Timed Fault Diagnosis

Louise Travé-Massuyès and Gabriela Calderón-Espinoza

Abstract— **Time has been shown to be determinant in many diagnosis situations. A way to account for time is to enrich the fault signature matrix with time information. However this requires to anticipate the symptoms occurrence order or provide estimates for the symptom occurrence dates. In this paper, we use the time information present in the behavior model of the system and automatically retrieve this information when a symptom occurs. A conflict-based approach allows us to generate the diagnoses. For this purpose, we extend the classical DX diagnosis generation algorithm based on hitting sets to deal with time and provide an incremental version. The algorithm takes as input *time labelled conflicts* that are obtained by checking the consistency of observations against a causal behavioral model including explicit time information. Our algorithm accounts automatically for time and outputs *time labelled diagnoses* that are updated in time. It is illustrated on a two tanks example including delays in which time aspects have a strong impact on diagnosis and fault discrimination.**

Keywords Model-based Diagnosis, temporal diagnosis, fault isolation

I. INTRODUCTION

Model-based diagnosis in dynamical systems is an active research domain. The standard approaches for continuous systems, either in the diagnosis community rooted in control (FDI community) or in the diagnosis community rooted in artificial intelligence (DX community), have a static view of diagnosis and do not account for aspects referring to time for isolating or identifying faults [1]. For example, the usual FDI fault signatures rely on symptoms that are based on residuals and it is assumed that all the symptoms are available simultaneously. This is not obviously the case in many situations. The same is true for the DX logical model-based diagnosis approaches since diagnoses are often generated from symptoms and their associated conflicts [1] independently of their time of occurrence.

Although model-based diagnosis approaches based on discrete event systems naturally account for time information by using modelling formalisms like automata or Petri Nets [2], [3], [4], continuous dynamics are generally not modelled and diagnosis performs in an event-driven way. The continuous systems diagnosis community has had some proposals to deal with time [5]. [6], [7], [8] but the problem remains partially explored. As a matter of fact, eventhough temporal diagnosis is at the core of several pieces of work, there is no general ”theory” of temporal diagnosis [9]. Recently, [10] [8] clearly showed the impact of using time information on diagnosis results.

Our paper starts from these considerations and shows that information referring to time can be automatically accounted for in the standard model based logical diagnosis framework of the DX community [11], [12]. The main contribution of this paper is thus to provide an extended version of the incremental hitting sets algorithm [11], [13], which takes into account aspects referring to time and introduces timed labels into the diagnosis generation process, under the permanent fault assumption.

In our approach, we consider that information referring to time is available and is present in the physical's behavioral model, i.e. propagation delays due for instance to transportation are known and given explicitly. This information is captured in the so-called conflicts when inconsistencies between model and observations are detected. It appears in the form of timed labels associated to the conflict elements. To do so, we propose a specific conflict generation algorithm based on a causal representation of the behavioral model but other approaches, for example [5] that deals with time in an ATMS framework, could be used as well.

When fed with time labelled conflicts, the proposed algorithm for diagnosis generation is able to output *time labelled diagnoses* that are updated in time. In the case of fault exoneration assumption, it is shown that possible diagnosis candidates can be discarded on the basis of time based consistency of the timed labels associated to the same component involved in different conflicts. The algorithm is incremental in the sense that it accounts for the time at which the symptoms appear, i.e., the time at which the conflicts are raised, and it is able to check the timed based consistency of the conflict element timed labels against the conflict occurrence time.

II. A WATER TRANSPORT SYSTEM EXAMPLE

Through this paper we use the example presented in [8] which is a two reservoirs system given in Fig. 1.

It consists in continuously supplying water to two consume areas (s_1 and s_2 are the corresponding flows) from two cascaded geographically distant reservoirs (y_1 and y_2 are the water levels in the respective reservoirs). The water transport between reservoirs is modelled as an open flow channel with a pump. τ_1 between the two reservoirs and τ_2 between the pump and reservoir 1 are the transport time delays.

This work was supported by Girona University, the Spanish government through DPI2004-07167-C02-02 and SGR00296 and the Mexican government (CONACyT).

L. Travé-Massuyès is with LAA-CNRS, Av. du Colonel Roche, 31077 Toulouse cedex 4, France. e-mail:louise@laas.fr

G. Calderón-Espinoza is with the Institut d'Informtica i Aplicacions. Universitat de Girona. Campus Montilivi, E-17071 Girona, Spain. e-mail:gcalder@eia.udg.es

[1] see [1] for the relation between residuals and conflicts.

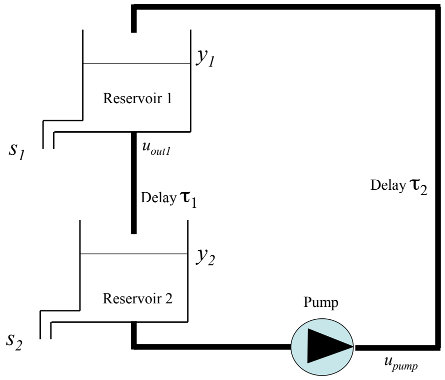

Fig. 1. Two reservoirs example

Variable u_{out1} represents the output flow from reservoir 1 and u_{pump} represents the flow through the pump.The mes index indicates measured variables. The system is modelled by the following set of discrete time equations, in which Δt is the sampled time period:

$$\begin{array}{ll}
\textit{Upper tank:} & y_1(t+\Delta t) = y_1(t) - k_1 s_1(t) \\
 & +k_2 u_{pump}(t-\tau_2) - k_3 u_{out1}(t) \qquad (1) \\
\textit{Output pipe:} & u_{out1}(t) = k\sqrt{y_1} \cong k_4 y_1(t) \qquad (2) \\
\textit{Pump:} & u_{pump}(t) = k[a(h-y_2)^2 + b(h-y_2) + c] \\
 & \cong k_5 + k_6 y_2(t) \qquad (3) \\
\textit{Lower tank:} & y_2(t+\Delta t) = y_2(t) - k_7 s_2(t) \\
 & +k_8 u_{out1}(t-\tau_1) - k_9 u_{pump}(t) \qquad (4) \\
\textit{Sensor tank1:} & y_{1mes} = y_1 \qquad (5) \\
\textit{Sensor tank2:} & y_{2mes} = y_2 \qquad (6) \\
\textit{Sensor area1:} & s_{1mes} = s_1 \qquad (7) \\
\textit{Sensor area2:} & s_{2mes} = s_2 \qquad (8)
\end{array}$$

