An Algorithm for Active Diagnosis of Hybrid Systems Casted in the DES Framework

Mehdi Bayoudh, Louise Travé-Massuyès

An Algorithm for Active Diagnosis of Hybrid Systems Casted in the DES Framework

Mehdi Bayoudh * **and Louise Travé-Massuyès** *

* *LAAS-CNRS, Université de Toulouse, France*
(e-mail: {bayoudh, louise}@laas.fr).

Abstract: On-line diagnosis must accommodate the existing sensoring capabilities of a system, which often results in limited diagnosability. However, although faults may not be always discriminable, there are generally operating modes of the system in which they are. Active diagnosis relies on applying specific inputs to the system so as to exhibit additional symptoms that help refining the diagnosis. The idea of this paper is to use the diagnosability properties to drive the system towards modes with increased diagnosability with respect of safety considerations. A new finite state machine called *the active diagnoser* is defined by abstracting continuous dynamics and taking into account controllability and safety constraints. The active diagnosis problem is then formulated as a conditional planning problem. Hence, the active diagnoser is transformed in an AND-OR graph and active diagnosis plans are computed by an appropriate graph exploration algorithm.

Keywords: Active diagnosis, conditional planing, hybrid systems, active diagnoser, diagnosability.

On-line diagnosis is often approached as a passive task that takes as input the available observations provided by the sensoring devices instrumenting a physical system and returns an estimation of its state, interpreted in terms of the status of each of the components. However diagnosis is originally defined as a process (c.f. Hamscher et al. (1992)) that interlinks the determination of a belief state and the proposal of new tests that provide additional information allowing the diagnoser to refine the belief state and ultimately end with a non ambiguous state estimation. This way to go is quite common for solving post-mortem diagnosis problems and the diagnosis is often formulated as a test sequencing problem or related in some way to testing as presented in Struss (1994); Abramovici et al. (1999); Nicolaidis and Zorian (1998). The proposed tests can take the following forms or a mixture of them: new variables to be sensored, new input pattern defined by specific signals to be applied to the system or new configuration in which the system should be put. Referring to on-line diagnosis, there are very few works putting diagnosis and testing together. There are two main reasons for that : the first one is that measurements are generally limited to a small number defined by the available sensors and the second is that the system's inputs are used to achieve the normal operation tasks of the system. Nevertheless, interlinking diagnosis and testing on-line, i.e. performing active diagnosis, is possible and may be necessary in some application domains, particularly those requiring autonomy.

Among the very few works dealing with active diagnosis, the most representative ones are Sampath et al. (1998) and Niemann (2006) for Discrete-Event Systems (DES) and Continuous Systems (CS), respectively.

This paper presents a framework to achieve active diagnosis for hybrid systems. Starting with an ambiguous belief state, our method calls for diagnosability analysis results to determine a new system configuration in which fault candidates can be discriminated. The command inputs to be applied to the system to drive it into this configuration are then determined, paying attention to avoid states that could be dangerous for the system. A finite state machine called *the active diagnoser* is defined to perform the active diagnosis guided by diagnosability and controllability properties of the system and with respect to safety considerations. The paper is organized as follows: our hybrid modeling framework is presented in Section 1, followed by the background results that are used by the active diagnosis approach in Section 2. The active diagnosis problem and the proposed active diagnosis scheme are detailed in Section 3. An illustrative example is presented in Section 4. Finally Section 5 concludes the paper.

1. HYBRID MODELING FRAMEWORK

A hybrid system is modeled as a hybrid automaton (c.f. Henzinger (1996)), $S = (\zeta, Q, \Sigma, T, C, (q_0, \zeta_0))$, where:

- ζ is the set of continuous variables, which includes observable and non observable variables. The set of observable variables is denoted by ζ_{OBS}.
- Q is the set of discrete states. Each state $q_i \in Q$ represents a behavioral mode of the system. It includes nominal and anticipated fault modes.
- Σ is the set of events that correspond to discrete control inputs, spontaneous mode changes and fault events. Events corresponding to spontaneous mode changes are triggered upon guards that depend on continuous variables. The event set Σ is partitioned as $\Sigma = \Sigma_{uo} \cup \Sigma_o$, where Σ_{uo} (Σ_o) is the unobservable (observable) event set. Without loss of generality, we assume that fault events are unobservable (otherwise, these faults are obviously diagnosable).
- T is the partial transition function, $T \subseteq Q \times \Sigma \rightarrow Q$.
- $C = \bigcup C_i$ is the set of system constraints linking continuous variables. It represents the set of differential and algebraic equations modeling the continuous behavior of the system within operating modes.
- $(\zeta_0, q_0) \in \zeta \times Q$, is the initial condition.

The hybrid behavior is seen as the contribution of two underlying discrete-event and continuous system behaviors as presented in Bayoudh et al. (2008b).

