Diagnosability of Repairable Faults

Eric Fabre, Loïc Hélouët, Engel Lefaucheux, Hervé Marchand

***Abstract*— The diagnosis problem for discrete event systems consists in deciding whether some fault event occurred or not in the system, given partial observations on the run of that system. Diagnosability checks whether a correct diagnosis can be issued in bounded time after a fault, for all faulty runs of that system. This problem appeared two decades ago and numerous facets of it have been explored, mostly for permanent faults. It is known for example that diagnosability of a system can be checked in polynomial time, while the construction of a diagnoser is exponential. The present paper examines the case of transient faults, that can appear and be repaired. Diagnosability in this setting means that the occurrence of a fault should always be detected in bounded time, but also before the fault is repaired. Checking this notion of diagnosability is proved to be PSPACE-complete. It is also shown that faults can be reliably counted provided the system is diagnosable for faults and for repairs.**

I. Introduction

In its standard version [7], the diagnosis problem for discrete event systems starts with a dynamic system A with runs of two types : some runs are safe (they contain no fault event), and the others are faulty. More generally, one may assume a regular property P on runs of A. This property P is generally absorbing, in the sense that once P is satisfied by some partial run (like the fact of being faulty) it remains true in all extensions of that run. System A is supposed to perform some hidden run u, which is partially observed by an external supervisor : only some events of the hidden run u are visible, possibly through some filtering operation, and the other events of u are silent. The problem then consists is deciding whether the hidden run u satisfies the property P of interest given the observed sequence and the model of A. Specifically, assuming at some point u satisfies P, one would like to detect it in bounded time. If this is feasible for all runs satisfying P, the system is declared diagnosable.

Property P can be seen as an abstraction on the behaviors of A, which are partially observed. Diagnosability then amounts to detecting when P holds. One may further be interested in deciding in bounded time after time t whether P holds or not at time t. A dual version of the problem relates to opacity: one would like to ensure that property P (a "secret") is never detectable by an external observer. Beyond its simple statement, the diagnosis problem thus has numerous implications in terms of security and of safety.

In the present paper, we examine the case of non persistent properties P, or non persistent faults, *i.e.* P may hold only on segments of the hidden run u. Diagnosing P thus means being able to detect that P holds in bounded time after it becomes true, and in any case *before P vanishes*.

All authors are with the SUMO team of INRIA Rennes, Campus de Beaulieu, 35042 Rennes cedex, France name.surname@inria.fr

Section II recalls standard results about the classical notion of diagnosability. The diagnosability for repairable faults, named T-diagnosability, is presented in Section III. It is proved that deciding T-diagnosability is PSPACE-complete. Section IV expresses that when both faults and repairs are T-diagnosable, one is able to count fault occurrences in the hidden run u. Finally, Section V relates these results to previous contributions on the topic.

II. Setting and known results

A. Diagnosis and diagnoser

A finite automaton over alphabet Σ is a tuple $A = (S, \Sigma, T, s_0)$, where S is a finite set of states, $s_0 \in S$ is the initial state, and $T \subseteq S \times \Sigma \times S$ is a set of transitions. Transitions take the form $t = (s, \alpha, s')$ and we denote $s^-(t) = s, \sigma(t) = \alpha, s^+(t) = s'$. *Paths* of A are finite sequences of transitions $u = t_1 ... t_n$ such that $s^+(t_i) = s^-(t_{i+1})$, and *runs* of A are paths rooted at s_0: $s^-(t_1) = s_0$. We denote $s^-(u) = s^-(t_1), s^+(u) = s^+(t_n)$, and $\sigma(u) = \sigma(t_1)...\sigma(t_n)$ the sequence of labels associated to a path u. The *language* of A is the set of label sequences produced by runs of A : $L(A) = \{\sigma(u),\ u \text{ run of } A\}$. An automaton is *deterministic* iff $\forall s, \alpha, (s, \alpha, s') \in T \wedge (s, \alpha, s'') \in T \Rightarrow s' = s''$.

Our starting point for the diagnosis problem, and without loss of generality, is a *deterministic* automaton A. Let us partition states of A into two subsets $S = S_N \uplus S_F$, and let us name S_N normal (or safe) states and S_F faulty states, to help intuition. The *faulty language* of A is derived from *faulty runs*, *i.e.* runs that terminate in a faulty state: $L_F(A) = \{\sigma(u),\ u \text{ run of } A, s^+(u) \in S_F\} \subseteq L(A)$. The *normal (safe) language* of A is defined similarly, and denoted $L_N(A)$. As A is deterministic, σ establishes a one to one correspondence between runs of A and words of its language $L(A)$, so $L_N(A) \cap L_F(A) = \emptyset$, or $L_N(A) \uplus L_F(A) = L(A)$. In this section, we assume that faults are permanent in A. Namely, there is no reachable path u in A such that $s^-(u) \in S_F$ and $s^+(u) \in S_N$. Equivalently, the faulty language of A is *saturated* in $L(A)$: $L_F(A)\Sigma^* \cap L(A) = L_F(A)$.

The diagnosis problem assumes partially observed systems, so we partition the label set Σ in two disjoint sets of observable and unobservable labels: $\Sigma = \Sigma_o \uplus \Sigma_u$. The *projection* on observable labels $\Pi : \Sigma^* \to \Sigma_o^*$ is defined as the monoid morphism generated by $\Pi(\alpha) = \alpha$ whenever $\alpha \in \Sigma_o$ and $\Pi(\alpha) = \varepsilon$ otherwise (ε = empty word). The *observable (or visible) language* of A is defined as $L_o(A) = \Pi(L(A))$. For technical reasons commented later, we define the inverse projection Π^{-1} as follows :

$$\forall w \in \Sigma_o^*,\ \Pi^{-1}(w) = \{v \in L(A) : \Pi(v) = w\} \cap \Sigma^*\Sigma_o \qquad (1)$$

i.e. we restrict the standard inverse projection to words of $L(A)$ that finish with an observable letter[1].