III. Diagnosis oriented causal modelling

Causal models are proposed and shown to be suitable for diagnosis in several pieces of work. In [14][15][16], model prediction is performed along causal influences and the model causal structure is proposed as a substitute of dependency recording mechanisms. Also, the generation of analytical redundancy relations from causal models is approached in [17]. The causal model approach is applied to fault detection and isolation on a real petro-chemical process in [18].

Causal models are generally supported by an oriented graph, also called *Causal Graph*, in which nodes represent variables and edges represent influences from variable to variable. An oriented edge from variable x to variable y exists if x has an influence on y, i.e. if a perturbation on variable x affects the value of variable y. x and y are called the *cause* and the *effect* variable of the influence, respectively.

Influences not only capture the causal structure of the model [19] but also behavioral information when adequately labelled by propagation parameters or functions. For our purpose, let us assume that every influence of the causal graph is labelled by a symbol I_i standing for the influence name, a symbol C_j standing for its supporting physical component, and an integer d capturing time information. d is given with respect to a sampled time of period Δt, i.e. it is equal to an integer so that $d \times \Delta t$ is the time delay required for the effect variable to react to a change of the cause variable. When no ambiguous, d is also called the delay of the influence. Influences may also have an associated *activation condition* given by a boolean.

Three types of variables exist to model a system:

Input variables: these variables are exogenous to the system. Their values are controlled by the system's environment and assumed to be known.

Measured or Output variables: these variables are known, as provided by a sensoring device.

Unmeasured variables: these variables are internal to the model and their values are not known.

When fed with input variable values, the causal model can be used to predict the value for other variables by propagating them through the influence network. A fault detection mechanism can be based on checking the values predicted for output variables against their observed values. If the predicted value is not consistent with the observed value, then an alarm is activated. A discrepancy for variable y indicates a misbehavior and is noted with the predicate $MISB(y)$.

Causal ordering methods issued from the Qualitative Reasoning community can be advantageously applied to derive automatically the causal structure associated to a set of relations [19], [20]. As an example, consider the causal structure of the two tanks system example given in Fig. 2 obtained from (1-8). In this figure, dx stands for the variable $x(t+\Delta t)$; every influence I_i, $i = 1, ..., 14$, is associated with its underlying physical component: $C1$ corresponds to the level sensor of Tank 1 modelled by (5); $C2$ to the level sensor of Tank 2 modelled by (6); $C3$ to the flow sensor of consume area 1 that is modelled by (7); $C4$ to the flow sensor of consume area 2 that is modelled by (8); $C5$ to the output pipe; $C6$ to the pump; $C7$ to the upper tank and $C8$ to the lower tank. One should notice that one component may be associated to several influences like for C7 and C8. Equations which involve variables at different time points support dynamic influences, which have naturally a non zero delay. For example, the influence between a cause variable x and an effect variable $dx = x(t+\Delta t)$ has a natural delay of 1 sampling period.

IV. Conflicts and diagnoses

Let us call *CSD* the Causal System Description as presented in section III, *COMP* the set of physical components composing the physical system, and *OBS* the set of observations at some time point. When one or several variables misbehave, the diagnosis system must derive all sets of faulty components of *COMP* that may explain the fact. The influences that may be at the origin of the misbehavior of a variable V are those related to the edges belonging

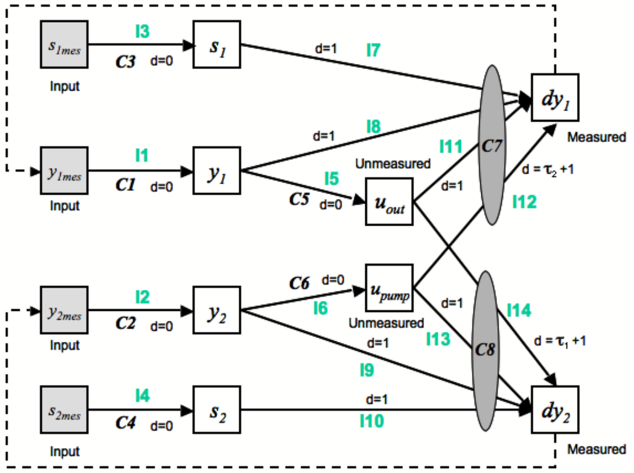

Fig. 2. Causal structure for the two tanks system

to the paths going from the measured nodes to the node representing V, also called *ascending influences*. The set of such influences (or, equivalently, set of corresponding components) is a *conflict* set, in the sense of Reiter [11]. Conflict sets are sets of components which cannot behave normally altogether according to the observations. A minimal conflict is a conflict that does not strictly include (in the sense of set inclusion) any conflict. [11] proved that minimal diagnoses can be computed from minimal conflicts. In the example of Fig. 2, if we have MISB(dy_1) then there is one (minimal) conflict set $\{C1, C2, C3, C5, C6, C7\}$.