2. BACKGROUND

In Bayoudh et al. (2008a) and Bayoudh et al. (2008b), we propose an approach for hybrid systems diagnosis and diagnosability analysis, respectively. It uses a hybrid modeling that is consistent with the one presented in section 1. This paper relies on the established results that are recalled below.

2.1 The mode signature

To check the consistency between the system model and the incoming observations, a set of consistency indicators, based on a set of constraints C_{obs_i} that involve only observable continuous variables and can therefore be evaluated, is linked with every operating mode $q_i \in Q$. Constraints of C_{obs_i} are determined by eliminating non observable variables in the constraints belonging to C_i. A consistency indicator called residual is associated to each constraint $C^k_{obs_i} \in C_{obs_i}$ and denoted r_{ik}. The residual is a boolean indicator. It is zero when the constraint $C^k_{obs_i}$ is satisfied, otherwise it is equal to 1.

Definition 1. (Mode Signature). Given the tuple $R^{q_i} = [r_{i1}, r_{i2}, ..., r_{iN_{r(q_i)}}]$ of system residuals in mode q_i, where $N_{r(q_i)}$ is the number of residuals in mode q_i, the q_i-mirror signature of mode q_j is given by the vector $S_{j/i} = [s_{1_{j/i}}, ..., s_{N_{r(q_i)_{j/i}}}]^T = [R^{q_i}(\zeta_{OBS_{q_j}})]^T$, where $\zeta_{OBS_{q_j}}$ denotes the value of observable variables when the system mode is q_j. The signature of a mode q_j is the vector obtained by the concatenation of all the mirror signatures of q_j, $Sig(q_j) = [S^T_{j/1}, S^T_{j/2}, ..., S^T_{j/j}, ..., S^T_{j/n}]^T$, where n is the number of system modes [1].

2.2 Abstraction of the continuous dynamics in terms of discrete events

Let us assume that the dynamics of the discrete control inputs are slower than the dynamics of residual generators (mode signatures establish between two consecutive discrete events). The abstraction function f_{CS_DES} associates a discrete-event that captures the change of mode signatures, to each discrete transition. This function aims to define Σ^{Sig}, as the set of discrete events issued from the abstraction of continuous dynamics.

$$f_{CS\_DES} : Q \times T(Q, \Sigma) \longrightarrow \Sigma^{Sig}$$

$$(q_i, q_j) \longmapsto \begin{cases} Ro_{ij} \in \Sigma^{Sig}_o \text{ if } Sig(q_i) \neq Sig(q_j) \\ Ruo_{ij} \in \Sigma^{Sig}_{uo} \text{ if } Sig(q_i) = Sig(q_j) \end{cases}$$

- Σ^{Sig}_o is a set of observable events, generated when the mode signature of the source mode is different from the mode signature of the destination mode.
- Σ^{Sig}_{uo} is a set of unobservable events generated when the mode signature of the source mode is equal to the mode signature of the destination mode.
- Σ^{Sig} is defined as $\Sigma^{Sig}_o \cup \Sigma^{Sig}_{uo}$.

[1] In our approach, nominal and fault modes have the same status and the signature of a given mode anticipates how it should be seen in terms of the indicator tuples of the different modes of the system (including itself).

2.3 Hybrid language and hybrid trajectories

The abstraction of the continuous dynamics changes in terms of discrete events allows us to define the language of the hybrid system, which describes the evolution of the system behavior. We denote by $\Sigma_{hyb} = \Sigma \cup \Sigma^{Sig}$ the alphabet that contains "natural" discrete events and events modeling signature switches. Σ_{hyb} can be partitioned into $\Sigma_{hyb} = \Sigma_{hyb_o} \cup \Sigma_{hyb_{uo}}$ with $\Sigma_{hyb_o} = \Sigma_o \cup \Sigma^{Sig}_o$ and $\Sigma_{hyb_{uo}} = \Sigma_{uo} \cup \Sigma^{Sig}_{uo}$.
The behavior of the hybrid system is modeled by the prefix-closed language $L(S) \subseteq \Sigma^*_{hyb}$ over the event alphabet Σ_{hyb}, where Σ^*_{hyb} denotes the set of all finite strings of elements of the set Σ_{hyb} including the empty string (Σ^*_{hyb} is called the Kleene-Closure of Σ_{hyb} as presented in Ramadge and Wonham (1989)). A trajectory of the hybrid system is represented by a string of events of the hybrid alphabet Σ_{hyb}. The hybrid language $L(S)$ can be generated by its finite state generator representation (c.f. Ramadge and Wonham (1989)). In this paper, this automaton is called *the behavior automaton* denoted $B_A(S) = (Q_{beh}, \Sigma_{hyb}, T_{beh}, q_0)$ and mixes both "natural" discrete events and signature switches. In practice, non observable signature switches (Σ^{Sig}_{uo}) are useless because they do not convey additional information, hence they are not considered. In this case, $\Sigma_{hyb} = \Sigma \cup \Sigma^{Sig}_o$ (i.e. $\Sigma_{hyb_{uo}} = \Sigma_{uo}$).

