangini, & Walter, 2011; Genc & Lafortune, 2007; Lefebvre & Delherm, 2007; Ramirez-Trevino, Ruiz-Beltran, Rivera-Rangel, & Lopez-Mellado, 2007).
(iii) Statecharts and hierarchical state machines (Idghamishi & Zad, 2004; Paoli & Lafortune, 2008).

The use of some of these modelling formalisms for diagnosis is discussed next.

3.2.1. Diagnosis of timed and probabilistic automata

Diagnosis methods for Timed Systems based on timed automata have been proposed by Bouyer, Chevalier, and D'Souza, 2005; Cassez, 2009; Chen and Provan, 1997; Jiang and Kumar, 2006; Tripakis, 2002; Zad, Kwong, and Wonham, 2005, among others. These contributions are mainly based on the definition of time diagnosability, which requires the diagnosability condition to hold after a bounded time interval, instead of a bounded number of events. The issues arising in this context are concerned with the choice of time semantics – tick event or dense time –, the definition of diagnosability, the construction of diagnosers, the characterization and reduction of complexity, and the relations with untimed systems.

Diagnosis methods have also been extended to probabilistic systems leading to probabilistic diagnosers (Athanasopoulou & Hadjicostis, 2005; Fabre & Jezequel, 2010; Lunze & Schroder, 2001; Thorsley & Teneketzis, 2005; Thorsley, Yoo, & Garcia, 2008; Wang, Chattopadhyay, & Ray, 2004). The aim of these methods is to build a deterministic state machine that gives the probability distribution on states and diagnosis values, given any observed event sequence. Diagnosability notions and definitions for stochastic and probabilistic automata have also been proposed.

3.2.2. Fault diagnosis of Petri nets

The aim of PN based diagnosis methods is to use the structure, the analytical capabilities, and the intrinsically distributed nature of PN models – where the notions of state and action are local – to reduce the computational complexity of diagnosis problems by avoiding the exhaustive enumeration of the system's state space, as well as to deal with some classes of infinite state systems (non-regular languages). Results on the property of diagnosability within the framework of PNs have also been proposed recently.

Some PN based diagnosis methods consider that the marking of certain places is observable (Chung, 2005; Ghazel, Bigand, & Toguyéni, 2005; Hernandez-Flores, Lopez-Mellado, & Ramirez-Trevino, 2011; Lefebvre & Delherm, 2007; Miyagi & Riascos, 2010; Ramirez-Trevino et al., 2007; Ushio, Onishi, & Okuda, 1998; Wen, Li, & Jeng, 2005; Wu & Hadjicostis, 2005), while others are based on unobservable net markings but observable sets of transitions (Basile, Chiacchio, & De Tommasi, 2009; Benveniste, Fabre, Haar, & Jard, 2003; Cabasino, Giua, Possi, & Seatzu, 2011; Dotoli et al., 2009; Fabre, Benveniste, Haar, & Jard, 2005; Fanti et al., 2011; Genc & Lafortune, 2007; Jiroveanu & Boel, 2008; Jiroveanu & Boel, 2010). Diagnosis methods based on stochastic Petri nets have also been proposed (Aghasaryan, Fabre, Benveniste, Boubour, & Jard, 1998).

As an illustrative example, the approach proposed by Cabasino et al. (2010), Cabasino et al. (2011) for the diagnosis of labelled PNs is described. Given a sequence of observable events, w, the aim of this approach is to characterize the minimal sequences of unobservable events that are interleaved with w – whose firings explain w – and to determine the resulting reachable marking subset (called basis markings) using linear algebraic constraints. An on-line diagnosis procedure is proposed to associate a diagnosis state to each observation w and to each fault class. This procedure is based on matrix multiplications and the manipulation of integer constraint sets. In the case of bounded net systems, the basis reachability graph can be calculated off-line to provide fast on-line diagnosis. However, a very large memory size may be required for this graph. An extension of this approach to systems modelled using fluid PNs is proposed in Mahulea, Seatzu, Cabasino, and Silva (2012) to exploit the convexity property of these nets and improve computational cost of the diagnosis in some cases.

3.3. Classification of diagnosis methods with respect to fault representation

We discuss further fault diagnosis based on nominal and faulty system behaviour.

3.3.1. Diagnosis using models including faulty behaviour

As discussed earlier in this paper, fault diagnosis (including both detection and isolation/identification) requires the knowledge of the faulty behaviour of the system. This approach can provide good diagnosis results in the case of predictable faults. However, it is not always realistic to exhaustively foresee all the faults and, therefore, only those faults that are explicitly considered in the system model can be detected and identified.

Faulty models are based on different types of fault representations, such as the execution of an event (event-based diagnosis), reaching a faulty state (state-based diagnosis), the execution of a supervision pattern, or the verification of partial temporal constraints.

Event-based diagnosis – such as the approach of Sampath et al. (1995) reviewed in Section 2 – decides if a fault has occurred and its type based only on the observation of event sequences. Event-based diagnosis methods can be used to diagnose intermittent faults since they consider a fault as the occurrence of an unobservable event (Contant, Lafortune, & Teneketzis, 2004). These approaches require the initialization of both the diagnoser and the model at the same time because the diagnoser makes its decision on the basis of the observed sequences of events. This initialization is not always easy to achieve in real systems, which may necessitate the introduction of additional unobservable events at the modelling stage.

State-based diagnosis is based on partitioning the state space of the system according to the failure status. The approaches proposed by Zad, Kwong, and Wonham (2003) and Lin (1994) are related to systems with binary inputs and outputs. Each state is labelled with the binary vector of its associated outputs and the

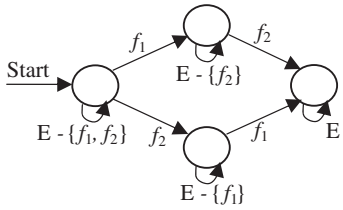

Fig. 9. A supervision pattern representing the occurrence of two faults f_1 and f_2.

diagnoser uses the sequence of binary output vectors associated with the system states to detect and isolate failures. The state-based diagnosis methods are well adapted to diagnose permanent faults since they consider the fault as the fact of reaching at a faulty state. However, these methods are generally not well suited to diagnose intermittent faults. Since a state-based diagnoser can determine the occurrence of a failure mode based on the generated state output, no information about the state or the failure status of the system is required before the initialization of the diagnoser. Therefore, the advantage of state-based diagnosis is that there is no need to initialize the system and the diagnoser simultaneously, and the diagnoser may be initialized at any time while the system is in operation.

An approach combining the advantages of state and event based diagnosis is proposed in Sayed-Mouchaweh, Philippot, and Carré-Ménétrier (2008).

A fault can also be represented as an execution of a given supervision pattern, which is a temporal property related to the occurrence of a set of trajectories/events that must be diagnosed (Jéron, Marchand, Pinchinat, & Cordier, 2006). The notion of supervision patterns is general enough to cover an important class of diagnosis objectives, including detection of permanent faults, but also transient faults, multiple faults, repeated faults, as well as quite complex sequences of events. For example, the supervision pattern given in Fig. 9 represents the occurrence of two faults f_1 and f_2. This supervision pattern behaves as an acceptor that accepts all the events of the system in any state: a fault advances the pattern to a new state while any other event is accepted through the self-looping transition which maintains the pattern in its current state.

