Monitoring and Active Diagnosis for Discrete-Event Systems

Elodie Chanthery [*,**] **Yannick Pencolé** [*]

[*] *LAAS-CNRS, University of Toulouse, Toulouse, France*
(e-mail: [elodie.chanthery, yannick.pencole]@laas.fr)
[**] *University of Toulouse, INSA, Toulouse, France*

Abstract: This article presents an original way to enrich the monitoring of discrete-event systems named *active diagnosis*. The objective of on-line active diagnosis is to find an admissible sequence of actions (or plan) that refines the diagnosis without radically changing the mission plan. This paper has two major contributions. First, active diagnosis is formally defined in the finite-state automata theory framework. This leads to the definition of a complete active diagnoser which on line monitors the system behavior. Secondly, from the complete active diagnoser is defined a planning problem. The goal is to find a conditional plan that defines an admissible sequence of actions. These actions are applied on the physical system and may conduct the active diagnoser into a diagnosable region.

Keywords: Fault diagnosis, discrete-event systems, finite state machines, decision trees, planning

1. INTRODUCTION

Autonomy is the ability to independently perform decisions and act in a changing environment. One of the most important characteristic of an autonomous system is its ability to take care of itself when performing its mission. This ability, also called self-maintenance, requires on-line diagnosis (fault detection and isolation) and on-line planning/replanning (Chanthery et al. (2005)) launched by processes integrated in the on-board architecture of the system. On-line diagnosis is usually considered as a task that reacts to a flow of observations provided by sensors and returns estimations of the system's state. However, this process is often too limited in practice due to a lack of observations and does not take into account the fact that autonomous systems can act on themselves. One way to improve the performance of the diagnostic process is to use the action capabilities of the system in order to refine the diagnosis in case of ambiguity: this is the *active diagnosis* problem. There has been considerable amount of work about active diagnosis problem for static systems. Refining the diagnosis can be seen as a test sequencing problem and can be solved as a post-mortem diagnosis problem (Pattipati and Dontamsetty, 1992). This is usually an off-line process. In the case of dynamical systems that are currently performing a mission, the challenge of an active diagnosis process is to propose an admissible sequence of actions (or *plan*) that refine the diagnosis without radically changing the mission plan.

The contribution of this paper is a formal definition of the active diagnosis problem on discrete-event systems (DES for short). Relying on an on-board architecture, we propose to extend classical fault diagnosis techniques on DES to handle actions and the ability of the system to plan actions. The result of this formalisation is the *active diagnoser* that synthesises the necessary pieces of information to perform active diagnosis on the underlying DES.

This paper is organised as follows. Section 2 presents related work on active diagnosis on dynamical systems. Section 3 presents a formal background about fault diagnosis in a DES. Section 4 proposes an on-board architecture for active diagnosis and a formal characterisation of the active diagnoser. Section 5 explains how the planning problem for active diagnosis can be derived from the active diagnoser. It proposes a planning algorithm and discusses the difficulties involved by the integration of active diagnosis into an on-board architecture including the mission planning process. Section 6 presents an example of active diagnosis and some experimental results.

2. RELATED WORK

To our best knowledge, there are very few works about active diagnosis in dynamical systems. The main contribution on active diagnosis of discrete-event systems is the work of Sampath et al. (1998). Active diagnosis is formulated as a supervisory control problem (Ramadge and Wonham (1989)) where the legal language is an "appropriate" regular sublanguage of the regular language of the system. The proposed solution is to design a system controller in such a way that it satisfies specified control objectives and results in a diagnosable controlled system. In other words, the action domain is restricted so that the system always remains diagnosable. This approach seems to be too restrictive for autonomous systems that realize a mission. Indeed, they need to keep all their action capability, even if they intermittently loose their diagnosis capability. In this article, we reuse the idea that consists in combining monitoring, diagnosis capability and controller, without restricting the controller's action capabilities.

In the field of hybrid systems, Bayoudh et al. (2008) propose some ideas to achieve active diagnosis in a hybrid system framework. Starting in a non diagnosable region, the authors use the diagnosability analysis method to determine the sequence of controllable actions to be applied to the system in order to bring it into a diagnosable one. The authors propose to solve this problem as a conditional planning problem using an AND-OR graph, but the solving algorithm is not implemented. In particular, the type of tree exploration and the criterion for the exploration are not given. We propose to give the formal definition of active diagnoser and to discuss the planning problem more precisely.