2.4 The diagnoser construction

The diagnoser is a finite state machine $Diag(B_A(S)) = (Q_D, \Sigma_D, T_D, q_{D_0})$ built from the behavior automaton of the hybrid system and used on one hand to perform the on-line diagnosis and on the other hand to check the diagnosability property of the hybrid system as presented in Bayoudh et al. (2008b) and in Bayoudh et al. (2008a), respectively. The diagnoser construction is provided in Bayoudh et al. (2009). This paper recalls the concepts provided in Bayoudh et al. (2008a), to check the diagnosability of hybrid systems.

Definition 2. Uncertain state . Given a diagnoser state $q_D \in Q_D$, this state is F_i-uncertain if F_i does not belong to all the labels of q_D, whereas F_i belongs to at least one label of q_D. Formally: a state $q_D \in Q_D$ is F_i-uncertain if $\exists (q, l), (q', l') \in q_D$, such that $F_i \in l$ and $F_i \notin l'$.

Proposition 1. The hybrid system $S = (\zeta, Q, \Sigma, T, C, (\zeta_0, q_0))$ is not diagnosable if and only if the associated diagnoser computed from the corresponding behavior automaton contains an indeterminate cycle i.e. a cycle composed of F_i-uncertain states for which there exist two corresponding cycles in $B_A(S)$: one involves only states that carry the fault label F_i in their labels in the cycle in $Diag(B_A(S))$ and the other does not.

3. ACTIVE DIAGNOSIS OF HYBRID SYSTEMS CASTED IN THE DES FRAMEWORK

Let us assume that a hybrid system is continuously monitored and that its state is tracked following the passive diagnosis approach proposed in Bayoudh et al. (2008b). Assume that the current belief state returned by the diagnoser is faulty and uncertain, i.e. several faults are candidate. This is the starting point of an active diagnosis session. The active diagnosis problem is formulated as a conditional planning problem. From an uncertain state of the diagnoser, the plan defines how to find a controllable path leading to a certain state. The search of active diagnosis actions is guided by the observable response of the

system on active control inputs.
What is important to notice is that even when the conditions for non diagnosability as stated by Proposition 1 hold, there may be a way to enforce a sequence of transitions to drive the system towards a certain state of the diagnoser. Indeed, an indeterminate cycle of the diagnoser only indicates that the system may get stuck in the cycle. Therefore, we distinguish two situations for which an active diagnosis session is triggered:

- the uncertain state belongs to an indeterminate cycle, in this case the system is non diagnosable w.r.t this state and the active diagnosis aims at cutting the indeterminate cycle and bringing the diagnoser in a certain state.
- the uncertain state does not belong to an indeterminate cycle, in this case the system is diagnosable w.r.t this state, however, the active diagnosis aims at energizing the diagnoser to leave this state (the system does not wait for observations, the controller sets them off).

3.1 Controllable and induced controllable paths

Active diagnosis is closely linked to the property of controllability of the system. Indeed, active diagnosis consists in determining paths from the starting uncertain state of the active diagnoser to target states in which the diagnosis is precise (or more precise). Consequently the dynamics of the system along these paths must be controllable to allow the system to be driven to the target states. Hence, the concepts of *controllable events* and *induced controllable events* are introduced.
Let us consider the hybrid language $L(S) \subseteq \Sigma^*_{hyb}$ and let us call $\Sigma_c \subseteq \Sigma_o \subseteq \Sigma_{hyb}$ the set of controllable events [2] .

Definition 3. Controllable event. Controllable events fall in one of the categories below:

- discrete control inputs, $c \in \Sigma_c$ (for example, the software commands sent by embedded calculators).
- events, $\sigma_u \in \Sigma_c$, corresponding to spontaneous mode changes when the continuous dynamics model of the source mode are controllable in the sense of Terrell (1999). This means that there always exists a continuous control law u that leads to the occurrence of such event.

The set of possible transitions outgoing fault modes represents all the control actions that can be done to perform active diagnosis. The set of *allowed* control actions is different for the different fault modes, and is a mean to account for safety constraints.

Definition 4. Induced controllable event. Events whose occurrence always follow the occurrence of a controllable event are called induced controllable events and form the set $\Sigma_{hyb_{ic}} \subseteq \Sigma_{hyb}$.