Supervision patterns are very useful to generalize the properties to diagnose and clarify the separation between the diagnosis objectives and the system specifications. The diagnosis results can therefore be easily reused for new but similar diagnosis problems, due to their generic nature. This is a major advantage over other fault representation approaches whose result are rather difficult to reuse because they are usually associated with many different notions of diagnosability and they employ specific algorithms for the construction of the diagnoser and for the verification of diagnosability.

3.3.2. Diagnosis using fault-free models

Diagnosis with fault-free models is based on comparing the system's output with the model's nominal output. A fault is detected if an observed behaviour of the system cannot be reproduced by its model. However, fault isolation and identification may not be possible in this case because the model does not include the faulty behaviour and, therefore, the diagnosability of a given fault is not guaranteed.

The fault-free modelling approach proposed by Pandalai and Holloway (2000) uses condition templates to determine if the system generates events in the right order or within the given time delays. A fault is detected when there are missing or wrong reactions in the process. In these cases, the events related to the template help to isolate the fault. In Sayed-Mouchaweh (2012), expert

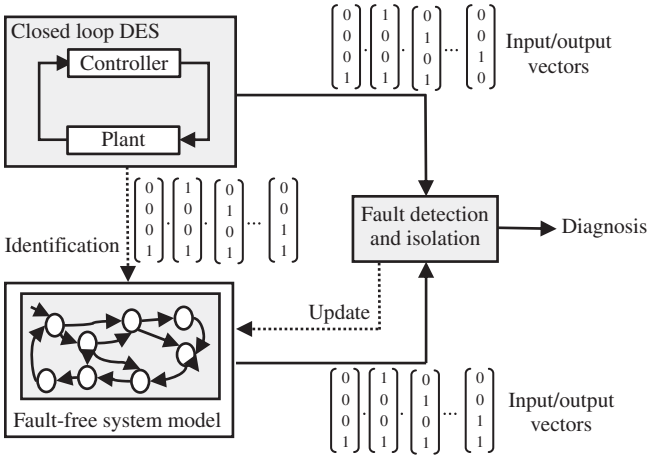

Fig. 10. Fault-free modelling for diagnosis (Roth et al., 2011).

knowledge is associated with condition templates to identify the faults related to missing or unexpected events, and progressive monitoring is used to reduce the set of fault candidates after the occurrence of new observable events.

Another practical fault-free modelling approach for fault diagnosis of manufacturing systems has been proposed by Roth, Lesage, and Litz (2011). This approach, depicted in Fig. 10, starts by identifying the fault-free model of the system. A fault is detected for any system behaviour that is not part of the identified model. Fault localization is inspired by the residual techniques commonly used in continuous systems. It is based on comparing the observed and expected sequences and calculating a small set of unexpected and missed fault candidates. This set is further reduced by applying a heuristic candidate-set reduction algorithm that provides a good estimate about the fault that may have occurred. Despite its simplicity and practicality, this approach can exhibit a high-rate of false alerts and it provides no guarantee concerning the diagnosability of certain faults. An extension to timed models was proposed in Schneider, Litz, and Danancher (2011).

3.4. Classification of diagnosis methods with respect to the decision structure

Three main processing structures, or architectures, are used to calculate the fault diagnosis decision: centralized, decentralized (Boel & van Schuppen, 2002; Chakib & Khoumsi, 2012; Debouk, Lafortune, & Teneketzis, 2000; Qiu & Kumar, 2006; Sengupta, 1998; Takai & Kumar, 2010; Wang, Yoo, & Lafortune, 2007; Zhou, Kumar, & Sreenivas, 2008) and distributed (Fabre et al., 2005; Genc & Lafortune, 2007; Pencolé & Subias, 2009; Ramirez-Trevino et al., 2007; Su, Wonham, Kurien, & Koutsoukos, 2002; Su & Wonham, 2004). Note that the distinction between the decentralized and the distributed structures is sometimes blurry. Generally speaking, decentralized approaches have a set of diagnosers, each with different observation capabilities, but all considering the global system model in their model-based inferencing. In distributed approaches, the individual diagnosers only use partial (local) system models as opposed to the global system model.

3.4.1. Centralized diagnosis

In the centralized structure, the diagnosis is calculated using one global (monolithic) diagnoser, which is constructed using the global model of the system to be diagnosed. This structure is depicted in Fig. 11. The Mask represents the observation function. It filters out the unobservable events of the plant and provides the sequences of observed events to the diagnoser.

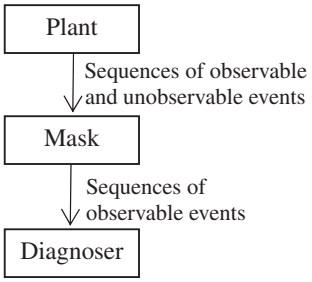

Fig. 11. Centralized diagnosis structure.

The main advantage of centralized diagnosis approaches is their diagnosis precision and conceptual simplicity. However, their main disadvantage is their prohibitive computational complexity, since they require a centralized plant model to generate the centralized diagnoser. The resulting models may become too large to be physically stored when a large-scale system is under consideration. Even if a centralized diagnoser exists physically, it may still suffer from weak robustness and low maintainability.

3.4.2. Decentralized structure with coordinated diagnosis

In the decentralized structure (Fig. 12), the system is partitioned into a number of sites. Each site knows the entire system model, has local observations, and uses a local diagnoser that computes a local diagnosis decision based on its partial observation of the whole system. A coordinator provides the final diagnosis decision as a function of the local diagnosis decisions that are communicated to it. The local diagnosers and the coordinator are constructed using a global model of the system. The local diagnosers may not communicate directly with each other, and usually only limited communication among them through the coordinator is permitted.

The main problem to address in decentralized architectures is about how the sites can jointly discover the occurrence of a fault, knowing that the available information can be ambiguous, incomplete, delayed, and possibly erroneous. The coordinator should therefore have some memory and processing capabilities to coordinate the required exchange of information between the local diagnosers to resolve the ambiguities of the local decisions. However, these capabilities should be constrained, otherwise, the centralized structure could be replicated at the coordinator's site by communicating all observations to it, which would defeat the purpose of the decentralized structure.

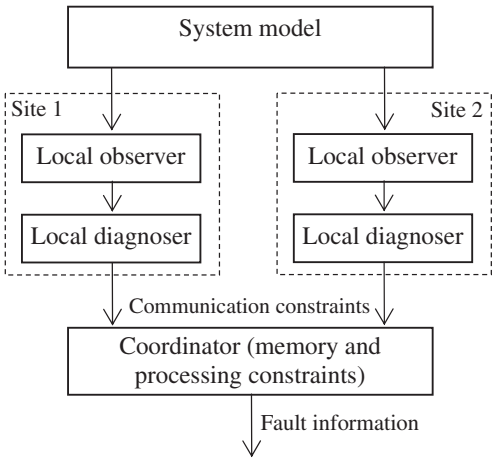

Fig. 12. Decentralized diagnosis structure.