The problem of diagnosis refinement by action planning has also been raised by McIlraith (1995). The author proposes a formal situation calculus framework for diagnostic problem solving which incorporates a theory of action and change.

3. FAULT DIAGNOSIS IN DES

This section recalls formal background about fault diagnosis in discrete-event systems.

3.1 Model

Definition 1. (Model of discrete-event system). A discrete-event system is modelled as a tuple $G = (X, \Sigma, T, x_0)$ where:

- X is a finite set of states;
- Σ is a finite set of events;
- $T \subseteq X \times \Sigma \times X$ is a finite set of transitions;
- x_0 is the initial state of the system.

A discrete event system generates a regular prefix-closed language $L(G) \subseteq \Sigma^\star$ where $\Sigma^\star$ denotes the Kleene closure of Σ. Each word w of $L(G)$ represents a finite sequence of events occurring on the system from the initial state x_0. The language $L(G)$ is prefix-closed as any prefix u of w (i.e. $\exists v \in \Sigma^\star, w = u.v$) is also part of $L(G)$. The set $\Sigma_f \subseteq \Sigma$ denotes the set of faults of the system. In order to perform diagnosis, it is then required to observe the system with the help of sensors. Sensors implement a function $Obs : \Sigma \rightarrow (\Sigma_o \cup \{\varepsilon\})$ that associates to any event of the system, either an observable event of Σ_o or the empty event ε. This function is called the observation mask in Jiang and Kumar (2004). If an event $e \in \Sigma$ is fully observable by the sensors, then $Obs(e) = e$, if e is not observable then $Obs(e) = \varepsilon$. It is also possible that two different events e_1 and e_2 cannot be discriminated by the sensors, in this case $Obs(e_1) = Obs(e_2) = o \neq \varepsilon$. By extension, let $w \in L(G)$, $Obs(w)$ denotes the observable trace of w and is recursively defined as follows:

$$\begin{aligned} Obs(w) &= \varepsilon \text{ if } w = \varepsilon \\ &= Obs(u).Obs(v) \text{ if } u \in \Sigma, v \in \Sigma^\star \end{aligned}$$

Definition 2. (Observed system). An *observed system* is represented as a pair (G, Obs) where G is a DES model and Obs is the observation mask.

To perform diagnosis, it is required to implement a *monitor* that is able to process any possible observations that is produced by the observed system, that is any observable sequence $Obs(w), \forall w \in L(G)$. The observable language of G, denoted $Obs(L(G))$, is thus a subset of $\Sigma_o^\star$.

3.2 Fault traces

The classical fault diagnosis problem on DES basically consists in recording observations from the system and providing the set of possible faults whose occurrence is consistent with these observations: given the recorded sequence of observations σ, the occurrence of a fault F is consistent with σ if there exists at least a sequence of events w in the model G ($w \in L(G)$) that contains F and that is such that $Obs(w) = \sigma$. In such a case, the observable sequence $Obs(w)$ is said to be a *trace* of F.

Let $F \in w$ denote the fact that the fault F belongs to the sequence of events $w \in L(G)$, the set of traces of F is defined as follows.

Definition 3. (Fault traces). The set of traces $Trc(F)$ of the fault F is the regular language defined as the set of finite observable sequences:

$$Trc(F) = \{\sigma \in \Sigma_o^\star, \exists w \in L(G), (F \in w) \wedge (Obs(w) = \sigma)\}.$$

Given the observations σ, a fault F is a possible solution to the diagnosis problem if $\sigma \in Trc(F)$. There may be several solutions as σ could be a trace of many faults. By extension, we also define the traces of the *absence of* F, denoted $Trc(\neg F)$:

$$Trc(\neg F) = \{\sigma \in \Sigma_o^\star, \exists w \in L(G), (F \notin w) \wedge (Obs(w) = \sigma)\}.$$

From $Trc(F)$ and $Trc(\neg F)$, it follows:

(1) if $\sigma \in Trc(F) \setminus Trc(\neg F)$, the occurrence of F is sure;
(2) if $\sigma \in Trc(\neg F) \setminus Trc(F)$, the occurrence of F is impossible;
(3) if $\sigma \in Trc(F) \cap Trc(\neg F)$, the occurrence of F is possible.