Induced controllable events model the response of the hybrid system after a control action, either a discrete input event or a continuous input signal. Induced controllable events fall in one of the categories below:

- $\Sigma^{Sig}_{ic} \subseteq \Sigma^{Sig}$: the set of induced controllable events that manifest the reaction of the continuous dynamics. Let $R_{ij} \in \Sigma^{Sig}$ denote a discrete event associated to a mode signature change. R_{ij} is an induced controllable event denoted R^{ic}_{ij} if the mode change is controlled by a controllable event.
- $\Sigma_{ic} \subseteq \Sigma$: the set of induced controllable events that manifest the reaction of discrete dynamics. $\sigma \in \Sigma_{ic}$ is an induced discrete event denoted σ_{ic} if its occurrence is always a consequence of a given controllable event.

The set of induced controllable events of the hybrid system is given as $\Sigma_{hyb_{ic}} = \Sigma_{ic} \cup \Sigma^{Sig}_{ic}$. Controllable events are those that provide means to act on the system. Induced controllable events are those that manifest the reaction of the system and allow us to discriminate ambiguous situations.

Definition 5. Controllable path. Consider the hybrid system behavior automaton and its associated hybrid language $L(S) \subseteq \Sigma^*_{hyb}$. A controllable path s is a string of controllable and induced controllable events, $s \in (\Sigma_c \cup \Sigma_{hyb_{ic}})^*$. Formally, a controllable path is $s = \alpha_1\beta_1, ..., \alpha_k\beta_k$, with $\alpha_i \in 2^{\Sigma_c}$ and $\beta_i \in \Sigma_{hyb_{ic}}$, $i = 1..k$, $k \in \mathbb{N}^*$.

A controllable path in the behavior automaton corresponds to a controllable observable path in the corresponding diagnoser.

3.2 The active diagnoser

The idea proposed in this paper is to use the diagnoser to guide the search for the sequence of actions that will disambiguate an ambiguous belief state in the diagnoser. However, in order to suit active diagnosis purposes, the diagnoser must be modified into an *Active Diagnoser* that involves only controllable paths. Classically, the control actions that appear in the diagnoser are supposed to be observed but not necessarily applied. In particular, a control event associated to a transition outgoing an uncertain state of the diagnoser is observed in *at least* one of the underlying faulty modes of the system. In our case, we want to actively apply the control event, which means that it must be applicable in *all* the faulty modes included in the concerned diagnoser state, otherwise it means that the control is forbidden as it may be dangerous in some underlying modes. The diagnoser is hence modified accordingly. Given an uncertain state of the diagnoser, outgoing transitions associated with controllable events are removed if there is no corresponding transition outgoing from *all* the corresponding modes of the behavior automaton.
In our modeling, all enabled control inputs in faulty modes are represented in the mode automaton and can be used to perform active diagnosis. Control inputs that do not appear in the mode automaton must be forbidden and can be dangerous for the system. The active diagnoser is embedded in the classic diagnoser [3] . It defines the sets of states whose uncertainty can be reduced i.e. states in which active diagnosis can be performed.

3.3 Conditional planning for determining an active diagnosis plan

As mentioned before, active diagnosis consists on exciting the hybrid system to exhibit additional observations. Given an uncertain state of the active diagnoser, the active diagnosis problem is how to find controllable paths leading to certain states. In the uncertain state, the active diagnosis is performed by triggering a sequence of consecutive controllable events, observing the system reaction, and deciding about the next sequence. The choice of the consecutive controllable event sequence depends on the last observed induced controllable event. This problem is formulated as a conditional planning in a full observable

[2] Controllable actions are assumed to be observable.

[3] the active diagnoser construction is provided in Bayoudh et al. (2009)

environment problem (c.f. Russel and Norvig (2003); Jimenez and Torras (2000)). The active diagnoser is seen as an *AND-OR* graph. The "OR" nodes (squares) correspond to the selection of a possible sequence of consecutive controllable events, the "AND" nodes (circles) correspond to the resulting induced events as shown in Figure 1. The classical *MINIMAX* algorithm (c.f. Russel and Norvig (2003)) is modified to resolve the conditional planning problem. The algorithm is performed from an uncertain state and searches all controllable paths leading to certain states. The active diagnosis session can be started only from an uncertain state that belongs to the active diagnoser ($q_D \in T_D^{act}(Q_D, \Sigma_{hyb_o})$).

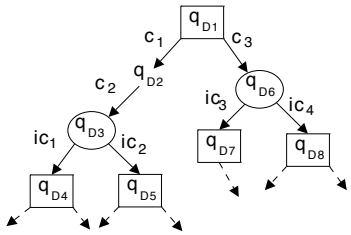

Fig. 1. The active diagnosis seen as a planning problem

3.4 Conditional planning algorithm

The mapping between the active diagnoser and the $AND\text{-}OR$ graph is described as follows:

- the state nodes (OR nodes) S_k of the graph correspond to the states q_{D_k} of the active diagnoser (represented by squares in Figure 1) in which sequences of controllable events are started (for example, states q_{D_1}, q_{D_4}, q_{D_5}, q_{D_7} and q_{D_8} of the active diagnoser shown in Figure 1).
- actions $a_i \in 2^{\Sigma_c}$ are the sequences of consecutive controllable events starting in state nodes (for example, in Figure 1, $a_1 = [c_1, c_2]$ and $a_2 = [c_3]$).
- the observation nodes O_j (AND nodes) of the graph (represented by circles in Figure 1) correspond to the state q_{D_j} of the active diagnoser in which the outgoing transitions are labeled with induced controllable events (for example, states q_{D_3} and q_{D_6} of the active diagnoser shown in Figure 1). An observation o_k outgoing an observation node O_j corresponds to an induced controllable event $\sigma_{ic_k} \in \Sigma_{hyb_{ic}}$ and leads to a next state node (for example $o_1 = ic_1$ and $o_2 = ic_2$ that are associated to the observation node $O_1 = q_{D_3}$ as well as $o_3 = ic_3$ and $o_4 = ic_4$ that are associated to the observation node $O_2 = q_{D_6}$, in the active diagnoser shown in Figure 1). We link in a pair (S_k, o_k) the state node S_k with the observation o_k that corresponds to the induced controllable discrete event leading this state (for example $(S_1, o_1) = (q_{D_1}, \emptyset)$, $(S_2, o_2) = (q_{D_4}, ic_1)$, $(S_3, o_3) = (q_{D_5}, ic_2)$, $(S_4, o_4) = (q_{D_7}, ic_3)$ and $(S_5, o_5) = (q_{D_8}, ic_4)$ in the active diagnoser shown in Figure 1 [4]).
- target states of the graph correspond to certain states of the active diagnoser. Let us assume that the active diagnosis session is started in a diagnoser state uncertain with respect to every fault F_{i_j} in a set $\mathcal{F} = \{F_{i_1}, F_{i_2}, ..., F_{i_n}\}$, i.e. F_{i_1}-uncertain, F_{i_2}-uncertain, ..., and F_{i_n}-uncertain state. Then the set of target states is composed by F_{i_1}, $F_{i_2}, ...$, and F_{i_n} certain states (i.e. $\mathcal{F}$-certain states) and denoted $\Delta_{certain}$. This set can be relaxed to a set of $2^{\mathcal{F}}$-certain states when the active diagnosis is not expected to achieve single fault diagnosis refinement.
- the initial state of the graph is a state node S_1 that corresponds to an uncertain state of the active diagnoser in which the active diagnosis session is started.

[4] Notice that the observation associated to the starting state of the active diagnosis session is the empty element.

Notice that in the active diagnoser shown in Figure 1 the state q_{D_2} is neither an AND node, nor an OR node because it is preceded and followed by a controllable event.
We define the SUCCESSORS function that for each pair (S_k, o_k) of node state and linked observation, associates an action a outgoing S_k and a set of corresponding successor node states (and their associated observations): $\bigcup_k \{(S_{k'}, o_{k'})\}$.
For conditional planning the minimax algorithm is modified as follows. First MAX and MIN nodes become OR and AND nodes. The plan needs to take some action at every state it reaches, but must account for every observation after an action is taken (c.f. Russel and Norvig (2003)). Second, the algorithm needs to return a conditional plan rather than just a single action. At an OR node, the plan is just the action selected, followed by whatever comes next. At an AND node, the plan is a nested series of if-then-else steps specifying subplans for each possible outcome, the tests in these steps being the associated state observations. More details are provided in Russel and Norvig (2003).
The algorithm is a recursive depth-first algorithm, an important point is that it deals with cycles, which often arise in non diagnosable system diagnosers. Indeed, when the current state is identical to a state on the path from the root, then it returns failure. This does not mean that there is no solution from the current state, but simply means that if there is one, it must be reachable from the earlier instance of the current state, so the new instance can be discarded. With this check, we ensure that the algorithm always terminates (the state space that is a part of the active diagnoser is finite) (c.f. Russel and Norvig (2003)).

Algorithm 1 AND-OR graph exploration algorithm

AND-OR-GRAPH-SEARCH()
OR-SEARCH($(S_0, \emptyset)$, [ ])

OR-SEARCH$((S, o), path)$
if $S = certain\text{-}state$ **then return** $the\text{-}empty\text{-}plan$
if $S \in path$ **then return** $failure$
for $(a, state\text{-}observation\text{-}set) \in$ SUCCESSORS$((S, o))$ **do**
$plan \leftarrow$ AND-SEARCH($state\text{-}observation\text{-}set$, $[S|path]$)
if $plan \neq failure$ **then return** $[a|plan]$
return $failure$

AND-SEARCH($state\text{-}observation\text{-}set$, $path$)
for $(S_i, o_i) \in state\text{-}observation\text{-}set$ **do**
$plan_i \leftarrow$ OR-SEARCH$((S_i, o_i), path)$
if $plan_i = failure$ **then return** $failure$
return $[if\ o_1\ then\ plan_1\ else\ \ if\ o_2\ then\ plan_2\ ...$
$else\ if\ o_{n-1} then\ plan_{n-1}\ else\ plan_n]$

3.5 Guaranteed and non guaranteed plans

Algorithm 1 explores the AND-OR graph corresponding to the active diagnoser and returns all controllable paths leading to certain states. Each path is a conditional plan for the active diagnosis. A plan can be then executed by the controller. Two types of plans can be distinguished:

Definition 6. Guaranteed plan. A conditional plan is said to be *guaranteed* if it guarantees to reach a certain state of the active diagnoser from the starting uncertain state.

