Intermittent Fault Diagnosis of Industrial Systems in a Model-Checking Framework

Abderraouf Boussif Mohamed Ghazel
Univ. Lille Nord de France F-59000, Lille, France
IFSTTAR, Cosys/Estas F-59666, Villeneuve d'Ascq, France
Email: {abderraouf.boussif, mohamed.ghazel}@ifsttar.fr

***Abstract*—In this paper, a formal verification approach for diagnosability analysis of intermittent faults is proposed. In this approach, the industrial systems are abstracted as discrete-event systems (DES) and modeled by finite state automata (FSA), then a model-checking framework is set to deal with diagnosability issues. Intermittent faults are defined as faults that can automatically recover once they occur. We first revisit two existing definitions of diagnosability of intermittent faults, regarding the occurrence of faults and their normalization (i.e., disappearance of faults). Then, necessary and sufficient conditions are developed based on the twin plant construction, and reformulated as linear temporal logic (LTL) formulas in order to use model-checking for actual verification. A benchmark is used to illustrate the contributions discussed and to assess the efficiency and the scalability of the proposed approach.** [1]

I. Introduction

automated monitoring and fault diagnosis is a crucial task in complex industrial systems. This problem has received considerable attention in both artificial intelligence and control engineering domains. In particular, an increasing amount of work has been devoted to diagnosis of discrete-event systems during the last 15 years as witnessed by the survey work in [1].

One of the main issues in fault diagnosis that must be addressed firstly is diagnosability investigation. The intention of studying diagnosability is to determine accurately whether any predetermined failures or class of failures can be detected and identified within a finite delay (as soon as possible) after the failures happened based on the delivered observations.The formal definition of diagnosability was first introduced in the seminal work of [2], where a systematic method to check diagnosability based on a diagnoser construction in an *event-based* approach was developed. A similar work based on a diagnoser construction in a *state-based* approach was proposed in [3]. An application to transportation systems is reported in [4]. Improvements in terms of complexity, based on the *verifier* and the *twin plant* structures were introduced later in [5], [6], where the basic idea is to build an intermediate structure by constructing a parallel composition of the system model with itself.

The methodologies referenced above for permanent faults are no longer adequate in the context of intermittent faults. Indeed, the case of intermittent faults shows some subtle configuration compared to the case of permanent failure. Consequently, some DES-based frameworks have been proposed to handle intermittent faults. One of the first contributions was made by [7], where a *state-based* DES modeling for the so called "*repeated faults*" was introduced. Various notions of diagnosability were discussed and polynomial algorithms for checking these properties were provided [8], [9]. These works focused on diagnosing how many times a fault has occurred. Hence, they do not deal with the diagnosability of intermittent faults in the sense that they do not discuss whether the fault occurrences can be accurately detected and the status of the system determined within a finite delay. Dealing with diagnosability of intermittent faults in this sense was first discussed by [10], [11]. In these studies, an FSA *event-based* approach is used, i.e., the faults and their recovery are considered as unobservable events. The purpose of these works is to determine which faults are present in the system and which faults have occurred and been recovered. Contant's work can be seen as an extension of the seminal work on diagnosability of permanent failures [2], where some modification are brought in terms of fault modeling and diagnoser construction. A similar work to [10] is reported in [12] with an illustration through an industrial process.

An extension of the *state-based* DES framework, introduced in [3], was proposed in [13] to deal with intermittent faults. Two notions of diagnosability were introduced, one for detecting the occurrence of a fault, and the other for detecting its normalization (recovery). The diagnoser is constructed in the same way as in [3] with the same time-complexity. Necessary and sufficient conditions for each notion are developed, and an algorithm to check diagnosability is provided.

Model-checking techniques [14], which have been developed for efficiently verifying complex dynamic systems, have been exploited to deal with diagnosis issues of permanent failures [15], [16]. Particularly, the seminal work [15] has proposed a formal model-checking framework for diagnosability analysis, where this issue is reduced to a reachability analysis problem in the twin plant using a CTL/LTL formulas to express the diagnosability conditions. Extensions and experimentation have been discussed in our previous work [16].

In this paper, we propose a formal verification approach for diagnosability analysis of intermittent faults using model-checking, based on the twin plant construction [5], and the reformulation of the diagnosability issues as temporal logic formulas workable with model-checking. We first revisit two existing definitions of diagnosability, proposed in [10], re-

[1]978-1-5090-0382-2/16/$31.00

garding the occurrence of faults and their recoveries, then necessary and sufficient conditions, for each definition, are first developed and then reformulated by means of LTL formulas in order to use model-checking for verification.