Proposition 1 (Reiter [11]) *$\Delta \subseteq COMP$ is a (minimal) diagnostic for* (CSD, COMP, OBS) *if and only if Δ is a (minimal) hitting set for the collection of (minimal) conflict sets of* (SD, COMP, OBS).

A hitting set of a collection of sets is a set intersecting every set of this collection. An incremental algorithm to generate all the minimal hitting sets based on a set of conflicts was originally proposed by [11], then corrected by [21]. This algorithm gives a means to compute diagnoses incrementally, under the permanent fault assumption.

The diagnosis algorithm builds a Hitting-Set tree (HS-tree) in which all the nodes but leaves are labelled by a conflict set. For each element s in the conflict label of node n, an edge labelled s joins n to a successor node. $H(n)$ is defined as the set of edge labels on the path from n to the root node. The HS-tree is built by considering every conflict in arbitrary order. Every new conflict is compared to every leaf of the HS-tree, and some new leaves are built if necessary. The resulting HS-tree is pruned for redundant or subsumed leaves before the next conflict is considered. At the end of the diagnosis procedure, the minimal hitting sets, and hence the minimal diagnoses that explain the system's misbehaviors, are given by the set of edge labels $H(l)$ associated to the open leaves l of the HS-tree.

We use the algorithm version by [13] which is more efficient because it uses less comparisons at each step. Indeed, [13] showed that to prune the tree when a new conflict set S has been compared to every leaf, it is sufficient to prune the new leaves. Moreover, each new leaf having the label $s \in S$ on its last edge has only to be compared to the old ones having the label s and no other label of S on its path to the root. This algorithm is given in section VI-C; bolded lines marked with a star should be ignored at this stage. For every new conflict S and every element s of the conflict, the algorithm builds two lists, *newleaves[s]* and *oldleaves[s]*, which are then compared. A new leaf l is closed if $H(n)$ contains $H(l)$ for some old leaf.

V. Accounting for time in conflict generation

Information referring to time is explicitly represented in *CSD* by the delays associated to influences or to their corresponding components (cf. Fig. 2). From now on and given the equivalence, we work indifferently with components or with influences. Faults are assumed to be permanent, although their effects (the associated symptoms) may be transient. Delays are assumed to be an order of magnitude above the slowest detection time of the consistency check tests.

A timed label called *failure time* is introduced to indicate how long, at the shortest, a component, say C, has been failing with reference to the current instant. In other words, given the current time t, the failure time indicates that the component C must have been faulty since at least time $(t - failuretime)$ and this is noted $C_{failuretime}$. The elements of the conflicts that are generated when a variable misbehaves in CSD are hence each labelled by a failure time. These conflicts are called *time labelled conflicts*.

[5] defines the *temporal extend* TE(α) of a proposition α as the set $\{t_i/\alpha \text{ holds at } t_i\}$. Let us denote the fact that C is faulty by the predicate AB(C). Then, $C_{failuretime}$ is a short way to represent TE$(AB(\text{C})) \supseteq [t - failuretime, t]$.

The following considerations are taken into account for reasoning about time from the causal model CSD, i.e., for assigning the failure times to the components when generating conflicts (cf. Fig. 2) :

1) The influence's delay represents the time needed by an effect variable to react to a variation of the cause variable. For example, a deviation on the variable u_{pump} takes $\tau_2 + 1$ periods of time Δt before it propagates to the variation of dy_1.
2) The effect of a fault on a component associated to an influence propagates instantaneously to the effect variable, regardless of its related delay. For example a fault on $C1$ propagates instantaneously to the effect variable y_1.
3) The occurrence of a fault on a component associated to an influence may account for a downstream misbehaving variable at time *t* if and only if the influence was active at time $t - AccuDelay$, where *AccuDelay* is equal to the sum of the delays on the path going from the directly influenced variable to the misbehaving variable, and the other influences along the path were active at times $t-$ their respective accudelays. In this case the whole path is said to be *active*. For example a fault on $C2$ can account for a misbehavior of dy_1 if

and only if influences $I2$ and $I6$ were both active at time $t-(\tau_2+1)$ and $I12$ was active at time t.

Given the above, the following proposition makes explicit how to determine the failure time associated to a component in a conflict.

Proposition 2 (Failure Time of a Conflict Element) *Consider a CSD and a given component C associated to an influence I located upstream from a variable y misbehaving at time t. Assuming that there are n active paths from the effect variable of I to the misbehaving variable y, then C is an element of the conflict associated to MISB(y) and its failure time is given by the* minimum *AccuDelay among the n active paths.*

This result is quite obvious. When a component has two or more ways to reach the same variable, the *minimum*AccuDelay is enough to explain that the variable has been influenced by the faulty component. Indeed, if a fault on component C has an effect on a variable x via a "short" channel with delay τ_1 and a "long" channel with delay τ_2, then C_{τ_1} implies C_{τ_2}, i.e. C faulty since at least time $t-\tau_1$ implies C faulty since at least time $t-\tau_2$.

Conflict generation is carried out for each misbehaving variables in the CSD as explained in section IV. The pseudo-algorithm for generating time labelled conflicts is the following :

```
FOR each conflict
    FOR each component in the conflict
    Calculate AccuDelay for each path
    going from the variable directly downstream the
    component to the misbehaving variable
      IF the set of active paths is not empty
      Select the minimum AccuDelay among the
      active paths
      Label the component with the minimum
      AccuDelay
      End IF
      ELSE
    Remove the component from the conflict
    END for each component
END for each conflict
```

VI. Timed Diagnosis generation

The diagnosis generation algorithm is devised to run synchronously with the sampled time, at that instants when new symptoms occur, i.e. conflict sets are generated. In this section, we extend the incremental diagnosis generation algorithm [11][21][13] to deal with aspects referring to time in two cases: simultaneous symptom occurrence and symptoms occurring in time. The algorithm takes as input time labelled conflicts and outputs *time labelled diagnoses.*

A. Management of simultaneous symptoms

This section analyzes the case of multiple simultaneous symptoms occurrence. The following principle, called *Maximum Failure Time Principle* is used to determine the diagnosis element failure times from the conflict element failure times.