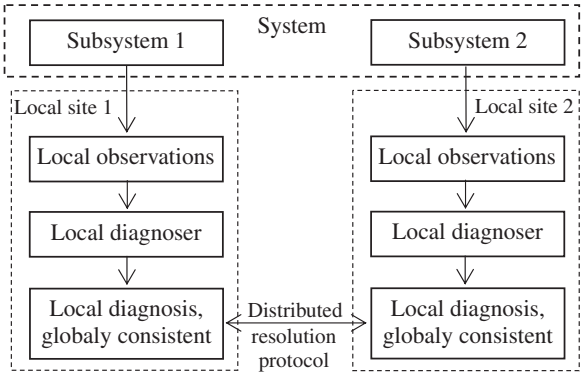

Fig. 13. Distributed diagnosis structure.

The main objective of decentralized diagnosis is therefore to design a set of protocols, analyze their "complexity–performance" trade-off, build diagnosers and compare their performance to the centralized case, and verify diagnosability. For example, three protocols using a coordinator with varying but limited memory and processing capabilities were proposed in Debouk et al. (2000).

The purpose of introducing decentralized diagnostic architectures is to perform fault detection and isolation in a manner that accounts for the decentralization of information in complex (interconnected) systems, while still preserving, if possible, the diagnostic capability of a centralized diagnoser. However, a decentralized diagnostic architecture still requires, in general, a centralized system model to construct the local diagnosers, and this will also result in complex computations.

3.4.3. Distributed diagnosis

Distributed approaches (Benveniste et al., 2003; Pandalai & Holloway, 2000; Pencolé & Subias, 2009; Su & Wonham, 2004) achieve diagnosis using a set of local models without referring to a global system model (Fig. 13). The aim is to improve scalability and robustness of diagnostic methodologies. Each subsystem knows only its own part of the global model and has its local diagnoser in order to perform diagnosis locally. This diagnosis computation is based on the local model and the information communicated directly to it by the other local diagnosers through a communication protocol. The information exchanged among local diagnosers is used to update their own information and compensates their partial observation.

A communication protocol must be defined to insure consistency among local diagnosers in case of conflicts between their local decisions. If the local models (subsystems) do not interact in a hierarchical or tree-like manner, the communication protocol may require time-consuming computation and a large state space. The challenge of distributed diagnosis is how to perform local diagnosis that is equivalent, if possible, to the global one, using a scalable communication protocol (with respect to the number of component modules), and without the need to use a global model.

Different settings and model structures – modular, hierarchical – have been proposed for distributed diagnosis (Debouk, Malik, & Brandin, 2002; Fabre, Benveniste, & Jard, 2002; Pencolé & Cordier, 2005; Qiu & Kumar, 2005b; Ricker & Fabre, 2000; Su & Wonham, 2005 and Su & Wonham, 2006). The aim of this diversity is to deal with different issues related to synchronization types, communication delays and losses, order preservation of information, model structure and complexity.

As a consequence of the various decentralized and distributed diagnosis structures that have been considered by researchers, many corresponding diagnosability notions and properties have been defined and analyzed. These include local diagnosability

(Pencolé, 2004), independent diagnosability (Sengupta, 1998), joint diagnosability (Sengupta & Tripakis, 2002), codiagnosability (Qiu & Kumar, 2006), conditional codiagnosability (Wang, Yoo, & Lafortune, 2005), D-codiagnosability (Wang et al., 2007), modular diagnosability (Contant, Lafortune, & Teneketzis, 2006), and many others. It is therefore not easy for the non-expert reader to clearly understand the differences between – and the positioning of – these diagnosability notions and properties. A comparative review of this area of research would be most useful, but is beyond the scope of the present paper.

4. Related problems

This section presents some complementary issues that are directly related to fault diagnosis. They include fault prediction, design problems, sensor selection and reliability, robust diagnosis, active diagnosis, and fault-tolerant control.

4.1. Predicting faults

The task of fault prediction is to ascertain the occurrence of an impending fault prior to its occurrence, based on the strings of observable events (Genc & Lafortune, 2006; Jéron, Marchand, Genc, & Lafortune, 2008; Kumar & Takai, 2008 and Kumar & Takai, 2010). This helps provide timely reaction to an impending fault so that corrective actions may be initiated before its occurrence. Note the contrast with the task of diagnosis, which requires the detection and the localization of a fault after its occurrence. The corresponding notion of predictability has also been proposed and investigated by Genc and Lafortune (2006), Kumar and Takai (2008). A system is predictable if each faulty trace possesses a non-faulty prefix such that any indistinguishable trace will inevitably lead to the fault. Many contributions have been concerned with the development of algorithms for the (polynomial) verification of predictability and the construction of off-line, on-line and decentralized predictors. Algorithms to predict the occurrences of a pattern that describes event sequences have also been proposed (Jéron et al., 2008).

4.2. Design issues: sensor selection and dynamic activation

A common approach to ensure diagnosability and build diagnosable systems is to change its observability properties by equipping the system with an appropriate set of sensors. The challenge is then to determine the optimal, feasible set of sensors that will meet the requirements: Which sensors to use? How many of them? Where to place them? To answer these questions, many economic, security, and energy-related factors can be considered, such as the cost of measurements, sensors availability and their lifespan, and battery power.

In diagnosability analysis, different approaches have been proposed to select a minimal subset of sensors (Debouk, Lafortune, & Teneketzis, 2002; Jiang, Kumar, & Garcia, 2003; Yoo & Lafortune, 2002b), a least expensive set of sensors (Ribot, Pencolé, & Combacau, 2008), or an optimal sensor configuration to balance the cost-performance trade-offs (Lin, Yoo, & Garcia, 2010; Lin, Garcia, & Yoo, 2013). Basilio, Souza Lima, Lafortune, and Moreira (2012) have proposed a method to compute minimal or optimal event subsets that ensure diagnosability by exploiting the structure of the diagnoser. The idea is to avoid selecting the events that lead to indeterminate cycles by focusing on events of traces that take uncertain states to some cycles of certain states of the diagnoser.

Other recent contributions deal with the problem of dynamic activation and deactivation of sensors for diagnosis purposes (Cassez & Tripakis, 2008; Dallal & Lafortune, 2011; Shu et al., 2010; Thorsley & Teneketzis, 2007; Wang, Lafortune, Girard, & Lin, 2010). This dynamic modification of the set of events to observe is very important in communication networks, for example, where the sensors are not purchased for the whole duration of the process and/or where a small cost is incurred every time the sensor is used. The problem here is to determine which sensors to activate – which information is really needed – after the occurrence of a trace of events, as sensor measurements may be costly for bandwidth, security, energy, or other considerations. Many issues have been studied and formalized in this context, including ensuring diagnosability while solving an optimal control problem that captures the number or cost of the sensors, and/or their commutation frequency.