The paper is organized as follows: Section 2 introduces the considered system model and the modeling of intermittent faults as well as some related notions and notations. In section 3, two existing definitions of diagnosability are revisited. Section 4 presents the twin plant construction and gives the necessary and sufficient conditions for each definition. The formulation of diagnosability of intermittent faults as a model-checking issue, and necessary and sufficient conditions as LTL specifications are discussed in Section 5. We illustrate the concepts discussed through a benchmark in Section 6. Finally, Section 7 draws some concluding remarks and points to future directions.

II. Preliminaries

A. System model

The approach introduced in this paper applies to industrial systems, abstracted as discrete-events systems, and modeled by finite state automata (FSA). Let $G = \langle X, \Sigma, \delta, x_0 \rangle$ be an FSA 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 model G accounts for the normal and faulty behavior of the system. The system behavior is described by the prefix-closed language $L \in \Sigma^*$ generated by G, where Σ^* denotes the Kleene-closure of set Σ.

The partial observability issue plays a central role in fault diagnosis of industrial systems. In this regard, some events in Σ are observable, i.e., their occurrences can be observed, while the rest are unobservable. Thus, event set Σ can be partitioned as $\Sigma = \Sigma_o \biguplus \Sigma_u$, where Σ_o denotes the set of observable events and Σ_u the set of unobservable events.

In the context of diagnosis of intermittent faults, let $\Sigma_f \subseteq \Sigma_u$ denote the set of fault events and let $\Sigma_r \subseteq \Sigma_u$ denote the set of fault reset events. Faults and their recoveries are usually represented using unobservable events, since their detection and diagnosis would be trivial if they were observable. Thus, the set of fault events (resp. the set of reset events) is partitioned into disjoint fault classes $\Sigma_f = \Sigma_{f_1} \biguplus \Sigma_{f_2} \biguplus \cdots \biguplus \Sigma_{f_m}$, where $\Sigma_{f_i} (i = 1, 2, \ldots, m)$ denotes one class of faults (resp. $\Sigma_r = \Sigma_{r_1} \biguplus \Sigma_{r_2} \biguplus \cdots \biguplus \Sigma_{r_m}$, where $\Sigma_{r_i} (i = 1, 2, \ldots, m)$ denotes the recovering class of faults in Σ_{f_i}).

Let us recall some standard notations and operations that will be used in the sequel. The empty event-trace is denoted by ϵ. Let $s \in \Sigma^*$ be an event-trace. We denote by L/s the post-language of L upon s, i.e., $L/s := \{t \in \Sigma^* : s.t \in L\}$. We will use $\psi(\Sigma_{f_i})$ to denote the set of event-traces in L that end with faulty events in Σ_{f_i}. That is, $\psi(\Sigma_{f_i}) := \{s\sigma_{f_i} \in L : \sigma_{f_i} \in \Sigma_{f_i}\}$. Similarly, we will use $\psi(\Sigma_{r_i})$ to denote the set of event-traces in L that end with reset events in Σ_{r_i}. That is, $\psi(\Sigma_{r_i}) := \{s\sigma_{r_i} \in L \mid \sigma_{r_i} \in \Sigma_{r_i}\}$. Consider $\sigma \in \Sigma$ and $s \in \Sigma^*$. We use the notation $\sigma \in s$ to denote the fact that σ is an event in trace s, By abuse of notation, we write $\Sigma_f \in s$ to denote that a fault event from Σ_f is an event in trace s (i.e., $\exists f_i \in \Sigma_{f_i}\ s.t.\ f_i \in s$).

To reflect the limitation in terms of observation, we define the projection operator as a function $P : \Sigma^* \to \Sigma_o^*$. 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 $s \in \Sigma^*$, $\sigma \in \Sigma$. That is, P simply erases the unobservable events in any event-trace. The inverse projection operation P_L^{-1} is defined by $P_L^{-1}(y) = \{s \in L(G) : P(s) = y\}$. The projection operator can be extended to language L by applying the projection to all traces of L. Therefore, if $L \subseteq \Sigma^*$, then $P(L) = \{t \in \Sigma_o^* : (\exists\, s \in L)\,[P(s) = t]\}$.