From a run u performed by A, or equivalently from the word $v = \sigma(u)$, one only observes the visible actions *i.e.* the word $w = \Pi(v) = \sigma_o(u) \in L_o(A)$, where $\sigma_o = \Pi \circ \sigma$. The *diagnosis* consists in deciding whether a fault has occurred in system A given this observed sequence w. A *diagnoser* for A can be seen as a function $\Delta : L_o(A) \to \{N,F,U\}$ where

$$\Delta(w) \quad = \quad \begin{cases} N \text{ iff } \Pi^{-1}(w) \subseteq L_N(A) \\ F \text{ iff } \Pi^{-1}(w) \subseteq L_F(A) \\ U \text{ otherwise} \end{cases} \qquad (2)$$

Letters N,F,U stand for "normal," "faulty," and "uncertain," (or "ambiguous") as it clearly appears above.

A diagnoser can be derived from an observer (or state estimator) of A. This observer is built in two steps. The first step is the Σ_o-*closure* of A. The Σ_o-*closure* (to the left) of A is defined as $B = Red_{\Sigma_o}(A) = (S, \Sigma_o, T', s_0)$ where $(s,\alpha,s') \in T'$ iff there exists a path $u = t_1 ... t_n$ in A such that $\sigma_o(t_1 ... t_{n-1}) = \varepsilon$, $\sigma_o(t_n) = \alpha$, $s^-(u) = s$ and $s^+(u) = s'$. Intuitively, there is a transition from s to s' in T' iff there exists a path from s to s' with a single observable action labeling the last transition of the path. The Σ_o-*closure* of A is an ε-*reduction* assuming all labels of Σ_u are first replaced by ε in A. The second step is the determinization of the resulting B, performed by standard subset construction. Let $D = Det(B) = (Q, \Sigma_o, T'', q_0)$ where $Q = 2^S$, $q_0 = \{s_0\}$ and $t = (q,\alpha,q') \in T''$ iff $q' = \{s' \in S : \exists s \in q,\ (s,\alpha,s') \in T'\}$. Of course, both B and D can be trimmed to their reachable part.

Observe that $L(D) = L(B) = L_o(A)$. D is a state estimator of A in the following sense : let $w \in L_o(A)$, as D is deterministic, there exists a unique path r in D such that $\sigma_o(r) = w$. The final state $q = s^+(r) \in Q$ of path r in D satisfies $q = s^+(\sigma_o^{-1}(w)) \in 2^S$ in A, *i.e.* it contains all states of A that are reachable by runs that produce the observed sequence *w and that stop immediately after the last observable transition*. This last condition explains the specific definition of Π^{-1} and the choice of the Σ_o-closure of A *to the left*. Let us call $q \in Q = 2^S$ a *normal* subset iff $q \subseteq S_N$, a *faulty* subset iff $q \subseteq S_F$, and an *uncertain* (or *ambiguous*) subset otherwise. D yields a diagnoser for A as follows : $\Delta(w)$ is the type of $q = s^+(\sigma_o^{-1}(w))$ in D. By extension, D is often called *the diagnoser of* A : $D = Diag(A) = Det(Red_{\Sigma_o}(A))$.

Due to determinization, D can be exponentially larger than A and should not be used for online diagnosis. One should use instead a recursive state estimation driven by the observed sequence w, which has linear complexity in the size of w and A. D can thus be considered as a precompiled version of the diagnosis for all possible observed sequences.

B. Remarks and extensions

Let A_1, A_2 be two automata, with $A_i = (S_i, \Sigma_i, T_i, s_{0,i})$, their *synchronous product* (or simply *product* for short) is the automaton $A_1 \times A_2 = (S_1 \times S_2, \Sigma_1 \cup \Sigma_2, T_1 \otimes T_2, (s_{1,0}, s_{2,0}))$ where transitions in $T_1 \otimes T_2$ are triples $((s_1,s_2),\alpha,(s_1',s_2'))$ such that

$$\begin{aligned} (s_1,\alpha,s_1') \in T_1 \ \wedge\ (s_2,\alpha,s_2') \in T_2 \quad &\text{for} \quad \alpha \in \Sigma_1 \cap \Sigma_2 \\ (s_1,\alpha,s_1') \in T_1 \ \wedge\ s_2 = s_2' \in S_2 \quad &\text{for} \quad \alpha \in \Sigma_1 \setminus \Sigma_2 \\ s_1 = s_1' \in S_1 \ \wedge\ (s_2,\alpha,s_2') \in T_2 \quad &\text{for} \quad \alpha \in \Sigma_2 \setminus \Sigma_1 \end{aligned}$$

Faulty runs in A are often not identified by a partition on states, but rather by the firing of some transition carrying a "fault" label $f \in \Sigma_u$. This can be recast in the previous setting as follows. Consider the deterministic and complete *memory automaton* $M = (\{N,F\}, \Sigma, T, N)$ where (N,f,F) is the unique transition of T producing a state change. The product $A \times M$ does not change the language of A, but performs a state augmentation that keeps track of the firing of a faulty transition in A. The label N or F now attached to states of $A \times M$ defines a partition of the state set that characterizes faulty runs. This technique was generalized in [3] to detect/diagnose runs satisfying some regular pattern of labels, rather than the simple firing of a transition labeled by f.

When faults are non permanent in A, that is when there exist transitions from S_F to S_N, one may nevertheless be interested in detecting that some transient fault has occurred. This can again be captured by an obvious transform of A into A', that adds memory to states of A to propagate the fact that a fault occurred sometime in the past. With the assumption that A is deterministic, this amounts to saturating the fault language of A: $L_F(A') = L_F(A)\Sigma^* \cap L(A)$. This idea is a variant of the pattern recognition of [3]. It was used in [5] to track the occurrence of k transient faults. It is also present in [1] under the names of O-diagnosis (detection of the occurrence of a fault) and I-diagnosis (detection of the occurrence of a repair). All these notions are thus variants of the classical diagnosis approach, even if they are recast in the context of transient failures. In [1], the authors propose a "memory automaton" that can be composed with a specification to remember occurrences of faults and repairs. However, even if fault repair is considered, their automaton propagates the information that a fault occurred. In the next section, we consider a different setting, where diagnosis is considered as accurate if it detects a fault before it is repaired.