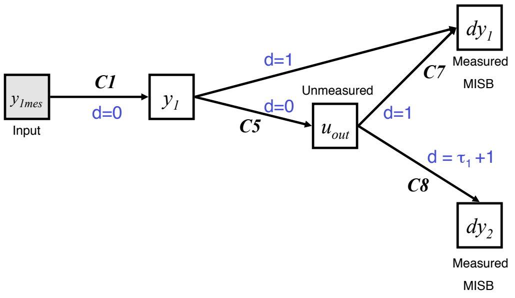

Fig. 3. One component in two conflict sets

Maximum Failure Time Principle Let us consider that one component is involved in the misbehavior of two, or more, variables (see Fig. 3 in which it is the case for $C1$ [2]), i.e. it appears in several conflicts with different failure times. Then, in the generated diagnoses, this component is labelled with the *maximum* failure time of the component in the conflicts.

Labelled with the maximum failure time, the component simultaneously explains the misbehavior of all the misbehaving variables, i.e., simultaneously covers all conflicts. This is a direct consequence of the "failure time" semantics that means that the labelled component must have been faulty since at least time ($t-failuretime$), t being the current time.

Let's take the two tanks example and its causal structure of Fig. 2. Consider two simultaneous misbehaving variables MISB(dy_1) and MISB(dy_2) at time t and the two corresponding conflicts:

$Conf(dy_1)=\{C1_1\ ,\ C2_{\tau_2+1}\ ,\ C3_1\ ,\ C5_1\ ,\ C6_{\tau_2+1}\ ,\ C7_0\}$
$Conf(dy_2)=\{C1_{\tau_1+1}\ ,\ C2_1\ ,\ C4_1\ ,\ C5_{\tau_1+1}\ ,\ C6_1\ ,\ C8_0\}$

Component $C1$ has a failure time of 1 in the first conflict and of τ_1+1 in the second one. Then, the single component time labelled diagnosis based on $C1$ is $\{C1_{\tau_1+1}\}$ because $max[1,\tau_1+1]=\tau_1+1$. The hitting-set tree corresponding to the management of these two conflicts by our algorithm is shown in Fig. 4. The diagnoses $D1$ to $D8$ given below result from the open leaves $ol1$ to $ol8$ with the following indexes $\{ol_1=1$, $ol2=2$, $ol3=4$, $ol4=5$, $ol5=9$, $ol6=12$, $ol7=15$, $ol8=18\}$ in the hitting-set tree. Notice that the *Maximum Failure Time Principle* has been used for all the single component diagnoses $D1$ to $D4$.

$D1=\{C1_{\tau_1+1}\}$
$D2=\{C2_{\tau_2+1}\}$
$D3=\{C5_{\tau_1+1}\}$
$D4=\{C6_{\tau_2+1}\}$
$D5=\{C3_1\ ,\ C4_1\}$
$D6=\{C3_1\ ,\ C8_0\}$
$D7=\{C7_0\ ,\ C4_1\}$
$D8=\{C7_0\ ,\ C8_0\}$

B. Management of symptoms occurring in time

When misbehaving variables and hence the corresponding conflicts appear at different times, failure times associated to

[2] Fig. 3 outlines part of the causal structure for the two tanks example.

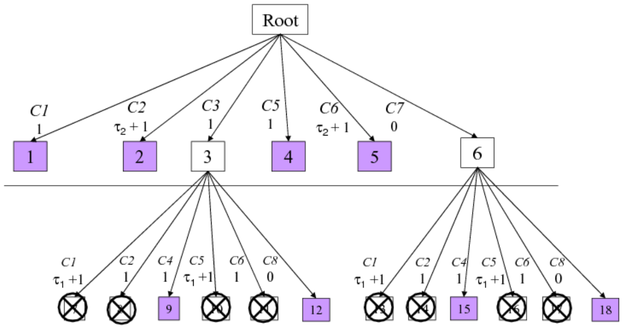

Fig. 4. Timed diagnosis with simultaneous symptom occurrence

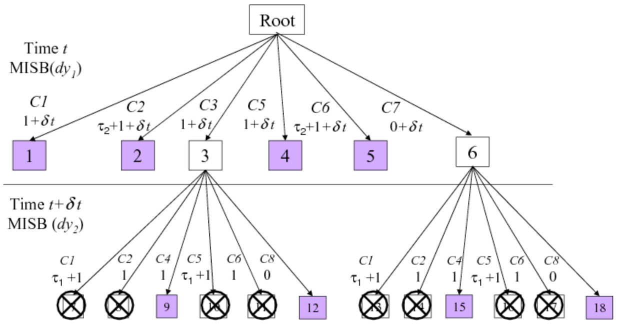

Fig. 6. Diagnoses at different times: MISB(dy_1) at time t and MISB(dy_2) at time $t + \delta t$, ($\delta t = 1$, $\tau_1 = 3$ and $\tau_2 = 5$)