Let $G_1 = \langle X_1, \Sigma_1, \delta_{G_1}, x_{0_1} \rangle$ and $G_2 = \langle X_2, \Sigma_2, \delta_{G_2}, x_{0_2} \rangle$ denote two finite state automata. The strict synchronous composition of G_1 and G_2 produces an automaton $G_{G_1 \| G_2} = \langle X_1 \times X_2, \Sigma_1 \cap \Sigma_2, \delta_{G_1 \| G_2}, (x_{0_1}, x_{0_1}) \rangle$, where $\delta_{G_1 \| G_2} \subseteq (X_1 \times X_2) \times (\Sigma_1 \cap \Sigma_2) \times (X_1 \times X_2)$ and $(x_1', x_2') \in \delta_{G_1 || G_2}((x_1, x_2), \sigma)$ if $x_1' \in \delta_{G_1}(x_1, \sigma)$ and $x_2' \in \delta_{G_2}(x_2, \sigma)$.

Finally, we define the non-deterministic automaton $G' = \langle X_o, \Sigma_o, \delta_{G'}, x_0 \rangle$ as "*the constructed generator*"of language $L(G') = P(L(G))$. The elements X_o, Σ_o, and x_0 are as defined before. The transition relation of G' is given by $\delta_{G'} \subseteq (X_o \times \Sigma_o \times X_o)$ and is defined as follows: $(x, \sigma, x') \in \delta_{G'}$ if $\delta(x, s) = x'$ s.t. $s = (\sigma_1, \sigma_2, \ldots, \sigma_n = \sigma)$: $\sigma_i \in \Sigma_u (i = 1, 2, \ldots, n-1)$ and $\sigma_n \in \Sigma_o$.

In the remainder of this paper, we consider a given system modeled by a finite state automaton $G = \langle X, \Sigma, \delta, x_0 \rangle$. For the sake of simplicity, here only one class of fault event Σ_f and its corresponding class Σ_r of reset events are considered.

B. Modeling of intermittent faults

Intermittent faults are non-permanent, in the sense that each occurrence of fault is followed by its reset within a finite delay. Regarding the system status, an intermittent fault takes the system from a normal state to a faulty state (by the occurrence of a fault event), and then the system is taken again into a recovery state after a finite delay (by the occurrence of the corresponding reset event).

In order to capture these changes in the status of the system, we use the supervision pattern Ω [17], shown in Figure 1, which is a label automaton that models the dynamic behavior of the system regarding intermittent faults. One can note that automaton Ω plays the role of the label function, which is usually used in fault diagnosis [2].

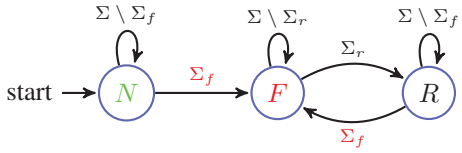

Fig. 1. The label automaton Ω

Actually, when the label automaton Ω is in state N (N for normal status), this means that the system executes a normal

behavior, which indicates that no event from Σ_f has occurred. However, when a fault event occurs, the label automaton Ω moves to state F (F for faulty status), and remains in that state for as long as the system executes a faulty behavior. If the fault is recovered, by the occurrence of a reset event, Ω changes to state R (for recovery status), where it stays while, the system continues to execute a non-faulty behavior. As we deal with intermittent faults, the system can execute again a fault event. Then the label automaton Ω can return to state F.

In order to keep track of the occurrence of faults and their corresponding resets along the system's evolution, we compute automaton G_ℓ as the parallel composition of automata G and Ω ($G_\ell = G \parallel \Omega$). In fact, the states of G_ℓ are the states of automaton G enriched with labels N, F, or R. The following example illustrates these notions.

Example 1: Consider automaton G, shown in Figure 2(a). The sets of observable and unobservable events are $\Sigma_o = \{a, b, c\}$ *and* $\Sigma_u = \{u, f, r\}$*, respectively. In addition,* $\Sigma_f = \{f\}$ *and* $\Sigma_r = \{r\}$*. Automaton* $G_\ell = G \parallel \Omega$ *is depicted in Figure 2 (b).*

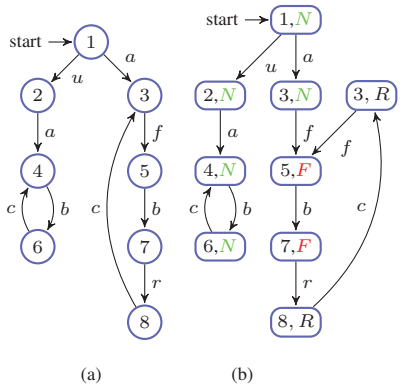

Fig. 2. (a) Automaton G, (b) Automaton G_ℓ of Example 1.