A guaranteed plan anticipates all the possible resulting induced controllable events following an action included in the plan. In the opposite, the plan is not guaranteed if it contains at least one action for which at least one possible resulting induced controllable event is not anticipated by the plan. When we execute a guaranteed plan, we have the guaranty that the system will reach a target state (a certain state) because all possible resulting situations after an action are taken into account.
In the contrary, when we execute an plan that is not guaranteed, the reachability of a certain state is not guaranteed. After an action, if an induced controllable event that is not anticipated occurs, the plan fails. When there is no guaranteed, the system must be able to choose the best plan among the non guaranteed plans. Costs as well as probabilities can be associated to the control actions in order to help with the decision, in this case, the AND-OR graph exploration could be achieved by AO^* type algorithms based on heuristic search.

3.6 Diagnosability and active diagnosis

This section addresses the link between active diagnosis and diagnosability.

Definition 7. Active diagnosability . The hybrid system is actively diagnosable if for any uncertain state of the diagnoser a *guaranteed* plan exists in the active diagnoser which starts from the uncertain state and leads to a certain state.

Definition 7 ensures that the system controller is able to bring the system out of any uncertain state. This definition is different from the classical diagnosability definition (c.f. Sampath et al. (1995); Bayoudh et al. (2008a)), in the sense that the definition of active diagnosability takes into account not only the observation system, but also the system controller properties.
A relaxed definition of active diagnosability called "non guaranteed active diagnosability" can also be proposed:

Definition 8. Non-guaranteed active diagnosability . The hybrid system is actively diagnosable if for any uncertain state of the diagnoser there exists a plan (guaranteed or non guaranteed) in the active diagnoser, which starts from this uncertain state and leads to a certain state.

4. ILLUSTRATIVE EXAMPLE

Let us consider a hybrid system consisting of three tanks of water, $T1$, $T2$ and $T3$. Valves $V1$ and $V2$ allow the flow to transfer between tanks. Valves are controlled by discrete control inputs $open_{V_1}$, $open_{V_2}$, $close_{V_1}$ and $close_{V_2}$.
The system is equipped with three level sensors that measure

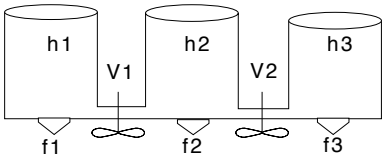

Fig. 2. The three-tanks system

the level of water in each tank. Hence, water levels h_1, h_2 and h_3 are observable. The discrete behavior of the system is described in Figure 3. Every nominal mode models a configuration of the system as shown in Table 1. Fault events f_1, f_2 and f_3 model leaks that may occur in tanks $T1$, $T2$ and $T3$, respectively. A fault event $f_j, 1 \leq j \leq 3$ may occur in any nominal mode $N1, N2, N3$ and $N4$ and leads to anticipated fault mode $1Fj$, $2Fj$, $3Fj$ and $4Fj$, respectively (c.f. Figure 3). The observable continuous behavior in every mode (nominal or faulty) is described by constraints linking observable variables. Boolean consistency indicators (residuals) $r_i, i = 1..6$,

Nominal mode	$N1$	$N2$	$N3$	$N4$
Valve V1	*opened*	*closed*	*closed*	*opened*
Valve V2	*opened*	*opened*	*closed*	*closed*