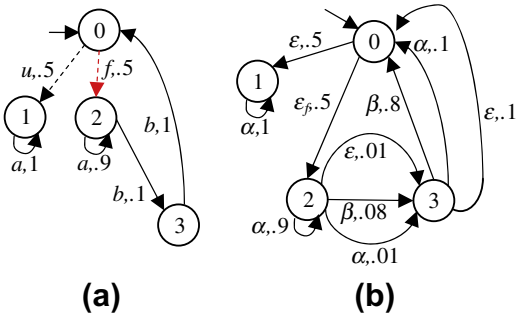

Fig. 14. A stochastic automaton (a) and its corresponding sensor output automaton (b).

4.3. Sensor reliability

Most of the established literature on fault diagnosis of DES classifies events as either observable, in which case a sensor outputs a reading when the event occurs, or unobservable, in which case no sensor outputs a reading. Faults are often modelled as instances of unobservable events. The implicit assumption in this sensor model is that all the sensors reading the observable events are perfectly reliable, that is, whenever an observable event occurs, the associated sensor will transmit its occurrence. In practice, in many safety hazard situations, such as in nuclear systems, this assumption is not valid because the difficulty in placing sensors and analyzing sensor data makes sensors inherently unreliable. In the same way, sensors reading observable events are not perfectly reliable in many other application domains.

The problem of unreliable sensors has been considered in Athanasopoulou, Li, and Hadjicostis (2006) and Thorsley et al. (2008). The approach proposed by Thorsley et al. (2008) consists in considering two main categories of sensor unreliability: *misclassification*, where a sensor reports an incorrect reading as a result of the occurrence of a particular event, or *misdetection*, where a sensor does not make a reading as a result of an event's occurrence. The approach uses Markov chain construction to generate a stochastic diagnoser. The objective of the fault diagnosis problem is to determine the probability that a fault has occurred given a sequence of observations, and the objective of the diagnosability problem is to determine conditions ensuring that any fault occurrence will be detected. Many notions of diagnosability are proposed to consider the probability of false negatives and false positives. This approach is illustrated using the stochastic automaton depicted in Fig. 14a (Thorsley et al., 2008). A probability is associated with each transition together with the corresponding event. The sum of the probabilities of the output transitions of a given state must be equal to 1. The observable event a is reliably observed and its occurrence will always result in an output of α. The events u and f (the fault) are unobservable and any occurrence of

these events will result in the null output ε. The probability of occurrence of each of these two events is .5 when the system is in state 0. The event b is both misclassifiable and misdetectable. An observation of β is the most likely outcome; however, there is probability .1 that an incorrect sensor reading of α will be observed, and there is probability .1 that no sensor reading at all will be made when b occurs. Fig. 14b depicts the sensor output automaton corresponding to the stochastic automaton of Fig. 14a and the sensor model just described. The transitions that are labelled with the event b in Fig. 14a are split into three transitions in Fig. 14b, including the corresponding misclassifiable output α and the misdetectable output ε with their associated probabilities. The resulting stochastic diagnoser is given in Thorsley et al. (2008). It can be used to check diagnosability and to determine the probability that a fault has occurred given a sequence of observations.

4.4. Robust diagnosis

In most cases, fault diagnosis of DES assume that sensors work properly. However, bad sensor operation can make sensors fail to report event occurrences. This means that a previously observable event related to a sensor may become unobservable when the sensor fails. Robust diagnosis can be used to obtain a diagnosable system despite sensors failures. Robust diagnosis finds its background in robust controller synthesis in the context of supervisory control of DES (Rohloff, 2005).

A global approach to robust diagnosis has been proposed in Basilio and Lafortune (2009) and Carvalho, Lima, Moreira, Basilio, and Lafortune (2013). This approach deploys the redundancy within the diagnosis bases – i.e., the different subset of observable events that guarantee diagnosability – to verify diagnosability and design a robust diagnoser despite sensor failures. The idea is to run a set of partial diagnosers in parallel, each designed for a particular diagnosis basis to work correctly under a certain combination of sensor failures, and to guarantee that at least one of these diagnosers will issue the correct diagnostic decision about unobservable faults. Given a set of possible combinations of sensors failures, the definition and testing of robust diagnosability for centralized and decentralized systems are also proposed. The extension of robust diagnosis and diagnosability to the case of intermittent loss of observations is considered in Carvalho, Basilio, and Moreira (2012). In this case, a sensor malfunction or a communication failure is modelled by duplicating its associated transition with an unobservable event transition.

The diagnosis problem introduced by (Takai, 2010, 2012) concerns the case of systems with multi-configurations, such as flexible manufacturing systems, and requires that the system be described by a set of possible models, each of which has its own nonfailure specification, over a common event set. The robust diagnosability formulation provides conditions for the existence of a single diagnoser that detects faults in any possible model within a bounded number of steps. Algorithms are proposed for verifying robust diagnosability and synthesizing a robust diagnoser.

A generalization of the two previous robust diagnosability notions was proposed by Carvalho, Moreira, and Basilio (2011) to consider uncertainties in both the system model and the observable event set.

4.5. Active diagnosis and fault tolerant control

Active diagnosis is concerned with the design of a controller that satisfies specified control objectives and results in a diagnosable controlled system (Chanthery & Pencolé, 2009; Sampath, Lafortune, & Teneketzis, 1998). This is usually achieved by restricting the behaviour of the system, by feedback control, to alter its diagnosability property in a way that guarantees that the system always remains diagnosable. Such an integrated approach to fault diagnosis and supervisory control was proposed in Sampath et al. (1998) to determine the supremal controllable, observable and diagnosable sublanguage of a given language – representing the desired behaviour – and to synthesize the corresponding controller to ensure diagnosability and the diagnoser to achieve on-line fault diagnosis. The design procedure is based on the elimination of the traces that go through indeterminate cycles in the diagnoser.

In problems that involve the two objectives of diagnosis and control, one is confronted with the trade-off of obtaining acceptable behaviour from a control perspective, while restricting the behaviour to ensure that the fault detection delay is minimized. Much remains to be understood regarding this trade-off.

Safe diagnosability for fault-tolerant supervision (Paoli & Lafortune, 2005) is concerned with detecting faults after their occurrence, but prior to the execution of a given set of forbidden executions, to prevent faults from developing into failures that could cause safety hazards. If the system is safe diagnosable, reconfiguration actions could be forced upon the detection of faults prior to the execution of unsafe behaviour. To extend safe diagnosability to the decentralized setting, Qiu and Kumar (2005a) have introduced the notion of safe codiagnosability and an approach to synthesis local diagnosers and use them for on-line diagnosis. Safe codiagnosability requires that when the system executes a faulty trace, there exists at least one diagnoser that can detect this within bounded delay, before the safety specification is violated.

Wen, Kumar, Huang, and Liu (2008) have developed a fault-tolerant controller that can force every post-fault behaviour to become equivalent to a non-faulty behaviour in a bounded number of steps.