Regarding the diagnosis of intermittent faults, we can infer that states of automaton G_ℓ can be partitioned into three types of state: 'Normal', 'Faulty', and 'Recovered' states, which can be identified using *fault-assignment function*:

$$\Psi : X \to 2^{\{N,F,R\}}.$$

III. Notions of Diagnosability

A. Assumptions

Besides the well-known assumptions considered for the diagnosis of permanent faults [2], i.e., that language $L(G)$ is live (i.e., there is an output transition defined at each state x in X) and it does not exist in G any cycle of only unobservable events. We add the following assumptions on the system behavior,

(A1) Each fault event σ_f has its corresponding reset event σ_r. Recall that both are unobservable events.

(A2) At least one observable event exists between the occurrence of a fault event σ_f and its corresponding reset event σ_r and between the occurrence of a reset event σ_r and a new occurrence of the fault event σ_f.

(A3) Each occurrence of fault event σ_f is followed by the occurrence of its corresponding reset event σ_r within a finite delay and each occurrence of a reset event σ_r is followed by a new occurrence of the fault event σ_f within a finite delay.

B. Definitions of diagnosability

In this paper, we discuss two definitions of diagnosability, firstly introduced in [10], regarding the detection of the occurrence of a fault and the detection of its recovery without necessarily identifying, at any moment, whether the current status of the system is precisely known (i.e., fault is present or not). In [10], these definitions are denoted as Type-O-diagnosability and Type-I-diagnosability respectively. Here we call them F-diagnosability and R-diagnosability instead.

Definition 1: (F-diagnosability)

An FSA G is said to be F-diagnosable w.r.t. projection function P, fault class Σ_f and reset event class Σ_r, if the following holds:

$$(\exists n \in \mathbb{N}) \quad [\forall s \in \psi(\Sigma_f)] \quad (\forall t \in L/s) \quad [\| t \| \geq n \Rightarrow D_F]$$

where diagnosability condition D_F is:

$$\omega \in [P_L^{-1}(P(s.t))] \Rightarrow (\Sigma_f \in \omega)$$

F-diagnosability, where F stands for fault occurrences, has the following meaning: for any event-trace s ending with a fault event in Σ_f, and t any continuation of s, then, $n \in \mathbb{N}$ exists such that, after the occurrence of at most n events, it is possible to detect the occurrence of the fault based on the captured observation. This implies that all the event-traces indistinguishable from $s.t$ contain at least one fault from Σ_f.

Example 2: Let us take automaton G of Example 1, shown in Figure 2, and consider infinite execution $\pi = 1, a, (3, f, 5, b, 7, r, 8, c)^*$. Let the infinite event-trace, corresponding to this execution, be noted $s.t$ with, $s = af$ (we can see that $s \in \Psi(\Sigma_f)$), and $t = (brcf)^*$. The resulting observed event-trace is $P(s.t) = a(bc)^*$. Moreover, there exists, in automaton G, an infinite execution $\pi' = 1, u, 2, a, (4, b, 6, c)^*$ and its corresponding observed event-trace is $\omega = a(bc)^*$. One can see that π' shares the same observed event-trace with π, i.e., $\omega = P(s.t)$. Thus, according to Definition 1, there is no bound n, such that after this limit, one can always deduce that fault event f has occurred. Therefore, G is not F-diagnosable.

As we deal with intermittent faults, each fault occurrence is followed later on by its corresponding reset. Then, it is also interesting to discuss the diagnosability of the recovery occurrence. Namely, this consists in checking whether we can detect within a finite delay that the system moves to its recovery behavior after the fault has been recovered. Hereafter, we introduce R-diagnosability which represents, in some way, the dual notion of F-diagnosability.

Definition 2: (R-diagnosability)

An FSA G is said to be R-diagnosable w.r.t. projection function P, fault class Σ_f and reset event class Σ_r, if the following holds:

$$(\exists n \in \mathbb{N}) \quad [\forall s \in \psi(\Sigma_r)] \quad (\forall t \in L/s) \quad [\| t \| \geq n \Rightarrow D_R]$$

where diagnosability condition D_R is:

$$\omega \in [P_L^{-1}(P(s.t))] \Rightarrow (\Sigma_r \in \omega)$$

R-diagnosability, where R stands for reset event occurrences, has the following meaning: for any event-trace s ending with a reset event in Σ_r (which means that at least one fault has occurred and recovered) and t any continuation of s, then, after at most n events it is possible to detect the fault recovery based on the captured observation. This implies that all the event-traces indistinguishable from $s.t$ have experienced at least one occurrence of a fault and its recovery. As already underlined, R-diagnosability does not guarantee identifying with certainty whether the system is in a recovery status or not at any moment (i.e., all the state-traces which share the same event-trace reach recovery states at the same moment).