A set of traces is defined as a regular language so it can be represented by a minimal deterministic finite-state machine (Hopcroft et al. (2001)). Let F be a fault (resp. $\neg F$ be the absence of the fault F), $Trc(F)$ (resp. $Trc(\neg F)$) can be represented by the minimal deterministic finite-state machine $M(F) = (S, \Sigma_o, \delta, s_0, tag)$ (resp. $M(\neg F) = (S, \Sigma_o, \delta, s_0, tag)$) where:

- S is a finite set of states;
- Σ_o is the alphabet of the machine;
- $\delta : S \times \Sigma_o \rightarrow S$ is the transition function;
- $s_0 \in S$ is the initial state;
- $tag : S \rightarrow \{F\text{-}possible, F\text{-}impossible\}$ (resp. $tag : S \rightarrow \{\neg F\text{-}possible, \neg F\text{-}impossible\}$).

Let $\delta^\star$ denote the extension of δ to sequences, that is: for all states $s \in S$, $\delta^\star(s, \varepsilon) = s$ and for any sequence $\sigma \in \Sigma_o^\star$ and any observable event $o \in \Sigma_o$, $(\delta^\star(s, \sigma.o) = s') \equiv (\delta(\delta^\star(s, \sigma), o) = s')$. For all states $s \in S$, $tag(s) = F\text{-}possible$ (resp. $\neg F\text{-}possible$) iff there exists $\sigma \in Trc(F)$ (resp. $\sigma \in Trc(\neg F)$) such that $\delta^\star(s_0, \sigma) = s$.

For the sake of simplicity in the following, the machines $M(F)$ and $M(\neg F)$ are extended to be complete by adding a fictive state s_{nonAdm}. For all states s, such that there is

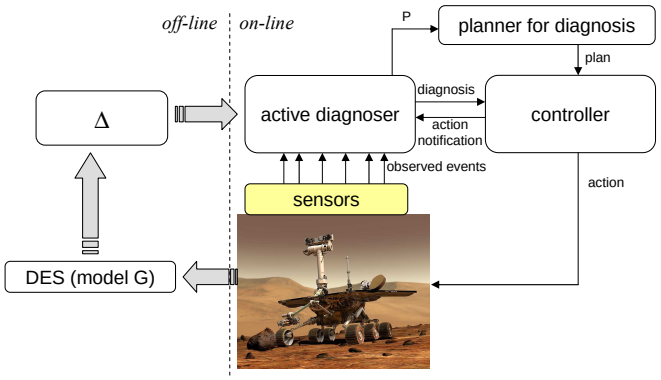

Fig. 1. Part of the embedded architecture for active diagnosis.

no state s' in S with $\delta(s,o) = s'$, the definition of δ is completed with $\delta(s,o) = s_{nonAdm}$. Finally, $tag(s_{nonAdm}) = F\text{-}impossible$ in $M(F)$ and $tag(s_{nonAdm}) = \neg F\text{-}impossible$ in $M(\neg F)$. Any sequence of transitions from state s_0 to state s_{nonAdm} represents a sequence of events of $\Sigma_o^\star$ that is not admissible by G (not in the observable language of G). In the following, we will always consider the complete version of $M(F)$ and $M(\neg F)$.

Both machines $M(F)$ and $M(\neg F)$ can be used as a monitor of the system and determine at any time whether the current sequence of observations is a trace of F or not. For instance, if $M(F)$ is in state s after observing the sequence σ with $tag(s) = F\text{-}possible$ then $\sigma \in Trc(F)$, if $tag(s) = F\text{-}impossible$ then $\sigma \notin Trc(F)$. The classical diagnoser (Sampath et al. (1996)) can be trivially characterised by the synchronous product of the set of machines $M(F), F \in \Sigma_f$ (Pencolé et al. (2006)).

4. ACTIVE DIAGNOSIS

This section extends the classical framework in order to perform *active diagnosis* on DES. As opposed to classical diagnosers like the one defined in Sampath et al. (1996) or more recently the ones defined in Pencolé et al. (2006), the proposed active diagnoser takes into account the fact that at a given time, it may be possible to act on the system and get a better diagnosis if such an action is performed. As shown in Figure 1, the purpose of the active diagnoser modelled by the machine Δ is firstly to provide a diagnosis for any observable situation (like any other diagnoser) and secondly to provide information about how useful the trigger of an active diagnostic session could be. Furthermore, it provides the input data as a planning problem P to a planner that could plan actions for refining the diagnosis (see Section 5).