C. Diagnosability

Let us recall the notion of diagnosability for permanent faults, *i.e.* when A has no transition from S_F to S_N. For simplicity, we assume that A is Σ_o-live : an observable transition is reachable from any state of A. Intuitively, A is diagnosable iff, whenever it reaches S_F, this is detected/diagnosed after a finite number of extra observations. Formally, A is *diagnosable* iff

$$\begin{aligned} &\forall v_1 \in L_F(A),\ \exists n \in \mathbb{N},\ \forall v_1 v_2 \in L(A), \\ &[\ |v_2|_o \geq n \ \Rightarrow\ \Pi^{-1} \circ \Pi(v_1 v_2) \subseteq L_F(A)\ ] \end{aligned} \qquad (3)$$

where $|v_2|_o$ is the length of $\Pi(v_2)$. This expression slightly differs from more frequent ones (for ex. [7]), but remains equivalent in essence. First, Definition (3) counts only visible

[1] Alternatively, we can define $L(A)$ as words that terminate with a letter of Σ_o, or equivalently by assuming faulty states in A that can only be reached by visible transitions, which does not reduce the generality of the setting.

transitions in $|v_2|_o$, instead of counting all transitions. It makes more sense to have an observable criterion to decide when to collect the diagnosis. And when A has no unobservable cycle, which is generally assumed when one uses $|v|$ instead of $|v|_o$, this rephrasing is harmless. Secondly, one generally assumes a uniform value of n covering all faulty words v_1. Again, taking account the finiteness of A, this uniform bound comes for free once (3) holds.

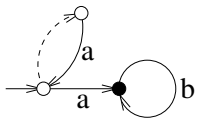

Fig. 1. Normal/faulty states are represented as white/black dots. The dashed line represents an unobservable transition. This automaton is diagnosable: after it reaches the faulty state, it can only produce a b which characterizes the occurrence of the fault. Nevertheless, driven by sequence a^n, the diagnoser outputs U^n. So uncertainty can be arbitrarily long.

Def. (3) states that a system is diagnosable iff, when uncertainty appears *after a faulty run*, it does not hold forever. Observe that the diagnosis may nevertheless remain uncertain for an arbitrarily long time, even for a diagnosable system, as long as no fault occurs (see Fig. 1). Conversely, A is *not diagnosable* iff after some faulty run uncertainty can never be resolved:

$$\exists v_1 \in L_F(A),\ \forall n \in \mathbb{N},\ \exists v_1v_2 \in L(A),$$
$$[\ |v_2|_o \geq n\ \wedge\ \Pi^{-1} \circ \Pi(v_1v_2) \cap L_N(A) \neq \emptyset\ ] \qquad (4)$$

The last term in (4) can be rephrased as $\Delta(\Pi(v_1v_2)) = U$ or equivalently $\exists v' \in L_N(A),\ \ \Pi(v_1v_2) = \Pi(v')$. This new formulation expresses that one can find an arbitrary long extension v_2 of some faulty word v_1 which is *observationally equivalent* (or *equivalent* for short) to a safe word v' of A, denoted by $v_1v_2 \sim_o v'$. As faults are permanent, any prefix of the safe word v' is also safe. Def. (4) thus opens the way to a polynomial test for (non-)diagnosability: one can build a twin-machine that recognizes pairs of runs made of a faulty one v_1v_2 and an equivalent (w.r.t observation) safe one v', and thus check how long uncertainty can last.

Consider $B = Red_{\Sigma_o}(A)$, the *twin machine* of A is obtained as $C = B \times B$. A run in C is a pair of runs of B that are observationally equivalent, from which one can recover a pair (v, v') of observationally equivalent words of $L(A)$. C has $S \times S$ as state set, so states of C can be called normal/safe, faulty or uncertain/ambiguous, as in D, the diagnoser of A. An *ambiguous cycle* in C is a reachable cycle that only goes through ambiguous (pairs of) states.

Proposition 1: A is diagnosable iff its twin machine C has no ambiguous cycle.

This result was proved in [4]. The only if part is obvious as the presence of an ambiguous cycle allows one to build an arbitrarily long suffix v_2 to a faulty word v_1 by repeating the cycle, while having this faulty word v_1v_2 equivalent to a safe one v'. This proves non-diagnosability. The if part uses the finiteness of A, as a long enough suffix v_2 necessarily contains a (faulty) cycle of B that can be matched to an observationally equivalent safe/normal cycle of B.

The original version of Proposition 1 actually relied on a twin machine directly built from A and not from $B = Red_{\Sigma_o}(A)$. Proposition 1 clearly yields a polynomial (quadratic) test for the diagnosability of A.

III. Diagnosability of Repairable Faults

A. Diagnosis and T-diagnosability

We still consider a Σ_o-live deterministic automaton A, and now assume that some faults in A can be repaired, i.e. A contains transitions from S_F to S_N, or equivalently that the fault language $L_F(A)$ is not saturated. The diagnosis of an observed sequence $w = \sigma_o(u)$ produced by some run u of A is defined as in (2). However, we reinforce the diagnosability criterion for A by requiring that, when some fault occurs, it is still detected in finite time, but also *before it is repaired*.

Let us first introduce some notation. We denote by $L_F^{min} = \{v\alpha \in L_F(A) \|\ v \notin L_F(A) \wedge \alpha \in \Sigma\}$ the set of minimal faulty words of A, *i.e.* words that correspond to a run ending with a transition from a normal state to a faulty one in A. For a word $v_1 \in L_F(A)$, let $v_1 \rightarrow v_1v_2 \in L_F(A)$ denote the continuous presence of a fault along v_2. Formally, $v_1 \rightarrow v_1v_2 \in L_F(A)$ iff $\forall v_2' \leq v_2,\ v_1v_2' \in L_F(A)$, where $\leq$ denotes the prefix relation on words.