Example 3: Let us take again automaton G of Example 1 (figure 2), and consider infinite execution $\pi = 1, a, (3, f, 5, b, 7, r, 8, c)^*$. Let the infinite event-trace, corresponding to this execution, be noted $s.t$ with, $s = afbr$ (we can see that $s \in \Psi(\Sigma_r)$) and $t = (cfbr)^*$. The resulting observed event-trace is $P(s.t) = a(bc)^*$. Moreover, There exists, in automaton G, an infinite execution $\pi' = 1, u, 2, a, (4, b, 6, c)^*$ and its corresponding observed event-trace is $\omega = a(bc)^*$. One can see that π' shares the same observed event-trace with π, i.e., $\omega = P(s.t)$. Thus, according to Definition 2, there is no bound n, such that, after this limit, one can always deduce that reset event r has occurred. Therefore, G is not R-diagnosable.

IV. Diagnosability Analysis of Intermittent Faults

The procedure we discuss for analyzing diagnosibility of intermittent faults will be carried out by combining the twin plant construction method [5], and some extensions that we develop, based on the LTL model-checking reformulation of diagnosability [15], [16]. In this section, we first recall the twin plant construction, and later we develop the necessary and sufficient condition for each notion of diagnosability introduced in the previous section.

A. Twin plant construction

The twin plant simply consists of two synchronized copies of generator G' of system model G, i.e., the parallel system event-traces are synchronized on the observable events. Thus, any event-trace in the twin plant corresponds to a pair of event-traces in the system model that share the same observation.

Definition 3: (Twin Plant)

A twin plant of G is an FSA $\mathcal{P} = \langle \mathcal{Q}, \Sigma_o, \Gamma, \amalg_0 \rangle$, where,

- $\mathcal{Q} \subseteq \{(x, x') \mid x, x' \in X_o \}$ is the set of states.
- Σ_o the set of the (observable) events.
- $\Gamma \subseteq \mathcal{Q} \times \Sigma_o \times \mathcal{Q}$ is the partial transition relation s.t. $(q, \sigma, q') \in \Gamma$, with $q = (x_1, x_2)$, and $q' = (x'_1, x'_2)$ if and only if (x_1, σ, x'_1), $(x_2, \sigma, x'_2) \in \delta_o$.
- $q_0 = (x_0 \times x_0) \in \mathcal{Q}$ is the initial state.

In order to simplify the twin plant construction, we perform the synchronous composition directly on constructed generator G'_ℓ, which allows us to preserve label tracking. Thus, the *fault-assignment* function is extended as follows:

$$\Psi : (X_o, X_o) \to \{N, F, R\} \times \{N, F, R\}$$

Hence, there are different types of state that can be distinguished in the twin plant. Hereafter, only those which will be used to develop necessary and sufficient conditions, are defined.

Definition 4: (Twin plant state types)

- 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: is a state $q = (x, x') \in \mathcal{Q}$, such that $\Psi(q) = (N, F)$. FN-state is defined similarly.
- NR-state: is a state $q = (x, x') \in \mathcal{Q}$, such that $\Psi(q) = (N, R)$. RN-state is defined similarly.
- $N1$-state: is a state $q = (x, x') \in \mathcal{Q}$, such that $\Psi(q) = (N, \triangle)$. with $\triangle \in \{N, F, R\}$.

B. Necessary and sufficient conditions

In a previous work [16], we have dealt with diagnosability of permanent faults using a twin plant-based structure in model-checking framework. The necessary and sufficient condition for diagnosability was the absence of *"infinite critical pairs"* in the constructed twin plant. This means the absence of cycles which are composed only of FN(or NF)-states. In the same way, we formalize a necessary and sufficient condition for the diagnosability of intermittent faults. In order to do so, we need to introduce the following definitions,

Definition 5: (F-confused cycle) *is a cycle $\pi = (q_1, \sigma_1, q_2, \ldots, q_n, \sigma_n, q_{n+1} = q_1)$, in the twin plant, s.t. $\forall\, 1 \leq i \leq n$, q_i is an $N1$-state, and $\exists\, 1 \leq j \leq n$, s.t. $q_j \in \pi$ is an NF-state.*

An *F-confused cycle* in twin plant corresponds to two cycles on the system model (automaton G) which generate the same observed trace, such that the first one has no fault event (a fault-free cycle) and the second one contains, at least, one fault event (which is depicted by the existence of an NF-state).

Definition 6: (R-confused cycle) *is a cycle $\pi = (q_1, \sigma_1, q_2, \ldots, q_n, \sigma_n, q_{n+1} = q_1)$, in the twin plant, s.t. $\forall\, 1 \leq i \leq n$, q_i is an $N1$-state, and $\exists\, 1 \leq j \leq n$, s.t. $q_j \in \pi$ is an NR-state.*

After having set up the necessary notions, we now establish the necessary and sufficient conditions for diagnosability of intermittent faults.