A recent approach to fault-tolerant control (Paoli, Sartini, & Lafortune, 2011) concerns the design of a parameterized controller to update the control law upon detection of faulty behaviour on the basis of on-line diagnosis. The notion of safe controllability is introduced to steer the system away from forbidden zones after the occurrence of a fault. Active fault tolerance is achieved by safely continuing operation after faults to guarantee pre-specified (eventually degraded) performance objectives for the faulty system. The synthesized controller-diagnoser can safely detect faults and switch between the nominal control policy and a bank of reconfigured control policies.

5. Conclusion and future directions

Fault diagnosis of DES is an active scientific area that has been recently reinforced with many established and well-recognized formal methods and models. Many extensions of early results have been proposed to deal with a variety of modelling tools and system structures, and to improve design methods and algorithmic efficiency. These extensions have been accompanied with many adaptations of the notion of diagnosability and associated algorithms to construct diagnosers and verify diagnosability.

A major problem facing the diagnosis of DES is related to the complexity of calculations due to the curse of dimensionality. The reader can refer to Cassez (2012) for issues related to complexity and decidability for different fault diagnosis configurations of timed and untimed systems. To reduce the complexity of calculations, abstraction-based approaches can provide interesting means to optimize the search space for diagnosability (Grastien & Torta, 2011; Schmidt, 2010; Ye & Dague, 2010).

To improve the applicability and the diffusion of fault diagnosis methods of DES, it is important to develop easy-to-use general-purpose software tools to construct diagnosers and verify diagnosability (e.g., Lafortune, Ricker, & Genc, 2006; Clavijo, Basilio, & Carvalho, 2012), as well as tools for specific methodologies (e.g.,

Pencolé, Cordier, & Rozé, 2001). Providing practical diagnosis approaches for complex systems also requires the combination of DES-based diagnosis methods with artificial intelligence and machine learning techniques (e.g., Cordier et al., 2004; Lamperti & Zanella, 2003; Provan & Chen, 1998; Sayed-Mouchaweh, 2012) on the one hand, and with approaches from continuous systems to deal with hybrid system dynamics (Bayoudh & Travé-Massuyès, 2009) on the other hand. Moreover, designing a reliable, safe and secure system also requires developing a global approach to link diagnosis with control, identification, and predictive maintenance.

Finally, we point out that the property of *opacity* in computer security (see Bryans, Koutny, & Ryan, 2005), which is related to the inability of an external observer (intruder) to detect a "secret" about the system behaviour, is in some sense the dual of diagnosability: for opacity to hold, the secret should not be diagnosable from the viewpoint of the intruder. Recent works have explored this connection more formally (e.g., Lin, 2011), and many research opportunities lie ahead.

Acknowledgements

This work was supported by the CPER project AidCrisis, sponsored by the Champagne-Ardenne region and the French ministry of higher education and research. The second author wishes to acknowledge support from the US National Science Foundation. The authors wish to acknowledge the pertinent comments of Manuel Silva and of an anonymous reviewer, which helped to improve the paper.

References

Aghasaryan, A., Fabre, E., Benveniste, A., Boubour, R., & Jard, C. (1998). Fault detection and diagnosis in distributed systems: An approach by partially stochastic Petri nets. *Discrete Event Dynamic Systems, 8*(2), 203–231.

Athanasopoulou, E., Li, L., & Hadjicostis, C. (2006). Probabilistic failure diagnosis in finite state machines under unreliable observations. In *Proc. 8th international workshop on discrete event systems (WODES'06)*.

Athanasopoulou, E., & Hadjicostis, C. N. (2005). Probabilistic approaches to fault detection in networked discrete event systems. *IEEE Transactions Neural Networks, 16*(5), 1042–1052.

Basile, F., Chiacchio, P., & De Tommasi, G. (2009). An efficient approach for online diagnosis of discrete event systems. *IEEE Transactions Automatic Control, 54*(4), 748–759.

Basile, F., Chiacchio, P., & De Tommasi, G. (2008). Sufficient conditions for diagnosability of Petri nets. In *Proc. 9th international workshop on discrete event systems (WODES'08)* (pp. 436–442).

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

Basilio, J. C., Souza Lima, S. T., Lafortune, S., & Moreira, M. V. (2012). Computation of minimal event bases that ensure diagnosability. *Discrete Event Dynamic Systems, 22*, 2012.

Bavishi, S., & Chong, E. (1994). Automated fault diagnosis using a discrete event systems framework. In *Proc. 9th IEEE int. symp. intelligent contr.* (pp. 213–218).

Bayoudh, M., & Travé-Massuyès, L. (2009). An algorithm for active diagnosis of hybrid systems casted in the DES framework. In *Proc. 2nd IFAC workshop on dependable control for discrete systems*, (pp. 287–292).

Benveniste, A., Fabre, E., Haar, S., & Jard, C. (2003). Diagnosis of asynchronous discrete event systems: A net unfolding approach. *IEEE Transactions Automatic Control, 48*(5), 714–727.

Boel, R.K., & van Schuppen, J.H. (2002). Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers. In *Proc. 6th international workshop on discrete event systems (WODES'02)*.

Bouyer, P., Chevalier, F., & D'Souza, D. (2005). Fault diagnosis using timed automata. In *Proc. 8th International Conference on Foundations of Software Science and Computation Structures. LNCS* (3441, pp. 219–233). Springer.

Bryans, J. W., Koutny, M., & Ryan, P. Y. A. (2005). Modelling opacity using Petri nets. *Electronic Notes in Theoretical Computer Science, 121*, 101–115.

Cabasino, M. P., Giua, A., Possi, M., & Seatzu, C. (2011). Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems. *Control Engineering Practice, 19*(9), 989–1001.

Cabasino, M. P., Giua, A. N., & Seatzu, C. (2010). Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica, 46*(9), 1531–1539.

Caines, P., Greiner, R., & Wang, S. (1991). Classical and logic based dynamic observers for finite automata. *IMA Journal of Mathematical Control and Information, 8*, 45–80.

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

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

Carvalho, L. K., Lima, S. T. S., Moreira, M. V., Basilio, J. C., & Lafortune, S. (2013). Robust diagnosis of discrete-event systems against permanent loss of observations. *Automatica, 49*, 223–231.

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

Cassez, F. (2009). A note on fault diagnosis algorithms. In *Proc. 48th IEEE conference on decision and control and 28th Chinese control conference*.

Cassez, F. (2012). The complexity of codiagnosability for discrete event and timed systems. *IEEE Transactions Automatic Control, 57*(7), 1752–1764.

Cassez, F., & Tripakis, S. (2008). Fault diagnosis with static and dynamic observers. *Fundamenta Informaticae, 88*(4), 497–540.

Chakib, H., & Khoumsi, A. (2012). Multi-decision diagnosis: Decentralized architectures cooperating for diagnosing the presence of faults in discrete event systems. *Discrete Event Dynamic Systems, 22*(3), 333–380.

Chanthery, E., & Pencolé, Y. (2009). Monitoring and active diagnosis for discrete-event systems. In *Proc. 7th IFAC symposium on fault detection, supervision and safety of technical processes (SafeProcess'09)*.