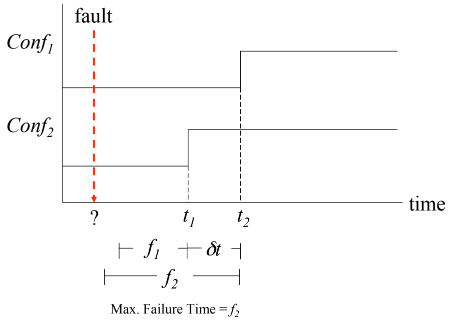

Fig. 5. Maximum failure time principle

TABLE I

DIAGNOSES WITH FAILURE TIMES UPDATED

Diagnoses at time t	Diagnoses at time $t + \delta t$
$D1 = \{C1_1\}$	$D1 = \{C1_{\tau_1+1}\}$
$D2 = \{C2_{\tau_2+1}\}$	$D2 = \{C2_{\tau_2+1+\delta t}\}$
$D3 = \{C3_1\}$	$D3 = \{C5_{\tau_1+1}\}$
$D4 = \{C5_1\}$	$D4 = \{C6_{\tau_2+1+\delta t}\}$
$D5 = \{C6_{\tau_2+1}\}$	$D5 = \{C3_{1+\delta t}, C4_1\}$
$D6 = \{C7_0\}$	$D6 = \{C3_{1+\delta t}, C8_0\}$
	$D7 = \{C7_{0+\delta t}, C4_1\}$
	$D8 = \{C7_{0+\delta t}, C8_0\}$

the components in diagnoses must be updated to account for the interval of time in between conflict occurrence. This is referred to as the *Updated Failure Time Principle*.

Updated Failure Time Principle Let us assume that Ci_{f_1} is involved in a conflict at date t and that Cj_{f_2} is involved in a conflict at date $t + \delta t$, then the generated diagnosis must update the failure time of Ci to account for δt. The generated diagnosis candidate is hence $\{Ci_{f_1+\delta t}, Cj_{f_2}\}$.

After updating failure time(s), the Maximum Failure Time Principle applies like in the case of simultaneous conflicts. This is represented in Fig. 5 where f_1 and f_2 indicate two failure times for the same component C in two different conflicts $Conf_1$ and $Conf_2$ rised at t_1 and t_2. At time $t2$, C must be labelled with the maximum failure time f_2.

Note that the case of simultaneous conflicts can be viewed as a special case of conflicts occurring in time where $\delta t = 0$.

Fig. 6 illustrates the procedure. The results depend on the values of δt, τ_1 and τ_2, which determine the final values of the updated temporal labels. Let us consider $\delta t < \tau_1 < \tau_2$ with numeric values $\delta t = 1$, $\tau_1 = 3$, and $\tau_2 = 5$, then the generated diagnoses with failure times updated according to the *Updated Failure Time Principle* and *Maximum Failure Time Principle* are given in Table I.

Note that the diagnoses at time $t+\delta t$ are the same, in terms of components, as those that would be obtained if the two conflicts were simultaneous but components have different temporal labels.

In particular, the order of symptoms changes the temporal labels of the diagnoses elements. When conflicts appear at different times, the time of occurrence of conflicts is naturally taken into account by the *Updated Failure Time Principle*. It is easy to show that different temporal labels are obtained by the time of occurrence of the set of symptoms.

This is illustrated by inverting the order of the two symptoms in the previous example: MISB(dy_2) happens at time t and MISB(dy_1) at time $t + \delta t$ (see Fig. 7). The open leaves of the hitting set tree determine the diagnoses given in Table II, which differ from those obtained when MISB(dy_1) happens before MISB(dy_2) by the temporal label.

C. Timed diagnosis generation algorithm

The incremental hitting sets generation algorithm presented in section IV has been revised to manage timed labels, applying the Maximum Failure Time Principle and the Updated Failure Time Principle outlined in sections VI-A

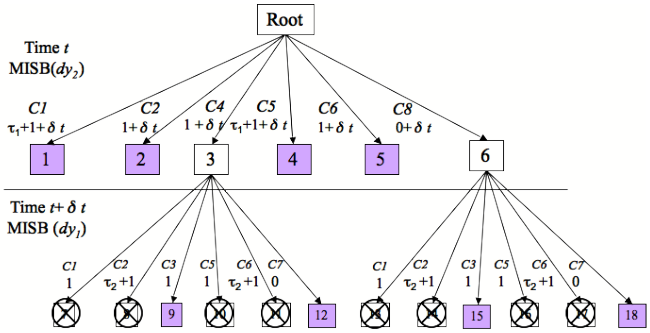

Fig. 7. Diagnoses at different times and inverted order: MISB(dy_1) at time t and MISB(dy_2) at time $t + \delta t$, ($\delta t = 1$, $\tau_1 = 3$ and $\tau_2 = 5$)

and VI-B. It is described in the following pseudo-algorithm in which the variants from the original Levy's algorithm are indicated by bolded and star line numbers.

The algorithm begins with a tree T consisting of an unlabelled root. $H(n)$ represents the set of edge labels on the path from the root node to the node n.

```
1. For Each time labelled conflict set S Do
2*    Determine $\delta t$ and apply the Updated Failure Time
      Principle to all the edge labels of the current HS-tree
3.    For Each (time labelled conflict) element s in S Do
4.       Initialize the lists with new-leaves[s]:=$\{\}$;
         old-leaves[s]:=$\{\}$;
5.    End For Each (time labelled conflict) s in S
6.    For Each leaf l of T Do

* New leaves creation *
7.       If H(l) $\cap$ S = $\{\}$ Then
8.          For Each s in S Do
9.             Add to l an edge labelled with s and a successor
               node $n_s$;
10.            Add the pair ($n_s$, H(l) $\cup\{s\}$) to new-leaves[s];)
11.         End For Each s in S

* Old leaves creation *
12.      Else If H(l) $\cap S = \{s\}$ Then /* s is a singleton*/
13.         Add the pair (l,H(l)) to old-leaves[s];
14*         Apply the Maximum Failure Time Principle and
            label the conflict element $\{s\}$ with the maximum
            failure time
15.      End if

16.   End For Each leaf l of T

* Closing leaves *
17.   For Each s in S Do
18.      For Each leaf n of new-leaves[s] Do
19.         If H(n) contains H(l) for some leaf l of
            old-leaves[s] Then close the branch in n;
20.         End if
21.      End For Each leaf n of new-leaves[s]
22.   End For s in S

23. End For Each conflict set S
```