Formally, an automaton A is *timely diagnosable* (*T-diagnosable* for short) iff

$$\forall v_1 \in L_F^{min}(A),\ \exists n \in \mathbb{N},\ \forall v_1v_2 \in L(A),$$
$$[\ |v_2|_o \geq n\ \ \Rightarrow\ \ \exists v_2' \leq v_2 :\ v_1 \rightarrow v_1v_2' \in L_F(A)$$
$$\wedge\ \Pi^{-1} \circ \Pi(v_1v_2') \subseteq L_F(A)\ ] \qquad (5)$$

T-diagnosability differs from Def. (3) mainly by requiring that the fault that appears in v_1 remains for the whole execution of prefix v_2'. This notion is illustrated in Fig. 2, that depicts several observationally equivalent runs, and shows observation times at which a correct diagnosis/detection can be produced (before repair). Observe that if faults are not repairable, $v_1 \in L_F^{min}(A)$ implies that $v_1 \rightarrow v_1v_2' \in L_F(A)$ for every v_2', and Def. (5) reduces to Def. (3) (condition $\forall v_1 \in L_F(A)$ in Def. (3) can equivalently be replaced by $\forall v_1 \in L_F^{min}(A)$). So, in a setting of permanent faults, T-diagnosability is equivalent to diagnosability.

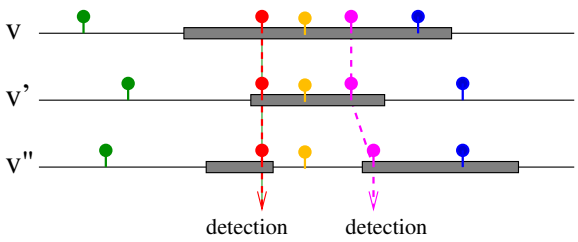

Fig. 2. A faulty word v and two equivalent words v', v''. The observed labels are represented as pins, and the faulty zones as grey rectangles. Detections correspond to times (in number of observations) where all equivalent words are faulty.

Fig. 3 illustrates the notion of T-diagnosability. Safe (resp. faulty) states are represented as white (resp. black) patches. One has $\Sigma = \{a, b, c, d\}$ and $\Sigma_o = \{a\}$. Ignoring the grayed

transitions at the bottom, the automaton is T-diagnosable as after the observation of sequence a a fault occurred in both runs at the top, and this fault is each time detected before it is repaired since $\Delta(a) = F$. By adding the bottom part, T-diagnosability is lost: once a has been observed, one knows for sure that a fault occurred, but no detection can take place before repair, in all runs, as now $\Delta(a) = U$.

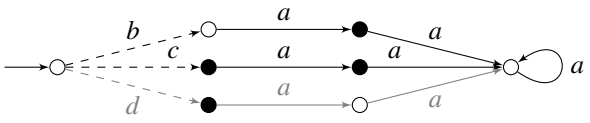

Fig. 3. A T-diagnosable system, when the path at the bottom is ignored.

B. Vanishing faults and repairs

T-diagnosability seems to be a reasonable first step towards the ability to count fault occurrences. Unfortunately, this is not the case as it is already apparent in Fig. 2: an automaton with such equivalent runs can be T-diagnosable, and nevertheless the same observed sequence matches a run with one fault (top) and one with two faults (bottom). The situation is even worse. Let us call a *vanishing fault* a fault that occurs and is repaired in the silent part of a run of A (*i.e.* between two observations), and similarly for a *vanishing repair*. Then automaton A can exhibit runs with an arbitrary number of vanishing faults and repairs without losing its T-diagnosability.

This is illustrated by the example in Fig. 4: $\Sigma = \{a,b\}$, $\Sigma_o = \{a\}$. In this automaton A, one has $\Delta(a) = F$. A vanishing repair appears at the end of word ab and a vanishing fault at the end of abb. Nevertheless, T-diagnosability holds: for $v_1 = a \in L_F^{min}(A)$ one gets immediate fault detection ($v_2 = \varepsilon$ works), for $v_1 = ab^2 \in L_F^{min}(A)$ one has $\Pi^{-1}(\Pi(v_1)) = \{a\} \subseteq L_F(A)$ so again the fault detection is "immediate" with $v_2 = \varepsilon$, and similarly for $v_1 = ab^4 \in L_F^{min}(A)$.

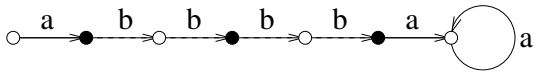

Fig. 4. An arbitrary number of vanishing faults and repairs may exist in a T-diagnosable automaton.

It is quite counter-intuitive that the "immediate" detection of the fault occurring at $v_1 = ab^2$ actually relies on the detection of the fault that took place previously, at $v_1 = a$. This phenomenon is due to the fact that T-diagnosability, just as diagnosability, only refers to runs that stop at a visible transition. Everything that happens between observations is almost ignored. For permanent faults, this is harmless: it only shifts the detection by one observation. For repairable faults, it introduces odd phenomena. A natural way to make fault detection causal (and to open the way to a counting of faults) is thus to forbid the existence of vanishing repairs

$$\begin{aligned}\nexists v = v_1 v_2 \alpha \in L(A) :\ & v_1 \in L_F(A) \wedge v_1 v_2 \in L_N(A) \\ & \wedge\ \alpha \in \Sigma \wedge v_1 v_2 \alpha \in L_F(A) \wedge \Pi(v_2) = \varepsilon \end{aligned} \tag{6}$$

and of vanishing faults

$$\begin{aligned}\nexists v = v_1 v_2 \alpha \in L(A) :\ & v_1 \in L_N(A) \wedge v_1 v_2 \in L_F(A) \\ & \wedge\ \alpha \in \Sigma \wedge v_1 v_2 \alpha \in L_N(A) \wedge \Pi(v_2) = \varepsilon \end{aligned} \tag{7}$$

Under these assumptions, at most one transition from S_F to S_N or from S_N to S_F can take place between two visible events. Let us say that fault detection is causal when observations following the fault enable its detection, so the detection does not depend on observations that occurred strictly before the fault as in the pathological cases above. Such causality can then be expressed as follows.