Chen, Y.L., & Provan, G. (1997). Modelling and diagnosis of timed discrete event systems. In *Proc. American control conference* (pp. 31–36).

Chung, S. L. (2005). Diagnosing PN-based models with partial observable transitions. *International Journal of Computer Integrated Manufacturing, 18*, 158–169.

Cieslak, R., Desclaux, D., Fawaz, A., & Varaiya, P. (1988). Supervisory control of discrete-event processes with partial observations. *IEEE Transactions Automatic Control, 33*, 249–260.

Clavijo, L.B., Basilio, J.C., & Carvalho, L.K. (2012). DESLAB: A scientific computing program for analysis and synthesis of discrete-event systems. In *Proc. 11th international workshop on discrete event systems, (WODES 2012)*.

Contant, O., Lafortune, S., & Teneketzis, T. (2004). Diagnosis of intermittent faults. *Discrete Event Dynamic Systems, 14*(2), 171–202.

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

Cordier, M.-O., Dague, P., Lévy, F., Montmain, J., Staroswiecki, M., & Travé-Massuyès, L. (2004). Conflicts versus analytical redundancy relations: A comparative analysis of the model-based diagnostic approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man and Cybernetics – Part B, 34*(5), 2163–2177.

Dallal, E., & Lafortune, S. (2011). A framework for optimization of sensor activation using most permissive observers. In *Proc. 50th IEEE conference on decision and control* (pp. 2711–2717).

Debouk, R., Malik, R., & Brandin, B. (2002b). A modular architecture for diagnosis of discrete event systems. In *Proc. IEEE conference decision and control* (pp. 417–422).

Debouk, R., Lafortune, S., & Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete event dynamic systems, 10*, 33–86.

Debouk, R., Lafortune, S., & Teneketzis, D. (2002). On an optimization problem in sensor selection. *Discrete event dynamic systems, 12*, 417–445.

Dotoli, M., Fanti, M., & Mangini, A. (2009). Fault detection of DES by Petri nets and integer linear programming. *Automatica, 45*(11), 2665–2672.

Fabre, E., & Jezequel, L. (2010). On the construction of probabilistic diagnosers. In *Proc. 10th international workshop on discrete event systems (WODES 2010)*.

Fabre, E., & Benveniste, A. (2007). Partial order techniques for distributed discrete event systems: Why you cannot avoid using them. *Discrete Event Dynamic Systems, 17*(3), 355–403.

Fabre, E., Benveniste, A., Haar, S., & Jard, C. (2005). Distributed monitoring of concurrent and asynchronous systems. *Discrete Event Dynamic Systems, 15*(1), 33–84.

Fabre, E., Benveniste, S., & Jard, C. (2002). Distributed diagnosis for large discrete event dynamic systems. In *Proc. IFAC world congress*.

Fanti, M., Mangini, A.M., & Walter, U. (2011). Fault detection by labelled Petri nets and time constraints. In *Proc. 3rd international workshop on dependable control of Discrete Systems* (pp. 170–175).

Garcia, E., Correcher, A., Morant, F., Quiles, E., & Blasco, R. (2005). Modular fault diagnosis based on discrete event systems. *Discrete Event Dynamic Systems, 15*, 237–256.

Genc, S. (2008) Formal methods for intrusion detection of Windows NT attacks. In *Proc. 3rd annual symposium on information assurance (ASIA'08)* (pp. 71–79).

Genc, S., & Lafortune, S. (2006). Predictability in discrete-event systems under partial observation. In *Proc. 6th IFAC safeprocess symposium*.

Genc, S., & Lafortune, S. (2007). Distributed diagnosis of place-bordered Petri nets. *IEEE Transactions Automation Science and Engineering, 4*(2), 206–219.

Ghazel, M., Bigand, M., & Toguyéni, A. (2005). A temporal-constraint based approach for monitoring of DESs under partial observation. In *Proc. 16th IFAC world congress*.

Grastien, A., Torta, & G. (2011). A theory of abstraction for diagnosis of dicrete-event systems. In *Proc. symposium on abstraction, reformulation and approximation SARA'11* (pp. 50–57).

Hernandez-Flores, E., Lopez-Mellado, E., & Ramirez-Trevino, A. (2011). Diagnosticability analysis of partially observable deadlock-free Petri nets. In *Proc. 3rd int. workshop on dependable control of discrete systems* (pp. 176–181).

Idghamishi, A.M., & Zad, S.H. (2004). Fault diagnosis in hierarchical discrete-event systems. In *Proc. 43rd IEEE conference on decision and control* (pp. 63–68).

Jéron, T., Marchand, H., Pinchinat, S., & Cordier, M. (2006). Supervision patterns in discrete event systems diagnosis. In *Proc. 8th international workshop on discrete event systems (WODES 2006)*.

Jéron, T., Marchand, H., Genc, S., & Lafortune, S. (2008). Predictability of sequence patterns in discrete event systems. In *Proc. IFAC world congress* (pp. 537–543).

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

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

Jiang, S., Kumar, R., & Garcia, H. E. (2003). Optimal sensor selection for discrete-event systems under partial observation. *IEEE Transactions Automatic Control, 48*(3), 369–381.

Jiroveanu, G., & Boel, R. (2008). On-line monitoring of large Petri net models under partial observation. *Discrete Event Dynamic Systems, 18*, 323–354.

Jiroveanu, G., & Boel, R. (2010). The diagnosability of Petri net models using minimal explanations. *IEEE Transactions Automatic Control, 55*(7), 1663–1668.

Kumar, R., & Takai, S. (2008). Diagnosis prognosis of failures in discrete event systems. In *Proc. 9th international workshop on discrete event systems (WODES 2008)* (pp. 376–381).

Kumar, R., & Takai, S. (2010). Decentralized prognosis of failures in discrete event systems. *IEEE Transactions Automatic Control, 55*, 48–59.

Lafortune, S., Ricker, L., & Genc, S. (2006). DESUMA: A tool integrating GIDDES and UMDES. In *Proc. 8th international workshop on discrete event systems (WODES 2006)* (pp. 388–389).

Lamperti, G., & Zanella, M. (2003). *Diagnosis of active systems: Principles and techniques*. The Netherlands: Kluwer Academic Publishers.

Lefebvre, D., & Delherm, C. (2007). Diagnosis of DES with Petri net models. *IEEE Transactions Automation Science and Engineering, 4*(1), 114–118.

Lin, F. (1994). Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems, 4*(1), 197–212.

Lin, F. (2011). Opcaity of discrete event systems and its applications. *Automatica, 47*, 496–503.

Lin, W.C., Yoo, T.S., & Garcia, H.E., (2010). Sensor configuration selection for discrete-event systems under unreliable observations. In *Proc. 6th IEEE conference automation science and engineering* (pp. 477–484).

Lin, W. C., Garcia, H. E., & Yoo, T. S. (2013). A diagnoser algorithm for anomaly detection in DEDS under partial and unreliable observations: Characterization and inclusion in sensor configuration optimization. *Discrete Event Dynamic Systems, 23*, 61–91.