TABLE II

DIAGNOSES WITH FAILURE TIMES UPDATED AND DIFFERENT ORDER OF SYMPTOMS

Diagnoses at time t	Diagnoses at time $t+\delta t$
$D1 = \{C1_{\tau_1+1}\}$	$D1 = \{C1_{\tau_1+1+\delta t}\}$
$D2 = \{C2_1\}$	$D2 = \{C2_{\tau_2+1}\}$
$D3 = \{C4_1\}$	$D3 = \{C5_{\tau_1+1+\delta t}\}$
$D4 = \{C5_{\tau_1+1}\}$	$D4 = \{C6_{\tau_2+1}\}$
$D5 = \{C6_1\}$	$D5 = \{C4_{1+\delta t}, C3_1\}$
$D6 = \{C8_0\}$	$D6 = \{C4_{1+\delta t}, C7_0\}$
	$D7 = \{C8_{0+\delta t}, C3_1\}$
	$D8 = \{C8_{0+\delta t}, C7_0\}$

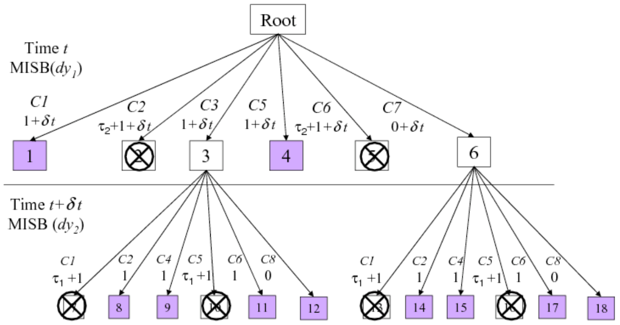

Fig. 8. Diagnoses at different times and exoneration: MISB(dy_1) at time t, MISB(dy_2) at time $t+\delta t$, ($\delta t = 3$, $\tau_1 = 3$ and $\tau_2 = 5$)

Timed Minimal Hitting Sets Algorithm

VII. FAULT EXONERATION ASSUMPTION CASE

In the temporal framework proposed in section VI, the semantics of the failure time are quite weak due to the no exoneration fault assumption. Indeed, a fault may be present without manifesting at the level of misbehaving variables. Hence, the failure time of a component C acts just as a memory of when the first conflict including C occurred. Now, when the exoneration assumption is applicable, this assumption can be advantageously used to strengthen the failure time semantics. In this case, the failure time indicates *exactly* the time that the component has been failing. One can consequently perform a time based consistency check that leads to discard time inconsistent diagnoses, producing less ambiguous diagnoses.

In the diagnosis generation algorithm, this means that some leaves can be closed because of time based inconsistency (see scenario of Fig. 8 explained below).

The following principle, called *Minimal Required Time Principle* is used to check time consistency. It is based on the fact that the failure times of a common component involved in two consecutive conflicts must account for the time interval elapsed between the symptoms occurrence.

Minimum Required Time Principle Without loss of generality, let us assume two misbehaving variables appearing at times t and $t+\delta t$ and a component C involved in the two corresponding conflicts with failure times f_1 and f_2, respectively. Then, C leads to a time consistent diagnosis if and only if:

$$f_2 - f_1 = \delta t \tag{9}$$

The time consistency check may as well be performed after applying the Updated Failure Time Principle. Let us define the updated label $f_1^{new} = f_1 + \delta t$, then the condition in (9) equivalently expresses as:

$$f_2 = f_1^{new} \tag{10}$$

TABLE III

DIAGNOSES WITH EXONERATION IN TIME

Diagnoses at time t	Diagnoses at time $t + \delta t$
$D1 = \{C1_1\}$	$D1 = \{C1_{\tau_1+1}\}$
$D2 = \{C2_{\tau_2+1}\}$	$D2 = \{C5_{\tau_1+1}\}$
$D3 = \{C3_1\}$	$D3 = \{C3_{1+\delta t}, C2_1\}$
$D4 = \{C5_1\}$	$D4 = \{C3_{1+\delta t}, C4_1\}$
$D5 = \{C6_{\tau_2+1}\}$	$D5 = \{C3_{1+\delta t}, C6_1\}$
$D6 = \{C7_0\}$	$D6 = \{C3_{1+\delta t}, C8_0\}$
	$D7 = \{C7_{0+\delta t}, C2_1\}$
	$D8 = \{C7_{0+\delta t}, C4_1\}$
	$D9 = \{C7_{0+\delta t}, C6_1\}$
	$D10 = \{C7_{0+\delta t}, C8_0\}$

Note that in this case, the Maximal Failure Time Principle is useless because time consistency implies that a given component has necessarily equal time labels.