Proposition 2: Assuming (6) and (7), A is T-diagnosable if and only if

$$\begin{aligned}\forall v_1 \in L_{F,o}^{min}(A) \cup L_{F,u}^{min}(A)\Sigma_u^*\Sigma_o,\ \exists n \in \mathbb{N},\ \forall v_1 v_2 \in L(A),& \\ [\ |v_2|_o \geq n\ \Rightarrow\ \exists v_2' \leq v_2 :\ v_1 \to v_1 v_2' \in L_F(A)& \\ \wedge\ v_1 v_2' \in \Sigma^*\Sigma_o \wedge\ \Pi^{-1} \circ \Pi(v_1 v_2') \subseteq L_F(A)\ ]& \end{aligned} \tag{8}$$

where $L_{F,o}^{min}(A) = L_F^{min}(A) \cap \Sigma^*\Sigma_o$ represent minimal faulty runs that terminate with a visible event, and $L_{F,u}^{min}(A) = L_F^{min}(A) \setminus L_{F,o}^{min}(A)$ represent those that terminate with a silent event.

Proof: The extra condition $v_1 v_2' \in \Sigma^*\Sigma_o$ requires that the fault detection takes place at the moment one gets an observation. This could have been introduced in (5) without loss of generality, as silent events at the end of v_2' are useless to the criterion $\Pi^{-1} \circ \Pi(v_1 v_2') \subseteq L_F(A)$. So the only novelty lies in the first term. Recall that $L_F^{min}(A) = L_{F,o}^{min}(A) \uplus L_{F,u}^{min}(A)$. Words $v_1 \in L_{F,o}^{min}(A)$ are considered by both (5) and (8). But words $v_1 \in L_{F,u}^{min}(A)$ in (5) are replaced by words $v_1 \in L_{F,u}^{min}(A)\Sigma_u^*\Sigma_o$ in (8). In other words, for faults that occur silently, detection takes places after the next visible event.

Only if part. Assume A is T-diagnosable, and let $v_1 \in L_{F,u}^{min}(A)\Sigma_u^*\Sigma_o$. v_1 decomposes uniquely as $v_1 = v_0 u_3$ where v_0 is the $L_{F,u}^{min}$ part and u_3 the extension in $\Sigma_u^*\Sigma_o$. v_0 further decomposes as $v_0 = u_1 u_2$ where u_2 is the longest silent suffix of v_0. Thanks to (6) and (7), one has that $u_1 \in L_N(A)$, and $u_1 u_2 \to u_1 u_2 u_3 \in L_F(A)$. As A is diagnosable and $v_0 \in L_F^{min}(A)$, let us take any long enough extension $v_2 \geq u_3$ for the fault detection in Def. (5), and let $v_2' \leq v_2$, $v_2' \in \Sigma^*\Sigma_o$ be the detection time. One can not have $v_2' < u_3$ because in that case $\Pi(v_0 v_2') = \Pi(v_0) = \Pi(u_1)$ and $u_1 \in L_N(A)$. So the detection of the fault can not occur before the extra observation lying at the end of u_3. Since $v_2' \geq u_3$, one has $v_2' = u_3 v_2''$ and $v_0 \to v_0 u_3 v_2'' \in L_F(A)$. This proves the existence of a detection time v_2'' after $v_1 = v_0 u_3$ which satisfies (8).

If part. Assume A satisfies (8) and let $v_1 \in L_{F,u}^{min}(A)$. v_1 decomposes uniquely as $v_1 = u_1 u_2$ where u_2 is the longest silent suffix of v_1. Thanks to (6) and (7), one has $v_1 \in L_N(A)$. Let $v_1 v_2 \in L(A)$, with v_2 long enough, in particular $|v_2|_o \geq 1$. One can write $v_2 = u_3 u_4$ with $u_3 \in \Sigma_u^*\Sigma_o$. Thanks to (6) and (7) again, one has $v_1 \to v_1 u_3 \in L_F(A)$. As $v_1 u_3 \in L_{F,u}^{min}(A)\Sigma_u^*\Sigma_o$ and u_4 is long enough, there exists a prefix $u_4' \leq u_4$ such that $v_1 u_3 \to v_1 u_3 u_4' \in L_F(A)$ and $\Delta(\Pi(v_1 u_3 u_4')) = F$. Taking $v_2' = u_3 u_4'$ thus satisfies the conditions of (5). ■

C. A T-diagnosability test

As in Section II-C, one can consider the converse of (5). Specifically, A is *not T-diagnosable* iff

$$\exists v_1 \in L_F^{min}(A),\ \forall n \in \mathbb{N},\ \exists v_1 v_2 \in L(A) : |v_2|_o \geq n, \quad (9)$$
$$\forall v_2' \leq v_2,\ v_1 \rightarrow v_1 v_2' \notin L_F(A) \ \vee\ \Pi^{-1} \circ \Pi(v_1 v_2') \not\subseteq L_F(A)$$

In words, A is not T-diagnosable whenever it is possible to find a minimal faulty sequence v_1 and arbitrarily long extensions v_2 such that along the longest faulty prefix $v_2' \leq v_2$ of v_2, the detection of the fault can not occur.

It is worth noticing that the twin-machine idea used to check the diagnosability of permanent faults is not sufficient to check the T-diagnosability of repairable faults. The main obstacle is that T-diagnosability can not be characterized by pairs of equivalent runs. It is rather a global property on classes of equivalent runs in A. This is illustrated in Fig. 5, where unobservable transitions are depicted as dashed arrows ($\Sigma_o = \{a\}$). This automaton is not T-diagnosable. However, by checking only *pairs* of equivalent runs, one always finds a time where ambiguity seems to vanish. For example, considering only the top and central loops, a^{3n+1} seem to be detection times for the faults that appear in these runs. To reveal that T-diagnosability does not hold, one would have to check triples of equivalent runs here. And it is easy to design examples where triples are not sufficient and one needs to escalate to quadruples of equivalent runs to reveal the non T-diagnosability, etc. This suggests a non polynomial complexity of the T-diagnosability test.