Lin, F., & Wonham, W. M. (1988). On observability of discrete-event systems. *Information Sciences, 44*, 173–198.

Lunze, J., & Schroder, J. (2001). State observation and diagnosis of discrete-event systems described by stochastic automata. *Discrete Event Dynamic Systems, 11*, 319–369.

Mahulea, C., Seatzu, C., Cabasino, M. P., & Silva, M. (2012). Fault diagnosis of discrete-event systems using continuous Petri nets. *IEEE Trans SMC: Part A, 42*(4), 970–984.

Miyagi, P. E., & Riascos, L. A. M. (2010). Modeling and analysis of fault-tolerant systems for machining operations based on Petri nets. *Control Engineering Practice, 14*(4), 397–408.

Ozveren, C. M., & Willsky, A. S. (1990). Observability of discrete event dynamic systems. *IEEE Transactions Automatic Control, 35*(7), 797–806.

Pandalai, D. N., & Holloway, L. E. (2000). Template languages for fault monitoring of timed discrete event processes. *IEEE Transactions Automatic Control, 45*(5), 868–882.

Paoli, A., & Lafortune, S. (2005). Safe diagnosability for fault tolerant supervision of discrete-event systems. *Automatica, 41*(8), 1335–1347.

Paoli, A., & Lafortune, S. (2008). Diagnosability analysis of a class of hierarchical state machines. *Discrete Event Dynamic Systems, 18*, 385–413.

Paoli, A., Sartini, M., & Lafortune, S. (2011). Active fault tolerant control of discrete event systems using online diagnostics. *Automatica, 47*, 639–649.

Pencolé, Y. (2004). Diagnosability analysis of distributed discrete event systems. In *Proc. international workshop on principles of diagnosis (DX'04)* (pp. 173–178).

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

Pencolé, Y., Cordier, M.-O., & Rozé, L. (2001). A decentralized model-based diagnostic tool for complex systems. In *Proc. 13th IEEE int. conf. on tools with artif. intel. (IC-TAI'01)* (pp. 95-102).

Pencolé, Y., & Subias, A. (2009). A chronicle-based diagnosability approach for discrete timed-event systems: Application to web-services. *Journal of Universal Computer Science, 15*(17), 3246–3272.

Philippot, A., Sayed-Mouchaweh, M., & Carré-Ménétrier, V. (2009). Discrete event model-based approach for fault detection and isolation of manufacturing systems. In *Proc. 2nd IFAC workshop on dependable control of discrete systems* (pp. 69–74).

Prock, J. (1991). A new technique for fault detection using Petri nets. *Automatica, 27*, 239–245.

Provan, G., & Chen, Y.-L. (1998). Diagnosis of timed discrete event systems using temporal causal networks: Modeling and analysis. In *Proc. 4th international workshop on discrete event systems (WODES '98)* (pp. 152–154).

Qiu, W., & Kumar, R. (2005a). Decentralized diagnosis of event-driven systems for safely reacting to failures. In *Proc. IFAC world congress* (pp. 140–145).

Qiu, W., & Kumar R. (2005b). Distributed Diagnosis under bounded-delay communication of immediately forwarded local observations. In *Proc. American control conference* (pp. 1027–1032).

Qiu, W., & Kumar, R. (2006). Decentralized failure diagnosis of discrete event systems. *IEEE Transactions on Systems, Man and Cybernetics: Part A, 36*(2), 628–643.

Ramadge, P.J. (1986). Observability of discrete-event systems. In *Proc. 25th IEEE conference decision and control* (pp. 1108–1112).

Ramirez-Trevino, A., Ruiz-Beltran, E., Rivera-Rangel, I., & Lopez-Mellado, E. (2007). Online fault diagnosis of discrete event systems. A Petri net-based approach. *IEEE Transactions Automation Science and Engineering, 4*(1), 31–39.

Ribot, P., Pencolé, Y., & Combacau, M. (2008). Design requirements for the diagnosability of distributed discrete event systems. In *DX'08, 19th international workshop on principles of diagnosis*.

Ricker, L., & Fabre, E. (2000). On the construction of modular observers and diagnosers for discrete event systems. In *Proc. 39th IEEE conference decision and control* (pp. 2240–2244).

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

Roth, M., Lesage, J.-J., & Litz, L. (2011). The concept of residuals for fault localization in discrete event systems. *Control Engineering Practice, 19*, 978–988.

Rozé, L., & Cordier, M.-O. (2002). Diagnosis discrete-event systems: Extending the diagnoser approach to deal with telecommunication networks. *Discrete Event Dynamic Systems, 12*, 43–81.

Sampath, M., Godambe, A., Jackson, E., & Mallow, E. (2000). Combining qualitative and quantitative reasoning – A hybrid approach to failure diagnosis of industrial systems. In Proc. IFAC safeprocess conf. (pp. 494–501).

Sampath, M., Lafortune, S., & Teneketzis, D. (1998). Active diagnosis of discrete event systems. *IEEE Transactions Automatic Control, 43*(7), 908–929.

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

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

Sayed-Mouchaweh, M. (2012). Decentralized fault free model approach for fault detection and isolation of discrete event systems. *European Journal of Control, 18*(1), 82–93.

Sayed-Mouchaweh, M., Philippot, A., & Carré-Ménétrier, V. (2008). decentralized diagnosis by Boolean discrete event system model: Application on manufacturing systems. *International Journal of Production Research, 46*(19), 5469–5490.

Schmidt, K. (2010). Computation of projections for the abstract-based diagnosability verification. In *Proc. 10th international workshop on discrete event systems (WODES 2010)*.

Schneider, S., Litz, L., & Danancher, M. (2011). Timed residuals for fault detection and isolation in discrete event systems. In *Proc. 3rd international workshop on dependable control of discrete systems*.

Sengupta, R. (1998). Diagnosis and communication in distributed systems. In *Proc. 4th international workshop on discrete event systems (WODES'98)* (pp. 144–151).

Sengupta, R., & Tripakis, S. (2002). Decentralized diagnosability of regular languages is undecidable. In *Proc. 41st IEEE conference on decision and control* (pp. 423–428).

Shu, S. Huang, Z., & Lin, F. (2010). On-line detection and sensor activation for discrete event systems. In *Proc. 10th international workshop on discrete event systems (WODES 2010)* (pp. 197–202).

Şimşek, H.T., Sengupta, R., Yovine, S., & Eskafi, F. (1999). Fault diagnosis for intra-platoon communication. In *Proc. IEEE conf. decision and control*.

Sreenivas, V. S., & Jafari, M. A. (1993). Fault detection and monitoring using time Petri nets. *IEEE Transactions Systems, Man and Cybernetics, 23*, 1155–1162.

Su, R., & Wonham, W.M. (2004). A model of component consistency in distributed diagnosis. In *Proc. 7th international workshop on discrete event systems (WODES'04)* (pp. 427-432).