Theorem 1: (Necessary & sufficient conditions) *A system model G, w.r.t a projection function P, a class of fault events Σ_f and its corresponding class of reset events Σ_r, is:*

1) *F-diagnosable, if and only if no F-confused cycle exists in its corresponding twin plant.*
2) *R-diagnosable, if and only if no R-confused cycle exists in its corresponding twin plant.*

The proof of the theorem is omitted here due to lack of space.

Proposition 1: Let an automaton G satisfy assumptions (A1),(A2), and (A3). Then, G is F-diagnosable if and only if G is R-diagnosable.

Similar arguments as in [10] can be used to prove this corollary.

V. Diagnosability Verification using Model-Checking

A. Model-Checking

Model-checking is an automatic formal verification technique that is widely applied for the design of complex dynamic systems [14]. It allows verifying whether the system behavior (modeled by a Kripke Structure) satisfies a given property expressed as a temporal logic formula, using efficient algorithms based on exhaustive exploration of the system state-space.

B. The Kripke structure

A Kripke structure is a non-deterministic state/transition system with atomic propositions assigned to the states. Each state of the Kripke structure represents some possible configuration of the system, while a labeling function associates with each state the properties holding in it. In order to formulate a twin plant as a Kripke structure, one can simply encode states (of the two copies of the system) and the observed events of the twin plant in the state space of the Kripke structure, i.e., a state in the Kripke structure is defined as a vector (x_1, x_2, σ), where x_1, x_2 are the states of the system copies and σ is a feasible (observable) event from both x_1 and x_2.

To formulate the two notions of diagnosability as model-checking problems, we first express each diagnosability condition as an LTL formula. For simplicity, we introduce these atomic propositions: N1, NF, and NR, which mean respectively: the state q is an $N1$-state, NF-state, NR-state.

C. F-diagnosability as a Model-Checking problem:

The LTL formula which characterizes each state of an F-confused cycle in the twin plant is, $\phi_1 : G(\mathbf{N1} \wedge F\,\mathbf{NF})$

The specification can be read as follows: "*a path from the current state in the twin plant exists, where all states are $N1$-states and at least one state is an NF-state*". Therefore, property (N1 $\wedge$ F NF) is satisfied by each state in the cycle.

The model-checking problem expressing F-diagnosability is: $K_{\mathcal{P}}, S_{\mathcal{P}} \models \neg\, F(\, G(\,\mathbf{N1} \wedge F\,\mathbf{NF}\,)\,)$

where $K_{\mathcal{P}}$ is the Kripke structure corresponding to the twin plant $\mathcal{P}$ of G, and $S_{\mathcal{P}}$ is the initial state in $K_{\mathcal{P}}$.

D. R-diagnosability as a Model-Checking problem:

The LTL formula that characterizes each state of an R-confused cycle in the twin plant is, $\phi_2 : G(\mathbf{N1} \wedge F\,\mathbf{NR})$

The specification can be read as follows: "*a path from the current state in the twin plant exists, where all states are $N1$-states and at least one state is an NR-state*". Therefore, property (N1 $\wedge$ F NR) is satisfied by each state in the cycle.

The model-checking problem expressing R-diagnosability is: $K_{\mathcal{P}}, S_{\mathcal{P}} \models \neg\, F(\, G(\,\mathbf{N1} \wedge F\,\mathbf{NR}\,)\,)$

VI. Experimental Results

In order to assess the effectiveness and the scalability of the proposed approach, we perform experimentation based on a benchmark that depicts the concept of intermittent faults with assumptions A1, A2, and A3. For the verification, we use the symbolic model-checker NuSMV, which is widely used for formal verification in both academia and industry.

A. Presentation of the DES Benchmark

The DES benchmark, depicted in Figure 3, describes a manufacturing system composed of a normal part and a faulty one. Each part contains several similar production lines modeled using a Labeled Petri Net (LPN). Many parameters can be taken into account, such as the number of tokens in place P_0, the line length, or the number of production lines. In our study, we consider only the number of production lines as a variable parameter (k). Transitions $t_1, t'_1, t_2, t'_2, t_{4,i}, t'_{4,i}$ are observable $\forall 1 \leq i \leq k$. t_1 is labeled with a, t'_1 can be labeled with a or b (we consider tow tests as it will be detailed after), t_2, t'_2 are labeled with b, and $\forall 1 \leq i \leq k$, $t_{4,i}, t'_{4,i}$ are labeled with c. Transitions $t_{3.i}, t'_{3.i}, t_5, t'_5$ are unobservable $\forall 1 \leq i \leq k$. All the unobservable transitions are labeled with u excepted transition $t'_{3,1}$ and t'_5 that correspond respectively to the fault event (labeled with$f \in \Sigma_f$) and the corresponding reset event (labeled with $r \in \Sigma_r$).