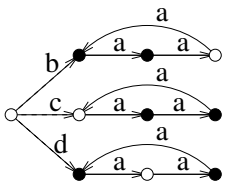

Fig. 5. This system is not T-diagnosable, but this is not apparent if only pairs of equivalent runs are considered.

The idea of the twin-machine construction is to check whether a faulty run can create an ambiguity that can never be resolved. For repairable faults, this ambiguity signal can be directly derived from $Diag(A)$, the diagnoser of A. Consider the (deterministic) automaton $G = A \times Diag(A)$. $Diag(A)$ is a deterministic automaton over alphabet $\Sigma_o \subseteq \Sigma$, and $L(Diag(A)) = L_o(A)$. So $L(G) = L(A)$: the construction of G performs a simple state augmentation on A, without changing its behavior (just like the memory automaton mentioned above). This state augmentation attaches an ambiguity status to each state of A as follows. States of G take the form $(s,q) \in S \times Q$ where $Q = 2^S$. So they can be labeled by elements in $\{N,F\} \times \{N,U,F\}$: for example (s,q) is of type (N,U) iff $s \in S_N$ and q is uncertain. $L_N(A)$ and $L_F(A)$ are easily identifiable in G as words terminating in a state of type $(N,.)$ or $(F,.)$ respectively. A state (s,q) is said to be *minimally faulty* iff s is the terminal state of a run $v_1 \in L_F^{min}(A)$.

Theorem 1: With notation above, A is not T-diagnosable if and only if there exists a reachable minimally faulty state $(s,q) \in S \times Q$ in G such that (s,q) is of type (F,N) or (F,U) and either

1) there exists a state (s',q') of type (N,N) or (N,U)
2) or there exists a cycle of (F,U) states

that is reachable from (s,q) through a (possibly empty) sequence of (F,N) states followed by a sequence of (F,U) states.

Proof: By construction of G, observe that if word $v \in L(A)$ reaches state s in A, then word v reaches state (s,q) in G and $\Delta(\Pi(v))$ is the type of state $q \in Q$, either N,F or U.

For the only if part, consider the witness $v_1 \in L_F^{min}(A)$ of non T-diagnosability in (9), which reaches state (s,q) in G. (s,q) is necessarily of type (F,N) or of type (F,U), as if (s,q) is of type (F,F) then the correct diagnosis is output with $v_2' = \varepsilon$. For a given n, let v_2 be the extension of v_1 satisfying (9), and let v_2' be the longest prefix of v_2 such that $v_1 \rightarrow v_1 v_2' \in L_F(A)$. All along v_2', the correct diagnosis can not be output, so G only crosses states of type (F,N) or (F,U). States of type (F,N) come first (if they exist), then (after the first observable event in v_2') one only crosses states of type (F,U) as at least one faulty run lies in the inverse projection. If there exists $a \in \Sigma$ such that $v_2' a \leq v_2$, then $v_1 v_2' a$ reaches state (s',q') which is either of type (N,N) or of type (N,U). (N,F) is not possible as this would mean that the correct diagnosis was produced for $v_1 v_2'$. This makes point 1 in the theorem. If point 1 never occurs for any n, this means that in the discussion above one always has $v_2' = v_2$. As G is finite, it then contains a cycle with at least one observable event (recall that n counts observations). This cycle is thus made of (F,U) states, which makes point 2 in the theorem.

The if part can be derived in a similar manner, starting from conditions in the theorem and building a witness v_1 and the associated v_2 for every n satisfying (9). ■

D. Complexity of T-diagnosability

Theorem 2: Deciding whether an automaton A is T-diagnosable is a PSPACE-complete problem.

Proof: First, we can easily show that T-diagnosability belongs to PSPACE. Following the result of Theorem 1, A is not T-diagnosable iff one can find a witness cycle of type (F,U) or a witness state of type (N,N) or (N,U) reachable after a minimally faulty sequence ending in a state of type (F,N) or (F,U) in G. First of all, the size of G is at most $2^{|A|}.|A|$. To witness a minimally faulty sequence ending in a (F,N) of (F,U) state, one only needs to non-deterministically explore paths of size smaller than $2^{|A|}.|A|$, which can be done with polynomial memory size (to remember current state and whether previous state is faulty). Then, to witness ambiguous cycles or moves to (N,N) or (N,U) states, one can again non-deterministically explore paths of G of size smaller than $2^{|A|}.|A|$ with polynomial memory. Hence, finding witness paths for non-T-diagnosability is a NPSPACE process, and using Savitch's theorem, and remembering that PSPACE is closed under complementation, this shows that T-diagnosability is in PSPACE. The second

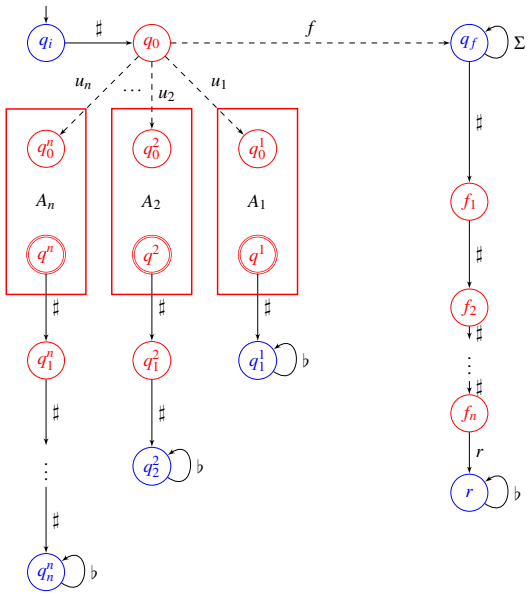

Fig. 6. PSPACE-hardness of T-diagnosability. Red states are faulty, and blue states normal states.

step of the proof shows hardness of the problem by reduction from a language inclusion problem, which is known to be PSPACE-complete [6]. The problem can be formulated as follows: given $A_1,...,A_n$ some deterministic finite automata, $\bigcap_{i\in 1..n} L(A_i) = \emptyset$?