Su, R., Wonham, W.M., Kurien, J., & Koutsoukos, X. (2002). Distributed diagnosis for qualitative systems. In *Proc. 6th international workshop on discrete event systems (WODES'02)* (pp. 169–174).

Su, R., & Wonham, W. M. (2005). Global and local consistencies in distributed fault diagnosis for discrete-event systems. *IEEE Transactions Automatic Control, 50*, 1923–1935.

Su, R., & Wonham, W. M. (2006). Hierarchical fault diagnosis for discrete-event systems under global consistency. *Discrete Event Dynamic Systems, 16*, 39–70.

Takai, S. (2010) Robust failure diagnosis of partially observed discrete event systems. In *Proc. 10th international workshop on discrete event systems (WODES'10)*.

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

Takai, S., & Kumar, R. (2010). Decentralized diagnosis for nonfailures of discrete event systems using inference-based ambiguity management. *IEEE Transactions on Systems, Man and Cybernetics Part A, 40*(2), 406–412.

Thorsley, D., Yoo, T.S., & Garcia, H.E. (2008). Diagnosability of stochastic discrete-event systems under unreliable observations. In *Proc. American control conference* (pp. 1158–1165).

Thorsley, D., & Teneketzis, D. (2005). Diagnosability of stochastic discrete-event systems. *IEEE Transactions Automatic Control, 50*(4), 476–492.
Thorsley, D., & Teneketzis, D. (2007). Active acquisition of information for diagnosis and supervisory control of DES. *Discrete Event Dynamic Systems, 17*, 531–583.
Tripakis, S. (2002). Fault diagnosis for timed automata. In *Proc. int. conf. on formal techniques in real time and fault tolerant systems. LNCS* (Vol. 24, pp. 205–224). Springer.
Ushio, T., Onishi, L., & Okuda, K. (1998). Fault detection based on Petri net models with faulty behaviors. In *Proceedings SMC'98 IEEE international conference on systems, man, and cybernetics* (pp. 113–118).
Velilla, S., & Silva, M. (1988). The SPY: a mechanism for safe implementation of highly concurrent systems. In *Proc. 15th IFAC/IFIP workshop on real-time programming* (pp. 75–81).
Viswanadham, N., & Johnson, T.L. (1988). Fault detection and diagnosis of automated manufacturing systems. In *Proc. IEEE conference decision and control* (pp. 2301–2306).
Wang, X., Chattopadhyay, I., & Ray, A., (2004). Probabilistic fault diagnosis in discrete event systems. In *Proc. 43rd IEEE conference decision and control* (pp. 4794–4799).
Wang, Y., Yoo, T.S., & Lafortune, S. (2005). Decentralized diagnosis of discrete event systems using unconditional and conditional decisions. In *Proc. 44th conference decision and control* (pp. 6298–6304).
Wang, W., Lafortune, S., Girard, A. R., & Lin, F. (2010). Optimal sensor activation for diagnosing discrete event systems. *Automatica, 46*, 1165–1175.
Wang, Y., Yoo, T. S., & Lafortune, S. (2007). Diagnosis of discrete event systems using decentralized architectures. *Discrete Event Dynamic Systems, 17*(2), 233–263.
Wen, Y., Li, C., & Jeng, M. (2005). A polynomial algorithm for checking diagnosability of Petri nets. In *Proc. IEEE SMC'05 conference on systems, man, and cybernetics* (pp. 2542–2547).
Wen, Q., Kumar, R., Huang, J., & Liu, H. (2008). A framework for fault-tolerant control of discrete event systems. *IEEE Transactions Automatic Control, 53*(8), 1839–1849.
Wu, Y., & Hadjicostis, C. N. (2005). Algebraic approach for fault identification in discrete-event systems. *IEEE Transactions Robotics and Automation, 50*(12), 2048–2053.
Ye, L., & Dague, P. (2010). An optimized algorithm for diagnosability of component-based systems. In *Proc. 10th international workshop on discrete event systems (WODES'10)*.
Yoo, T. S., & Lafortune, S. (2002a). Polynomial-time verification of diagnosis of partially-observed discrete event systems. *IEEE Transactions Automatic Control, 47*, 1491–1495.
Yoo, T. S., & Lafortune, S. (2002b). NP-completeness of sensor selection problems arising in partially observed discrete-event systems. *IEEE Transactions Automatic Control, 47*(9), 1495–1499.
Zad, S. H., Kwong, R. H., & Wonham, W. M. (2003). Fault diagnosis in discrete-event systems: Framework and model reduction. *IEEE Transactions Automatic Control, 48*(7), 1199–1212.
Zad, S. H., Kwong, R. H., & Wonham, W. M. (2005). Fault diagnosis in discrete-event systems: Incorporating timing information. *IEEE Transactions Automatic Control, 50*(7), 1010–1015.
Zaytoon, J., & Sayed Mouchaweh, M. (2012). Discussion on fault diagnosis methods of discrete event systems. In *Proc. 11th international workshop on discrete event systems (WODES'12)*.
Zhou, C., Kumar, R., & Sreenivas, R. (2008). Decentralized modular diagnosis of concurrent discrete event systems. In *Proc. 9th international workshop on discrete event systems (WODES'08)* (pp. 388–393).

Janan Zaytoon (BSc Eng./1983, MSc Eng./1986, DEA/1988, PhD/1993, Habilitation/1997) is a Professor at the University of Reims Champagne-Ardenne. He was the founding Director of the CReSTIC Research Centre (involving 140 researchers) of the University. He is the President-Elect of the International Federation of Automatic Control (IFAC) and is the Director of the French national research network/group "GDR MACS of CNRS", which involves all the researchers in the fields of Automatic Control Systems and Production systems in France (about 2000 researchers and PhD students). His scientific activity involves many theoretical, methodological and applied aspects of automatic control systems, including: discrete-event systems, hybrid systems, intelligent control of complex and hybrid systems. Janan Zaytoon is the Editor-in-Chief of *Nonlinear Analysis: Hybrid Systems*, and Associate Editor of *Control Engineering Practice* and *Discrete Event Dynamic Systems*. He served as General Chair, Chair/Co-Chair of the program and/or organization committee of 15 international conferences and 12 national conferences. He has been Invited plenary speaker for 5 international conferences and is a member of the Council of French Universities.

Stéphane Lafortune is a professor in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. He obtained his degrees from Ecole Polytechnique de Montréal (B.Eng), McGill University (M.Eng), and the University of California at Berkeley (PhD), all in electrical engineering. Dr. Lafortune is a Fellow of the IEEE (1999). His research interests are in discrete event systems and include multiple problem domains: modelling, diagnosis, control, optimization, and applications to computer systems. He is co-developer of the software packages DESUMA and UMDES. He co-authored, with C. Cassandras, the textbook *Introduction to Discrete Event Systems* (2nd Edition, Springer, 2008). He received the Presidential Young Investigator Award from the U.S. National Science Foundation in 1990 and the George S. Axelby Outstanding Paper Award from the Control Systems Society of the IEEE in 1994 (for a paper co-authored with S.L. Chung and F. Lin) and in 2001 (for a paper co-authored with G. Barrett).