4.1 Model extension for active diagnosis

From the system model point of view, an *action* performed by the controller is represented as a controllable event (Ramadge and Wonham (1989)).

Definition 4. (Action). An action is an event of the system that occurs only if the controller of the system performs the action.

In the following, we suppose that the controller notifies the diagnoser about the performed action which means that any action is actually fully observed by the diagnoser (see Figure 1). Let $\Sigma_a \subseteq \Sigma$ be the set of available actions on the observed system modelled by (G, Obs) then it follows that:

$$\forall a \in \Sigma_a, (a \in \Sigma) \wedge (Obs(a) = a)$$

in other words $\Sigma_a \subseteq \Sigma_o$.

Finally, in order to ensure that the controller is always able to perform an action on the system, the study of the active diagnosis problem is restricted to the subclass of discrete-event systems in which the following hypothesis holds.

Hypothesis 5. From any state $x \in X$, it is always possible to perform an action $a \in \Sigma_a$ after the occurrence of a finite sequence of reactive events $e \in \Sigma \setminus \Sigma_a$.

In the case that this hypothesis does not hold, it means that the system can reach a state from which the controller cannot perform any more actions. In this case, the active diagnosis problem has trivially no solution.

For the sake of readability, the definition of the active diagnoser Δ is decomposed into two steps. First, we introduce a specialised active diagnoser $\Delta(F)$ for a given fault F and then present the active diagnoser as the union of these specialised diagnosers.

4.2 Specialised active diagnoser

Like for any other diagnosers, the definition of an active diagnoser relies on the fundamental notion of fault traces. The challenge of active diagnosis is to refine diagnosis by pruning ambiguities about the presence or the absence of faults at a given time by acting on the system. The definition of the diagnoser $\Delta(F)$ relies on the following facts:

(1) if there is an active solution for diagnosing the fault F with certainty then this solution produces an observable trace that necessarily belongs to $Trc(F) \setminus Trc(\neg F)$;
(2) if there is an active solution for diagnosing the absence of the fault F with certainty then this solution produces an observable trace that necessarily belongs to $Trc(\neg F) \setminus Trc(F)$.

Let $M(F) = (S_1, \Sigma_o, \delta_1, s_{01}, tag_1)$ denote the machine representing $Trc(F)$ and $M(\neg F) = (S_2, \Sigma_o, \delta_2, s_{02}, tag_2)$ denote the one of $Trc(\neg F)$ (see section 3.2).

Definition 6. The active diagnoser of F is the complete deterministic finite-state machine $\Delta(F) = (S, \Sigma_o, \delta, s_0, tag)$ where:

- $S = S_1 \times S_2$ is the set of states;
- Σ_o is the set of observable events;
- $\delta : S \times \Sigma_o \to S$ is the transition function:
$$\forall s_1 \in S_1, s_2 \in S_2, o \in \Sigma_o, \quad \delta((s_1,s_2),o) = (\delta_1(s_1,o), \delta_2(s_2,o));$$
- $s_0 = (s_{01}, s_{02})$ is the initial state;
- $tag : S \to \{F\text{-}safe, F\text{-}sure, F\text{-}discriminable, F\text{-}undiscriminable, nonAdmissible\}$ is incrementally defined as follows:
  (1) $\forall s = (s_1, s_2) \in S$ such that $tag_1(s_1) = F\text{-}possible$ and $tag_2(s_2) = \neg F\text{-}impossible$, $tag(s) = F\text{-}sure$;

(2) $\forall s = (s_1, s_2) \in S$ such that $tag_1(s_1) = F\text{-}impossible$ and $tag_2(s_2) = \neg F\text{-}impossible$, $tag(s) = nonAdmissible$;
(3) $\forall s = (s_1, s_2) \in S$ such that $tag_1(s_1) = F\text{-}impossible$ and $tag_2(s_2) = \neg F\text{-}possible$, $tag(s) = F\text{-}safe$;
(4) $\forall s = (s_1, s_2) \in S$ such that $tag_1(s_1) = F\text{-}possible$ and $tag_2(s_2) = \neg F\text{-}possible$, two cases hold:
   (a) there exists a sequence of transitions from s to a state s' such that $tag(s') = F\text{-}sure$ or $tag(s') = F\text{-}safe$, in this case $tag(s) = F\text{-}discriminable$;
   (b) there is no such a sequence, then $tag(s) = F\text{-}undiscriminable$.