Let $n \in \mathbb{N}$ and for $1 \leq i \leq n$, $A_i = (S_i, \Gamma, T_i, q_0^i, F_i)$ be some deterministic finite automaton on alphabet Γ. We build the finite automaton $A = (S, \Sigma, T, q_0)$ (see Figure 6) where:

- $\Sigma = \Gamma \cup \{u_1, \ldots u_n\} \cup \{f, \sharp, \flat, r\}$
- $S = \{q_i, q_0, q_f, r\} \cup \{f_i \mid i \in 1..n\} \cup \{q_i^j \mid i, j \in 1..n \wedge j \geq i\} \cup \bigcup_{1\leq i\leq n} S_i$
- $T = \{(q_0, f, q_f), (q_i, \sharp, q_0)(f_n, r, r), (r, \flat, r), (q_f, \sharp, f_1)\} \cup \{(f_i, \sharp, f_{i+1}) \mid i \in 1..n-1\} \cup \{(q_f, a, q_f) \mid a \in \Sigma\} \cup \{(q_0, u_i, q_0^i) \mid i \in 1..n\} \cup \{(q_i^i, \flat, q_i^i \mid i \in 1..n\} \cup \{(q, \sharp, q_1^i) \mid q \in F_i, i \in 1..n\} \cup \{(q_i^j, \sharp, q_{i+1}^j \mid 1 \leq i < j \leq n\} \cup \bigcup_{1\leq i\leq n} T_i$

The set of safe states is $S_N = \{q_i, q_f, r\} \cup \{q_i^i \mid i = 1 \ldots n\}$. The set of faulty states is $S \setminus S_N$. We set $\Sigma_o = \Gamma \cup \{\sharp, \flat, r\}$.

We claim that A is T-diagnosable if and only if $\bigcap_{i\in 1..n} L(A_i) = \emptyset$.

First, remark that after observing $\sharp w \sharp^m$ for $m \leq n$, the current run is either in state f_m or in q_m^j for $j \geq m$ such that A_j accepts w.

Suppose that A is T-diagnosable. Let $w \in \Sigma^*$, v_1 be the unique run of A such that $\sigma(v_1) = \sharp f w \sharp$. As v_1 is a minimal faulty run, there exists $m \leq n$ such that the run $v_1 v_2$ with $\sigma(v_1 v_2) = \sharp f w \sharp^m$ verifies $\Pi^{-1} \circ \Pi(v_1 v_2) \subseteq L_F(A)$. From our earlier remark, as q_m^m is safe, it means that A_m does not accept w. As this is true for every $w \in \Sigma^*$, $\bigcap_{i\in 1..n} L(A_i) = \emptyset$.

Conversely suppose that $\bigcap_{i\in 1..n} L(A_i) = \emptyset$. Let v_1 be a minimal faulty run. Only two cases can appear: either v_1 is the word $v_1 = \sharp$ which ends in the faulty state q_0, or v_1 is of the form $v_1 = \sharp f w \sharp$. If $v_1 = \sharp$, then we know that A is in q_0 which is faulty and we can claim the fault. In the second case $s^+(v_1) = f_1$ and $\sigma(v_1) = \sharp f w \sharp$ with $w \in \Sigma^*$. As $\bigcap_{i\in 1..n}(A_i) = \emptyset$, there exists $i \in 1..n$ such that $w \notin L(A_i)$. Consider the run $v_1 v_2$ with $\sigma(v_1 v_2) = \sharp f w \sharp^i$, this run ends in f_i and was not repaired in between. Moreover, thanks to the earlier remark, for every $j > i$ such that A_j accepts w, runs with the same observation $w\sharp^i$ ends in state q_i^j for $j \geq i$. Thus, even if w is recognized by A_k for some $k < i$, no run visiting a state of A_k and with observation $w\sharp^i$ exists. As A_i does not accept w and as the states q_i^j for $j > i$ are faulty, $\Pi^{-1} \circ \Pi(v_1 v_2) \in L_F(A)$. Thus the fault can be claimed. As this is true for every minimal faulty run v_1, A is T-diagnosable. ■

IV. Counting faults

As faults are not permanent, counting the number of faults occurring at runtime is a useful information: even if a system is able to repair all occurrences of faults, a too large number of faults may indicate a major failure. To count faults, an immediate idea is to maintain a fault counter that is incremented each time the diagnoser goes from N to F and from U to F. Even if a diagnosis can be triggered in time, i.e. before the fault is repaired, T-diagnosability is not sufficient to correctly count faults along a trajectory. Fig. 2 reveals that this can not work as counting moves of the diagnoser from $\{N,U\}$ to F in this example would detect two faults, while v has only one fault and v'' has two. Conversely, counting only moves from N to F or from U to F leads to minoring the real number of faults that occurred in some runs. This section considers extra conditions that enable counting. A *fault counter* C of an automaton A is a function from Σ_o^* to $\mathbb{N}$ such that: for every run $v \in L(A)$, letting k_v be the number of fauls in v, $C(\Pi(v)) \in \{k_v - 1, k_v\}$. An automaton A is *fault countable* if there exists a fault counter of A.

Proposition 3: Assuming that there is no vanishing repairs/faults in A, deciding if A is fault countable w.r.t. F is in NLOGSPACE.

Remark that this does not immediately give the construction of a fault counter for the automaton. We will say that an automaton is T-Diagnosable w.r.t. N if repairs can be faithfully detected. Intuitively, this property can be checked by inversion of safe and faulty states, an then checking T-diagnosability of the so-obtained system. Consider the Diagnosis function $\Delta : L_o(A) \to \{N,F,U\}$ defined by (2).

We define the function $\sharp_\Delta^F$ from $L_o(A)$ to $\mathbb{N}$ as follows: Let $\mu \in L_o(A)$ and $\rho \in (N+U+F)^*$ the associated sequence of verdict emitted by Δ. Let $\rho' \in (N+F)^*$ be the projection of ρ on the verdicts $\{N,F\}$, then $\sharp_\Delta^F(\mu)$ is the number of occurrences of pairs NF that appear in ρ'. Intuitively, $\sharp_\Delta^F$ is a function that will be used to count the number of faults the diagnoser is able to detect. We can define similarly function $\sharp_\Delta^N$, counting the number of detected repairs, by inverting N and F in the previous definition.