The algorithm for the exoneration case is obtained as a variant of the algorithm in the no exoneration case by replacing the *Old leaves creation* set of lines by the following:

```
* Old leaves creation *
12.     Else If H(l) ∩S = {s} Then /*s is a singleton*/
 13*        If the Minimal Required Time Principle
            is satisfied for the timed labels of s
14.         Add the pair (l,H(l)) to old-leaves[s];
15.         Else close the branch in l
16.     End If

17.     End If

18.   End For Each leaf l of T
```

Applying (10) in the diagnosis scenario of Fig. 8 in which $\delta t = 3$, $\tau_1 = 3$ and $\tau_2 = 5$, it happens that component $C1$ fulfills the requirements to be a diagnosis: $f_1^{new} = 1+\delta t = 4$ and $f_2 = \tau_1 + 1 = 4$, hence $f_2 = f_1^{new}$. The same happens with $C5$. On the other hand, $C2$ and $C6$ are not considered as single diagnoses because of time inconsistency: $f_1^{new} = \tau_2 + 1 + \delta t = 9$ and $f_2 = 1$, which does not satisfy (10). On the HS-tree, leaf 2 corresponding to $C2$ as single component diagnosis candidate has been closed, the same is true for leaf 5 corresponding to C_6. The whole set of generated time labelled diagnoses is given in Table III.

Considering the case of inverted order symptoms, i.e. MISB(dy_2) happens at time t and MISB(dy_1) happens at time $t + \delta t$, with the same numeric values for δt, τ_1, and τ_2, (10) eliminates all single component diagnoses, as shown in Fig. 9 and in Table IV.

A. Fault exoneration relaxed case

As a relaxated case, one may be interested in distinguishing diagnoses only based on the order of occurrence of symptoms, i.e. checking temporal consistency but not timed consistency. Let us notice that in this case the condition of (9) becomes:

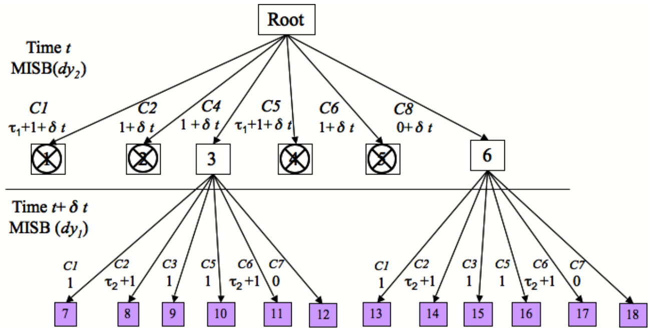

Fig. 9. Diagnoses at different times and exoneration: MISB(dy_2) at time t, MISB(dy_1) at time $t + \delta t$, ($\delta t = 3$, $\tau_1 = 3$ and $\tau_2 = 5$)

TABLE IV

DIAGNOSES WITH EXONERATION IN TIME AND DIFFERENT ORDER OF SYMPTOMS

Diagnoses at time t	Diagnoses at time $t + \delta t$
$D1 = \{C1_{\tau_1+1}\}$	$D1 = \{C4_{1+\delta t}, C1_1\}$
$D2 = \{C2_1\}$	$D2 = \{C4_{1+\delta t}, C2_{\tau_2+1}\}$
$D3 = \{C4_1\}$	$D3 = \{C4_{1+\delta t}, C3_1\}$
$D4 = \{C5_{\tau_1+1}\}$	$D4 = \{C4_{1+\delta t}, C5_1\}$
$D5 = \{C6_1\}$	$D5 = \{C4_{1+\delta t}, C6_{\tau_2+1}\}$
$D6 = \{C8_0\}$	$D6 = \{C4_{1+\delta t}, C7_0\}$
	$D7 = \{C8_{0+\delta t}, C1_1\}$
	$D8 = \{C8_{0+\delta t}, C2_{\tau_2+1}\}$
	$D9 = \{C8_{0+\delta t}, C3_1\}$
	$D10 = \{C8_{0+\delta t}, C5_1\}$
	$D11 = \{C8_{0+\delta t}, C6_{\tau_2+1}\}$
	$D12 = \{C8_{0+\delta t}, C7_0\}$

$$f_2 - f_1 > 0 \tag{11}$$

or equivalently:

$$f_2 > f_1^{new} - \delta t \tag{12}$$

In this case, the Maximum Failure Time Principle must be applied because the time consistency condition does not guarantee that the time labels of the same component are the same.

Diagnoses in the case of Fig. 8 remain the same. However, in the case of Fig. 9, C_2 and C_6 are now eligible as single component candidates, as shown in the list of diagnoses at time $t + \delta t$ in Table V.

B. Related work

The objectives of [8] are naturally close to ours but [8] performs along a pure FDI approach whereas our work is stated within the DX framework. To be more explicit, [8] generates diagnoses by comparing the observed signature to theoretical fault signatures forming the columns of the fault signature matrix. This has a number of important implications as explained in [1], in particular fault exoneration is implicitly assumed. Our framework is hence more general and our results are consistent with their results

TABLE V

DIAGNOSES WITH FAULT EXONERATION RELAXED CASE

Diagnoses at time t
$D1 = \{C2_{\tau_2+1}\}$
$D2 = \{C6_{\tau_2+1}\}$
$D3 = \{C4_{1+\delta t}, C1_1\}$
$D4 = \{C4_{1+\delta t}, C3_1\}$
$D5 = \{C4_{1+\delta t}, C5_1\}$
$D6 = \{C4_{1+\delta t}, C7_0\}$
$D7 = \{C8_{0+\delta t}, C1_1\}$
$D8 = \{C8_{0+\delta t}, C3_1\}$
$D9 = \{C8_{0+\delta t}, C5_1\}$
$D10 = \{C8_{0+\delta t}, C7_0\}$

when assuming fault exoneration, with or without relaxated time consistency check. Another difference comes from the required source information about time. Whereas [8] needs to predict by hand the symptoms occurrence order or provide estimates for the symptom occurrence dates, our method uses the time information already present in the behavior model of the system and derives the former automatically.