Proposition 7. $\Delta(F)$ is a monitor of G.

By Definition 6, any possible sequence of observable events $\sigma \in \Sigma_o^\star$ is recognised by the active diagnoser so any observable sequence $\sigma \in Obs(L(G)) \subseteq \Sigma_o^\star$ as well. If $\sigma \in Obs(L(G))$, the target state $s = \delta^\star(s_0, \sigma)$ is such that $tag(s) \neq nonAdmissible$. Any sequence of observable events that reaches the state s such that $tag(s) = nonAdmissible$ belongs to $\Sigma_o^\star \setminus Obs(L(G))$, it does not correspond to a behaviour of G. The fact that the active diagnoser is deterministic results from the synchronisation of $M(F)$ and $M(\neg F)$ that are deterministic.

Theorem 8. Let $\Delta(F) = (S, \Sigma_o, \delta, s_0, tag)$ be the active diagnoser of F, then $\forall s \in S$:

(1) $tag(s) = F\text{-}sure \equiv$ for any observable sequence σ such that $\delta^\star(s_0, \sigma) = s$, for any sequence of events w of the system G such that $Obs(w) = \sigma$, $F \in w$;
(2) $tag(s) = F\text{-}safe \equiv$ for any observable sequence σ such that $\delta^\star(s_0, \sigma) = s$, for any sequence of events w of the system G such that $Obs(w) = \sigma$, $F \notin w$;
(3) $tag(s) = F\text{-}discriminable \equiv$ for any observable sequence σ such that $\delta^\star(s_0, \sigma) = s$, for any sequence of events w of the system G such that $Obs(w) = \sigma$, there exists at least one finite sequence of observable events $\sigma_{s \to s'}$ such that $\delta^\star(s, \sigma_{s \to s'}) = s'$ and $tag(s') \in \{F\text{-}sure, F\text{-}safe\}$;
(4) $tag(s) = F\text{-}undiscriminable \equiv$ for any observable sequence σ such that $\delta^\star(s_0, \sigma) = s$, for any sequence of events w of the system G such that $Obs(w) = \sigma$, there is no finite sequence of observable events $\sigma_{s \to s'}$ such that $\delta^\star(s, \sigma_{s \to s'}) = s'$ and $tag(s') \in \{F\text{-}sure, F\text{-}safe\}$.

This result directly follows from Definition 6.

4.3 Active diagnoser

Now, we are finally ready to define the active diagnoser of the system G as the synchronous product of the specialised diagnosers. Let $F_1, \ldots, F_n$ be the set of anticipated faults, let $\Delta(F_i) = (S_i, \Sigma_o, \delta_i, s_{0i}, tag_i)$ be the active diagnoser of fault F_i.

Definition 9. (Active diagnoser). The active diagnoser is the complete deterministic finite state machine $\Delta = (S, \Sigma_o, \delta, s_0, tag)$ where:

- $S \subseteq S_1 \times \cdots \times S_n$ is the set of states;
- Σ_o is the set of observable events;
- $s_0 = (s_{01}, \ldots, s_{0n})$ is the initial state;
- $\delta : S \times \Sigma_o \to S$ is the transition function defined by the following induction:
  (1) $\forall o \in \Sigma_o, \delta(s_0, o) = (\delta_1(s_{01}, o), \ldots, \delta_n(s_{0n}, o))$;
  (2) from any couple of states (s, s') with $s' = (s'_1, \ldots, s'_n)$ such that $\delta(s, o) = s'$, it follows $\forall o' \in \Sigma_o, \delta(s', o') = (\delta_1(s'_1, o'), \ldots, \delta_n(s'_n, o'))$;
- $tag : S \to \{F_1\text{-}safe, F_1\text{-}sure, F_1\text{-}discriminable, F_1\text{-}undiscriminable, nonAdmissible\} \times \cdots \times \{F_n\text{-}safe, F_n\text{-}sure, F_n\text{-}discriminable, F_n\text{-}undiscriminable, nonAdmissible\}$ is defined as: $tag(s_1, \ldots, s_n) = (tag_1(s_1), \ldots, tag_n(s_n))$.