As said before, two tests are performed: (Test 1) where t'_1 is labeled with a, and (Test 2) where t'_1 is labeled with b. As the benchmark is modeled by an LPN, we first generate its reachability graph with the help of TINA Tool and then, perform our technique based on the generated reachability graph. In order to assess the scalability, we increase the number of production lines k progressively for each test.

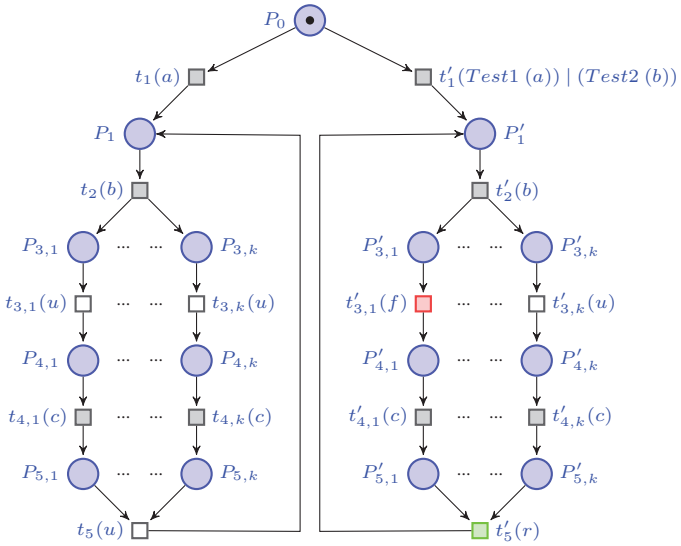

Fig. 3. The DES Benchmark

B. Results and discussion

All the experiments were conducted on a 64-bit PC, Ubuntu 14.04, an Intel Core i5, 2.5 GHz Processor and 4 GB RAM.

Table 1 summarize the obtained results, for different number of production lines. Columns from left to right correspond to the different tests, k: the number of production lines, G_S: the number of states in automaton G (i.e., the marking graph of the LPN model), G_T: the number of transitions in G, RS: the number of reachable states in the Kripke structure corresponding to the twin plant, T_{RS}: the time elapsed for generating the Kripke structure, $Diag$: the diagnosability verdict, and finally T_{Diag} : the time elapsed for verification.

TABLE I
EXPERIMENTAL RESULTS FOR LC MODELS

	k	G_S	G_T	RS	T_{RS}	$Diag$	T_{Diag}
Test 1	3	131	294	2077	0.02s	No	0.02s
	4	515	1542	38285	0.17s	No	0.28s
	5	2015	7686	663453	3.04s	No	2.70s
	6	8195	36870	11048100	57s	No	361s
Test 2	3	131	294	1040	0.01s	Yes	0.02s
	4	515	1542	19144	0.17s	Yes	0.14s
	5	2015	7686	331728	4.22s	Yes	2.12s
	6	8195	36870	5524040	63s	Yes	290s

It can be seen that for Test 1 G is non-diagnosable (non-F-diagnosable thus non-R-diagnosable). This is a logical result given the structure of the net, where the left part and the right part of the net depict the same structure (in terms of observable and unobservable transitions). However, the left part is fault-free unlike the right one, which contains an intermittent fault. Thus, two executions that share the same observations can exist in the twin plant, where in the first execution an intermittent fault can occur. However, no fault event occurs in the second one. In Test 2 (where transition t'_1 is labeled with b), the model is both F (and R)-diagnosable, since no executions that share the same observations exist, such that one execution contains intermittent fault f and the other is fault-free. The same reasoning can be considered for R-diagnosability.

Regarding the scalability of the approach, one can observe that the size of the marking graph G significantly increases with the number of production lines, which affects the size of the Kripke structure (i.e, the corresponding twin plant). This is not surprising since, on the one hand, twin plant computation is performed in a polynomial complexity regarding the size of model G. On the other hand, model-checking is very sensitive to the combinatorial explosion of the state space. Finally, three remarks relatively to the elapsed times for generating the twin plant and verifying diagnosability, can be emphasized:

1) The Model-Checker spends more time in verification than in generating the twin plant.

2) Elapsed times for generating the twin plant and verifying diagnosability stay in the order of milliseconds until 5 production lines, then it increases significantly.