Given a run u of A, $\sharp_A^F(u)$ denotes the number of times A moves from a normal state to a faulty state in u and $\sharp_A^N(\sigma(u))$ denotes the number of times A evolves from a faulty state to a normal state in u. We can now state the following proposition:

Proposition 4: If A is T-Diagnosable w.r.t. F and T-Diagnosable w.r.t. N, and has no vanishing faults nor repairs, then $\forall v \in L(A)$ and $\mu = \Pi(v)$, then

- $0 \leq \sharp_A^F(v) - \sharp_\Delta^F(\mu) \leq 1$.
- $0 \leq \sharp_A^N(v) - \sharp_\Delta^N(\mu) \leq 1$.

Moreover if $\Delta(\mu) = F$ then $\sharp_A^F(v) = \sharp_\Delta(\mu)$ and if $\Delta(\mu) = N$ then $\sharp_A^N(v) = \sharp_\Delta^N(\mu)$.

Intuitively, this proposition states that we can build from the diagnoser a function that counts the number of times the system becomes faulty (resp. is repaired) with a difference of at most 1. Furthermore, the difference is null as soon as the fault (resp repair) is diagnosed by the diagnoser.

The proofs of Proposition 3 and 4 can be found in [2]

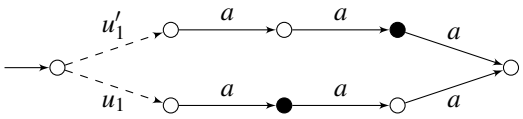

Fig. 7. Here the automaton is T-diagnosable w.r.t. N but not w.r.t. F, moreover the sequence of verdicts emitted by Δ is $NUUN$. However, after reading aa we know a single fault happened for sure.

V. Related work

The diagnosis of such transient faults has been considered in [1], which proposed four notions of diagnosability. One of them ("O-diagnosability") consists in detecting the occurrence of a transient fault, even after it has been repaired, which amounts to saturating $L_F(A)$ (see Section II-B). Symmetrically, the "I-diagnosability" aims at detecting the occurrence of a repair, even if fault(s) followed, which amounts to inverting the roles of S_F and S_N, or to saturating the safe language $L_N(A)$. Both notions thus match the standard (or historical) notion of diagnosability for a slightly modified version of A. In the same manner, the notions of "P-diagnosability" and "R-diagnosability" are dual of one another, so our work should only be compared to the notion of "P-diagnosability" proposed by [1]. "P-diagnosability" states that after the occurrence of a fault, it is always possible to detect the fact that the system is currently faulty, based on the observation (even though the fault has been repaired in the past). Our notion of T-Diagnosability is then stronger than P-diagnosability, as we require that detection fault occur before they are repaired. It is then easy to show that whenever a system is T-diagnosable then it is also P-diagnosable. Compared to [5], the notion of $[1 \cdots K]$-diagnosability of simply the K diagnosability, we introduce a sufficient condition under which it is possible to exactly count the number of faults that occurred in the system. Furthermore, similarly to [1], in the definitions of diagnosability introduced in [5], the authors do not request the detection of the fault before its repair.

VI. Conclusion

We have proposed a notion of "timely-diagnosability" that requires the detection (in bounded time) of transient faults after they occur, and before they are repaired. This notion was defined for a deterministic partially observed automaton. While this choice allows one to express most properties in terms of faulty and safe languages, it leads to quite complicated criteria for T-diagnosability, as in Theorem 1. It could be interesting to define T-diagnosability for non-deterministic automata, and to explore whether criteria simplify. For example, it is likely that in the absence of vanishing faults and of vanishing repairs, T-diagnosability is preserved by Σ_o-closure. Also, while the T-diagnosability of faults relies on a complicated criterion, it is likely that systems which are both T-diagnosable for faults and for repairs are much easily characterized. This subclass is quite interesting, as it corresponds to systems where all changes of state class are detected in bounded time, and in any case before they change again. So ambiguity, when it appears, can not last forever.

T-diagnosability is stronger than the P-diagnosability of [1] in the sense that the latter does not require that a transient fault be detected before it is repaired. Nevertheless, it is likely that P-diagnosability remains PSPACE complete, but this still has to be proved.

Besides these immediate perspectives, the future of this work is definitely in the direction of quantitative analysis. Being able to characterize exactly, after a bounded delay, in which state class lies system A is a very strong property. A more relevant question would be to determine how likely it is that A is in S_N or S_F given partial observations, and whether this relative certainty passes some threshold in a bounded time after system A has changed class.

Acknowledgements

The authors would like to thank Francois Godi, Xavier Montillet and Chen Qian, master students at ENS Rennes, for interesting discussions that led to this work.

References

[1] Olivier Contant, Stéphane Lafortune, and Demosthenis Teneketzis. Diagnosis of intermittent faults. *Discrete Event Dynamic Systems*, 14(2):171–202, 2004.

[2] E. Fabre, L. Hélouë, H. Marchand, and E. Lefaucheux. Diagnosability of repairable faults (long version). In *Workshop on Discrete Event Systems, WODES'16*, 2016. https://hal.inria.fr/hal-01302562.

[3] T. Jéron, H. Marchand, S. Pinchinat, and M-O. Cordier. Supervision patterns in discrete event systems diagnosis. In *Workshop on Discrete Event Systems, WODES'06*, pages 262–268, 2006.

[4] S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial time algorithm for diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 46(8):1318–1321, 2001.

[5] S. Jiang, R. Kumar, and H.E. Garcia. Diagnosis of repeated/intermittent failures in discrete event systems. *IEEE Transactions on Robotics and Automation*, 19(2):310–323, April 2003.

[6] Dexter Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 254–266. IEEE Computer Society, 1977.

[7] Meera Sampath, Raja Sengupta, Stéphane Lafortune, Kasim Sinnamohideen, and Demosthenis Teneketzis. Failure diagnosis using discrete-event models. *IEEE Trans. Contr. Sys. Techn.*, 4(2):105–124, 1996.