The active diagnoser Δ of G gathers the set of specialised diagnosers. For any sequence of observations σ, the active diagnoser reaches a state $s = \delta^\star(s_0, \sigma)$ that provides the following pieces of information:

(1) *Current diagnosis:* for any fault F, it provides the current status about the presence of F. For instance, if $\forall i \in \{1, \ldots, n\}$, the status of F_i is safe, it means that no fault has occurred.
(2) *Status of active diagnostic session:* it provides a general view about how useful the trigger of an active diagnostic session is. The optimal goal of such a session is to disambiguate between the presence and the absence of any fault F such that $tag(s)$ contains $F\text{-}discriminable$ if possible (see next section for details). If there is a plan of actions that reaches these goals, its execution is necessarily represented as a sequence of transitions from state s to a state s' such that $tag(s')$ contains $F\text{-}sure$ or $F\text{-}safe$. Any execution from state s to the state s' that contains $nonAdmissible$ is not admissible by G.

5. PLANNING FOR DIAGNOSIS

5.1 Planning problem formulation

The last section allows us to obtain the active diagnoser $\Delta = (S, \Sigma_o, \delta, s_0, tag)$. Suppose that Δ reached a state s^I where it exists at least one i in $\{1, \ldots, n\}$ such that $tag(s^I) = (\ldots, F_i\text{-}discriminable, \ldots)$. Then by Theorem 8: there exists at least one finite sequence of observable events σ such that $\delta^\star(s^I, \sigma) = s'$ and $tag_i(s'_i) \in \{F_i\text{-}sure, F_i\text{-}safe\}$. Let π_σ be the sequence of actions resulting from the projection of σ to Σ_a, then π_σ is a possible sequence of actions for deciding whether F_i has certainly occurred or not. The planning problem is to choose the best admissible sequence of actions to perform in this case in order to refine the diagnosis.

For the sake of readability, the formulation of the planning problem is decomposed into two steps in the same way that the active diagnosis. First, we formulate the problem for one fault F and then present the entire planning problem.

Planning problem for a single fault Suppose that Δ reached a state s^I for which it exists **only one** F that is tagged $F\text{-}discriminable$; the other faults F_j are tagged $F_j\text{-}sure$ or $F_j\text{-}safe$. Then the set of admissible sequences of actions for diagnosing F with certainty is contained in AP_F where:

$$AP_F = \{\pi_\sigma | \exists \sigma, \delta(s^I, \sigma) = s', \quad s' \text{ is tagged } F\text{-}sure \text{ or } F\text{-}safe\}$$

By Theorem 8:

Proposition 10. If it exists a sequence of actions that can be performed by the system and that refines the diagnosis, it is in AP_F.

The objective here is to reformulate the problem into a classical planning problem. Thus, we decide to define the problem as a tuple $P = (s_i, S, A, T, Goal, C)$ where the following pieces of information are extracted from the active diagnoser:

- the initial state s_i is s^I;
- $A = \Sigma_a$;
- S the finite set of states of Δ;
- $T : S \times A \times S \rightarrow \{0, 1\}$ is the transition function:
  - $T(s, a, s') = 1$ if s' can be reached when a is performed in s, i-e $\pi_{s \rightarrow s'} = a$;
  - $T(s, a, s') = 0$ otherwise [1] ;
- The set of goal states $Goal = \{s' \in S$ such that s' is tagged *F-sure* or *F-safe*$\}$;
- C a criterion to minimize.

The planning problem may be formulated as follows: finding a conditional plan of the type:

action;
if *observation*$_1$ **then** *action*$_1$
else if *observation*$_2$ **then** *action*$_2 \ldots$,

that will perform one of the sequences:
$<$ *action*; *action*$_1, \ldots >$; $<$ *action*; *action*$_2, \ldots >$, knowing that at least one of them leads the diagnoser from s_i to a state $s_p \in Goal$, minimizing a criterion C.