Table 1. The system configuration in nominal modes

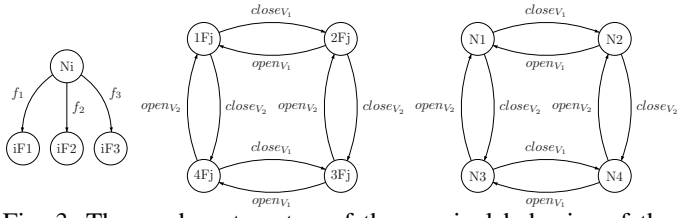

Fig. 3. The mode automaton of the nominal behavior of the three-tanks system

$N1, N2, N3, N4$	$\dot{h}_1 = 0\ (r_1), \dot{h}_2 = 0\ (r_2), \dot{h}_3 = 0\ (r_3)$
$1F1, 1F2, 1F3$	$\dot{h}_1 < 0\ (r_4), \dot{h}_2 < 0\ (r_5), \dot{h}_3 < 0\ (r_6)$
$2F1, 3F1$	$\dot{h}_1 < 0\ (r_4), \dot{h}_2 = 0\ (r_2), \dot{h}_3 = 0\ (r_3)$
$2F2, 2F3$	$\dot{h}_1 = 0\ (r_1), \dot{h}_2 < 0\ (r_5), \dot{h}_3 < 0\ (r_6)$
$3F2$	$\dot{h}_1 = 0\ (r_1), \dot{h}_2 < 0\ (r_5), \dot{h}_3 = 0\ (r_3)$
$3F3, 4F3$	$\dot{h}_1 = 0\ (r_1), \dot{h}_2 = 0\ (r_2), \dot{h}_3 < 0\ (r_6)$
$4F1, 4F2$	$\dot{h}_1 < 0\ (r_4), \dot{h}_2 < 0\ (r_5), \dot{h}_3 = 0\ (r_3)$

Table 2. Set of continuous constraints and associated residuals in each operating mode

are associated to every constraint and allow one to check the consistency between observations and system model (c.f. Table 2).
For the sake of clarity, shared constraints are considered only once in the mode signatures of the system. Given $[r_1, r_2, r_3, r_4, r_5, r_6]$ the vector of all system residuals, the mode signature is computed on-line by evaluating this vector using system observations.
Let's consider the case when any of the fault events $f1$, $f2$ or $f3$ occur in the nominal mode $N1$. The corresponding behavior automaton is shown in Figure 4. Events R^{ic}_{o1}, R'^{ic}_{o1}, R^{ic}_{o2}, R'^{ic}_{o2}, R^{ic}_{o3}, R'^{ic}_{o3}, R^{ic}_{o4} and R'^{ic}_{o4} correspond to the observable switches of the mode signature that follow the control inputs. They belong to the set of induced controllable events $\Sigma_{hyb_{ic}}$. R_{of} corresponds to the observable mode signature switch after the occurrence of any of the fault events $f1$, $f2$ or $f3$. As previously mentioned, non observable signature switches (Σ^{Sig}_{uo}) are not considered.
The diagnoser of the three-tanks system is computed from the behavior automaton. Let us focus on the part of the active diagnoser shown in Figure 5. The occurrence of the fault event $f1$, $f2$ or $f3$ is detected by the observation of the observable event R_{of}. The presence of the indeterminate cycle $[\{(\{2F2, \{F2\}), (2F3, \{F3\})\}, \{(\{21F2, \{F2\}), (21F3, \{F3\})\}, \{(\{1F2, \{F2\}), (1F3, \{F3\})\}, \{(\{12F2, \{F2\}), (12F3, \{F3\})\}]$ proves (c.f. Proposition 1) that the language of the hybrid system is not diagnosable.
The non diagnosability of the system language is due to the non diagnosability of faults $f2$ and $f3$ pointed out by the

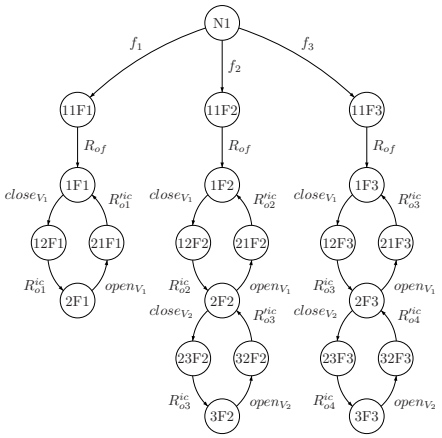

Fig. 4. Part of the behavior automaton of the three-tanks system

indeterminate cycle shown in Figure 5. However, we show that performing active diagnosis allows us to diagnose the system with certainty.

The active diagnosis consists in searching a conditional plan that permits to leave the starting uncertain state of the diagnoser and reach a certain state. These uncertain states may be crossed by indeterminate cycles (example: $\{(\{2F2,\{F2\}),(2F3,\{F3\})\}$) or not (example: $\{(1F1,\{F1\}),(1F2,\{F2\}),(1F3,\{F3\})\}$).

Given the system diagnoser, the occurrence of any fault event $f1, f2$ or $f3$ is detected by the observable events R_{of} and puts the diagnoser in the uncertain state $\{(1F1,\{F1\}),(1F2,\{F2\}),(1F3,\{F3\})\}$. From this uncertain state the active diagnosis plan is: $[close_{V_1},\ if\ R^{ic}_{o2}\ close_{V_2}\ Else\ [\ ]\ ]$.

Consequently, to perform active diagnosis, the controller sends the discrete-control-input $close_{V_1}$, if the resulting observed induced controllable event is R^{ic}_{o2} (i.e. the diagnoser state is $\{(2F2,\{F2\}),(2F3,\{F3\})\}$) then it sends the control input $close_{V_2}$ to discriminate between $F2$ and $F3$, else, the resulting observed induced controllable event is R^{ic}_{o1} (i.e. the diagnoser has reached the certain state $\{(2F1,\{F1\})\}$ and the controller does not send any more discrete control input.