The paper [5] deals with time propagation, in particular across ATMS labels, to permit prediction sharing across time. The output of this work provides time labelled conflicts, arising from the inconsistencies. Such conflicts could be used as an input for our method. However [5] does not consider the problem of generating time labelled diagnosis from such conflicts.

The methods proposed by the diagnosis community dealing with discrete event (DE) systems naturally account for temporal aspects by using automata or Petri nets models [2], [3], [4]. The main difference is that they are generally based on fault models, i.e. the DE model explicitly represents the sequence of events (symptoms) that are expected to occur after the occurrence of a fault. On the contrary, our approach is rooted along the consistency based diagnosis approach. In this respect, the two approaches are complementary and could be considered for integration.

VIII. CONCLUSION

We have extended the diagnosis generation algorithm based on hitting sets to deal with temporal aspects and provided an incremental diagnosis which outputs time labelled diagnoses. The timed labels associated to the components of the diagnosis candidates indicate how long ago the fault must have occurred according to the time it is detected. One advantage of this algorithm compared with other approaches like [8], [6] is that it directly uses the source temporal information included in the model of the physical system. The main feature of the approach is that it is formulated along the DX model based diagnosis framework: it introduces the temporal information necessary to diagnose a dynamical system into the widely used hitting sets generation algorithm that was originally devised for static diagnosis. This work allows us to foresee interesting perspectives for bridging with the dynamic diagnosis approaches used in the discrete event systems community.

REFERENCES

[1] M. Cordier, P. Dague, F. Lvy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès, “Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives,” *IEEE Transactions on Control Systems Technology*, vol. 34, no. 5, pp. 2163–2177, 2004.

[2] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, “Failure diagnosis using discrete-event models,” *IEEE Transactions on systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 4, no. 2, pp. 105–124, 1996.

[3] M. Cordier and C. Dousson, “Alarm driven monitoring based on chronicles,” in *IFAC SafeProcess*, Budapest, Hungary, 2000.

[4] Y. Pencolé and M.-O. Cordier, “A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks,” *Artificial Intelligence*, vol. 164, pp. 121–170, May 2005.

[5] O. Dressler and H. Freitag, “Prediction sharing across time and contexts,” in *AAAI94*, Seattle WA, USA, 1994.

[6] C. Combastel, S. Gentil, and J. Rognon, “Toward a better integration of residual generation and diagnostic decision,” in *IFAC SafeProcess*, Washington DC, USA, 2003.

[7] S. Gentil, J. Montmain, and C. Combastel, “Combining FDI and AI approaches within causal-model-based diagnosis,” *IEEE Transactions on systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 34, no. 5, pp. 2207–2221, 2004.

[8] V. Puig, J. Quevedo, T. Escobet, and B. Pulido, “On the integration of fault detection and isolation in model based fault diagnosis,” in *15th International Workshop on Principles of Diagnosis*, Carcassonne, France, 2004.

[9] V. Brusoni, L. Console, P. Terenziani, and D. Theiseder-Dupre, “A spectrum of definitions for temporal model-based diagnosis,” *Artificial Intelligence*, vol. 102, no. 1, pp. 39–79, 1998.

[10] L. Travé-Massuyès, M.-O. Cordier, and X. Pucel, “Comparing diagnosability in continuous systems and discrete events systems,” in *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS?06*, Beijing, P.R. China, Also in *17th International Workshop on Principles of Diagnosis DX’06*, Aranda de Duero (Spain), June 26-28, 2006, 2006.

[11] R. Reiter, “A theory of diagnosis from first principles,” *Artificial Intelligence*, vol. 32, pp. 57–95, 1987.

[12] W. Hamscher, L. Console, and J. d. Kleer, Eds., *Readings in model-based diagnosis*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992.

[13] F. Levy, “Reason maintenance systems and default theories,” Universit de Paris Nord. Internal report L.I.P.N., http://www-lipn.univ-paris13.fr/ levy/Publications/RMSaDT.pdf, 31p., Tech. Rep., 1991.

[14] G. Biswas and E. Manders, “Integrated systems health management to achieve autonomy in complex systems,” in *6th Symposium on Fault Detection, Supervision and Safety for Technical Processes*, Beijing, PPR China, 2006.

[15] E. Manders, G. Biswas, R. J., M. N., W. J., and A. S., “A model integrated computing tool-suite for fault-adaptive control,” in *15th Intl. Workshop on Principles of Diagnosis*, Carcassonne, France, 2004.

[16] L. Travé-Massuyès, T. Escobet, R. Pons, and S. Tornil, “The Ca-En diagnosis system and its automatic modelling method,” *Computacin y Sistemas Journal*, vol. 5, no. 2, pp. 128–143, 2001.

[17] J. Flaus and S. Gentil, “Hybrid system automatic modeling for diagnostic purposes,” in *IFAC SafeProcess*, Washington DC USA, 2003.

[18] B. Heim, S. Gentil, B. Celse, S. Cauvin, and L. Travé-Massuyès, “FCC diagnosis using several causal and knowledge based models,” in *IFAC SafeProcess*, Washington DC USA, 2003.

[19] L. Travé-Massuyès and R. Pons, “Causal ordering for multiple mode systems,” in *11th Int. workshop on Qualitative Reasoning about physical systems QR’97*, Cortona, Italy, 1997.

[20] Y. Iwasaki and H. Simon, “Causality in device behavior,” *Artificial Intelligence*, vol. 29, no. 1, pp. 3–32, 1986.

[21] R. Greiner, B. Smith, and W. Wilkerson, “A correction to the algorithm in reiter’s theory of diagnosis,” *Artificial Intelligence*, vol. 41, no. 1, pp. 79–88, 1989.