Definition 11. A plan is *admissible* from s_i iff it performs at least one sequence of actions $\{a_{i+1}, a_{i+2}, \ldots, a_p\}$ such that it exists $s_p \in Goal$ for which

$$\forall k \in \{i+1, \ldots, p\}, T(s_{k-1}, a_k, s_k) = 1$$

General planning problem Suppose that Δ reached a state s^I for which **there exists a subset** $D \subseteq \{1, \ldots, n\}$ such that for all i in D, s^I is tagged $F_i - discriminable$ and for all j in $\{1, \ldots, n\} \setminus D$, s^I is tagged F_j-*sure* or F_j-*safe*. Then the set of sequences of actions for diagnosing all the F_i for which i is in D with certainty is contained in AP where:

$$AP = \{\pi_\sigma | \exists \sigma, \delta(s^I, \sigma) = s', \\ \forall i \in D, s' \text{ is tagged } F_i\text{-}sure \text{ or } F_i\text{-}safe\}$$

The set AP may be the empty set. Actually,

Proposition 12. The set of sequence of actions that are admissible for diagnosing all the F_i for which i is in D is contained in the intersection of the AP_{F_i} sets.

$$AP = \bigcap_{i \in D} AP_{F_i}.$$

The reformulation of the problem into a classical planning problem is a tuple $P = (s_i, S, A, T, Goal, C)$ where s_i, S, A, T and C are the same as the one of the planning problem for a single fault and *Goal* is defined by

$$Goal = \{s' \in S \text{ such that } \forall i \in \{1, \ldots, n\}, \\ \text{if s is tagged } F_i\text{-}discriminable \\ \text{then s' is tagged } F_i\text{-}sure \text{ or } F_i\text{-}safe\}.$$

[1] $T(s, a, s') = 0$ in particular if a is an action that cannot be performed in s for security or physical reason.

The planning problem remains the same.

5.2 Planning algorithm

The on-line planning algorithm consists in these steps:

(1) According to the active diagnoser, find all the admissible conditional plans from s_i;
(2) According to the criterion C computed for each admissible plan, choose the best plan and send it to the controller that applies it;
(3) If the current state of Δ is in *Goal*, then *END*; if not, iterate in (1) if there exists at least one fault that is still tagged *F-discriminable*.

Finding all the admissible plans For finding all the admissible plans, we use an iterative depth-first search that explores the graph as it was an AND-OR tree (Nilsson, 1998) where OR nodes correspond to system states and AND nodes correspond to actions. We choose a depth-first search approach because once the first plan has been computed, the algorithm is any-time. So, even if all branches of the AND-OR tree are not explored for time reason, there exists an admissible plan.

Finally, we come back to a widely studied problem, even in the diagnosis area that is the exploration of an AND-OR tree (Pattipati and Dontamsetty (1992)). The solutions are several conditional plans. The planning solver has then to choose the best plan to perform.

Criterion Several criteria could be considered:

- Similarity between the plan and the mission plan: a plan that performs a sequence of actions similar to the mission plan would be preferred because it will let the system in a state that will be compatible with the mission. Similarly, a plan that performs a sequence of actions that is opposed to the mission achievement should be discarded.
- As most of the actual planners, it is possible to minimize only the number of actions to reach a goal state;
- Probability of the fault that could be incriminated by the actions plan;
- Criticity of the fault (reward);
- Cost of actions (cost) ...

Two main solutions could be investigated:

- Constructing a single aggregate objective function in a weighted sum of all these criteria: this solution is the simplest and the most intuitive solution, but there is a risk that some criteria compensate others and coming up with meaningful combinations of weights can be challenging.
- a multi-criteria optimization: each criterion above becomes a component of a vector that represents the plan. Many techniques exist and can be found in Branke et al. (2007).

6. EXPERIMENT

The generation of the active diagnoser has been implemented in C++ and tested in the framework of autonomous vehicles. We consider an electrical circuit with

one engine and one switch. The power source always works whereas both switch and engine may break down.

In order to be as clear as possible, the hypothesis of single fault is taken. There are three possible actions: *switch-on*, *switch-off* and *test-switch*. In the model, the switch test is supposed to be right as long as there is no action made. When one action is performed, the result of the test is supposed to become unknown. There are two observations: *test_obs* that can be either *ok*, *fail* (the switch works or not), or *unknown* (no available information about the switch status); *engine_obs* that can be either *on* or *off*. The two possible faults are $F_0 \equiv broken\_engine$ or $F_1 \equiv broken\_switch$. This experiment has the advantage to be quickly understood and may easily be extended (for example with several switches in parallel).
The generated system model has 50 states and 86 transitions. The active diagnoser has 39 states and 312 transitions. For the sake of readability, we present only a few trajectories extracted from the active diagnoser (Figure 2). In the initial state, the engine and the switch are off. Then, the engine is switched on. It works for a while, then it stops. The state is ambiguous: either the engine is broken or the switch. The solution is to use active diagnosis and plan actions that will refine the diagnosis. The plan will be the following:
test-switch;
if (*test_obs* = *ok* and *engine_obs* = *off)* **then** END: $tag(s) = (F_0\text{-}sure, F_1\text{-}safe)$
else if (*test_obs* = *fail* and *engine_obs* = *off)* **then** END: $tag(s) = (F_0\text{-}safe, F_1\text{-}sure)$.
Thanks to action, the diagnoser is able to give a non-ambiguous diagnosis.