Let us notice that this plan is guaranteed, because after the action $close_{V_1}$ ($close_{V_2}$), the possible resulting induced controllable events R^{ic}_{o1} and R^{ic}_{o2} (R^{ic}_{o3} and R^{ic}_{o4}) are anticipated by the plan.

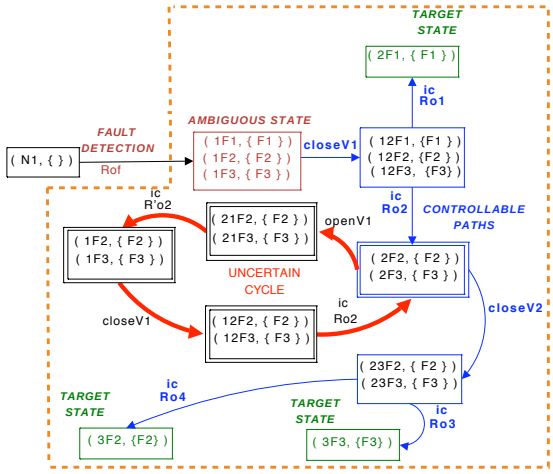

Fig. 5. Part of the active diagnoser of the three-tanks system

5. CONCLUSION

This paper deals with the problem of diagnosing hybrid systems that exhibit continuous and discrete event dynamics. The abstraction of the continuous dynamics in terms of discrete events allows one to use discrete event techniques to perform diagnosis. Based on these results, the diagnoser approach is used to perform on-line diagnosis. When the diagnoser is blocked in an ambiguous state, an active diagnosis process is needed. The concepts of controllable path, controllable induced events and active diagnoser are introduced and allow us to formulate active diagnosis as a conditional planning problem. From an ambiguous state, active diagnosis consists in defining a controllable path leading to a certain state. The choice of a control action depends on the observed response of the system after the previous action. Several problems remain, in particular, the existence of a "guaranteed" active diagnosis plan is not always achieved. Hence, when many "non guaranteed" plans are possible the system has to be able to choose the best one. Finally, the conditions for active diagnosability will be studied in future work.

REFERENCES

Abramovici, M., Strond, C., Hamilton, C., Wijesuriya, S., and Verma, V. (1999). Using roving stars for on-line testing and diagnosis of FPGAs infault-tolerant applications. In *Proceeding of the International Test Conference*, 973–982. Atlantic City, NJ (USA).

Bayoudh, M., Travé-Massuyès, L., and Olive, X. (2008a). Coupling continuous and discrete event system techniques for hybrid systems diagnosability analysis. In *Proceedings of the 18th European Conference on Artificial Intelligence ECAI*, 219–223. Patras (Greece).

Bayoudh, M., Travé-Massuyès, L., and Olive, X. (2008b). Hybrid systems diagnosis by coupling continuous and discrete event techniques. In *Proceedings of the 17th International Federation of Automatic Control, World Congress, IFAC-WC*, 7265–7270. Seoul (Korea).

Bayoudh, M., Travé-Massuyès, L., and Olive, X. (2009). Active diagnosis of hybrid systems guided by diagnosability properties. *To appear in the proceeding of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Safeprocess'09*.

Hamscher, W., Console, L., and de Kleer, J. (eds.) (1992). *Readings in model-based diagnosis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Henzinger, T. (1996). The theory of hybrid automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, 278–292. New Brunswick, New Jersey.

Jimenez, P. and Torras, C. (2000). An efficient algorithm for searching implicit and/or graphs with cycles. *Artificial Intelligence*, (124)1, 1–30.

Nicolaidis, M. and Zorian, Y. (1998). On-line testing for VLSI-A compendium of approaches. *Journal of Electronic Testing*, 12(1-2), 7–20.

Niemann, H. (2006). A setup for active fault diagnosis. *IEEE Transactions on Automatic Control*, 51(9), 1572–1578.

Ramadge, P.J. and Wonham, W.M. (1989). The control of discrete-event systems. *Proc. IEEE*, 77(1), 81–98.

Russel, S. and Norvig, P. (eds.) (2003). *Artificial Intelligence, A modern Approach, Second Edition*. Prentice Hall Series in Artificial Intelligence.

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

Sampath, M., Sengputa, R., Lafortune, S., Sinnamohideen, K., and Teneketsis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40, 1555–1575.

Struss, P. (1994). Testing for discrimination of diagnoses. In *Proceeding of the 5th International Workshop on Principles of Diagnosis DX'94*, 312–320. New Palttz (USA).

Terrell, W. (ed.) (1999). *Some Fundamental Control Theory I: Controllability, Observability, and Duality*, volume 106. Mathematical Association of America.