3) More time elapsed for verifying diagnosability when the system is diagnosable (cf. Test 2) than when the system is not diagnosable (cf. Test 1). This can be clearly observed in line 5 (Test 1: 2.70s $\rightarrow$ 663454 states, Test 2: 2.12s $\rightarrow$ 331728 states). This result is logical, since the Model-checker needs to analyze the whole state-space to conclude that the system is diagnosable. However, when the system is not diagnosable, the verification process is stopped as soon as a counter-example to the diagnosability condition is found, that is, only a part of the generated state-space is covered.

VII. CONCLUSION & FUTURE WORK

In this paper, a formal verification approach for diagnosability analysis of intermittent faults using model-checking is proposed. Industrial system modeling, intermittent fault modeling, and two notions of diagnosability have been discussed. Then, the corresponding necessary and sufficient conditions were established. Diagnosability issues are then formulated as LTL model-checking problems based on the twin plant construction. The effectiveness and scalability of the proposed approach are experimentally evaluated through a benchmark.

This work falls within the scope of our activities on reformulating issues related to fault diagnosis of industrial systems in a model-checking framework. We have already studied the case of permanent faults and we wish, on one hand, to extend our study to deal with more complex faults such as repeated faults and other concepts of intermittent faults in both untimed and timed contexts. On the other hand, we will investigate optimization techniques for the diagnosability analysis process in a model-checking framework, in such a way as to be able to deal with larger systems.

REFERENCES

[1] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for Discrete Event Systems," *Annual Reviews in Control*, vol. 37, no. 2, pp. 308–320, 2013.

[2] M. Sampath, R. Sengupta, and S. Lafortune, "Diagnosability of discrete-event systems," vol. 40, no. 9, pp. 1555–1575, 1995.

[3] S. H. Zad, R. H. Kwong, and W. M. Wonham *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1199–1212, 2003.

[4] B. Liu, M. Ghazel, and A. Toguyéni, "Model-based diagnosis of multi-track level crossing plants," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 546–556, 2016.

[5] S. Jiang, Z. Huang, V. Chandra, and R. Kumar, "A polynomial algorithm for testing diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 6, pp. 1318–1321, 2001.

[6] T. S. Yoo and S. Lafortune, "Polynomial-time verification of diagnosability of partially observed discrete-event systems," *IEEE Transactions on automatic control*, vol. 47, no. 9, pp. 1–12, 2002.

[7] S. Jiang, R. Kumar, and H. E. Garcia, "Diagnosis of repeated / intermittent failures in discrete event systems," *IEEE Transactions on Robotic and Automatic*, vol. 19, pp. 310–323, 2003.

[8] T. S. Yoo and H. E. Garcia, "Event diagnosis of discrete-event systems with uniformly and nonuniformly bounded diagnosis delays," *American Control Conference*, vol. 6, pp. 5102–5107, 2004.

[9] C. Zhou and R. Kumar, "Computation of Diagnosable Fault-Occurrence Indices for Systems with Repeatable-Faults," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 6311–6316, 2009.

[10] O. Contant, "Failure diagnosis of discrete event system: the case of intermittent faults," *International conference on decision and control*, vol. 4, pp. 4006–4017, 2002.

[11] O. Contant, S. Lafortune, and D. Teneketzis, "Diagnosis of Intermittent Faults," *Discrete Event Dynamic Systems*, vol. 14, pp. 171–202, 2004.

[12] A. Correcher, E. Garcia, F. Morant, and E. Quiles, "Intermittent failure diagnosis in industrial processes," *IEEE International Symposium on Industrial Electronics*, vol. 2, pp. 723–728, 2003.

[13] S. Biswas, "Diagnosability of discrete event systems for temporary failures," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1534–1549, 2012.

[14] E. M. . Clarke, O. Grumberg, and D. Peled, "Model Checking, The MIT Press Cambridge, MA," 1999.

[15] A. Cimatti, C. Pecheur, and R. Cavada, "Formal verification of diagnosability via symbolic model checking," *Int. Conference on Artificial Intelligence*, pp. 363–369, 2003.

[16] A. Boussif and M. Ghazel, "Diagnosability analysis of input/output discrete event system using model checking," *The 5th International Workshop on Dependable Control of Discrete Systems (DCDS'15)*, vol. 48, no. 7, pp. 71–78, 2015.

[17] L. Carvalho, J. Basilio, and M. V. Moreira, "Robust diagnosis of discrete event systems against intermittent loss of observations," *Automatica*, vol. 48, no. 9, pp. 2068–2078, 2012.