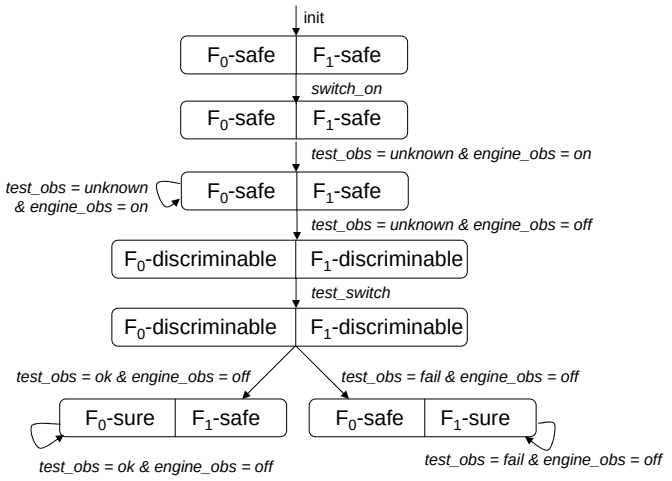

Fig. 2. Trajectories extracted from the active diagnoser of the switch-engine system

7. CONCLUSION AND FUTURE WORKS

This paper defines the concept of active diagnosis for discrete-event system without reducing the number of possible actions performed by the controller. This point is crucial for autonomous vehicles that have to realize a mission. The formal definition of an active diagnoser and the implementation of its generation in the framework of finite-state automata provide solid foundations for further works. This paper shows how the active diagnoser can be transformed into a planning problem and gives some key ideas for its resolution. An experiment shows that the synthesis of an active diagnoser is possible and illustrates its use.

In the future, the planning algorithm has to be improved and implemented. The exhaustive enumeration of all the possible plans would be intractable in practice. We could use heuristic search in order to choose the best action to perform in case of multiple choice, as in the AO* algorithm and prune the search tree.
Finally, the main problem that remains is that the result of the on-line planning for active diagnosis interact with the mission plan. The management of this interaction would be treated in the future. A more sophisticated way is to interlace both plans: the planner for diagnosis and the mission planner will be considered as two communicating processes and will negotiate the plan to be executed. This can be considered as a problem of distributed planning.

REFERENCES

M. Bayoudh, L. Travé-Massuyès, and X. Olive. Towards active diagnosis of hybrid systems. In *Proc of DX'08*, 2008.

J. Branke, K. Deb, K. Miettinen, and R. Slowinski. *Practical Approaches to Multi-Objective Optimization.* Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), 2007.

E. Chanthery, M. Barbier, and J-L Farges. On-line mission planning for autonomous vehicles. In *ICAPS'05 - International Workshop on "Planning under Uncertainty for Autonomous Systems"*, 2005.

J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to automata theory, languages, and computation (2nd ed).* Addison-Wesley, 2001.

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

S. McIlraith. Incorporating action into diagnostic problem solving (an abridged report). In *Working Notes of the 1995 AAAI Spring Symposium on Extending Theories of Action: Formal Theory and Practical Applications*, pages 139–144, 1995.

N.J Nilsson. *Artificial intelligence, a new synthesis.* 1998.

K. R Pattipati and M. Dontamsetty. On a generalized test sequencing problem. *IEEE transactions on systems, man, and cybernetics*, 22(2):392–396, 1992.

Y. Pencolé, D. Kamenetsky, and A. Schumann. Towards low-cost diagnosis of component-based systems. In *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Process (SAFEPROCESS)*, 2006.

P.J.G. Ramadge and W.M. Wonham. The control of discrete event processes. In *IEEE Proceedings: Special issue on Discrete Event Systems*, 1989.

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

M. Sampath, S. Lafortune, and D. Teneketzis. Active diagnosis of discrete-event systems. *IEEE Transactions on Automatic Control*, 43, 1998.