Iterative hybrid causal model based diagnosis: Application to automotive embedded functions

Renaud Pons, Audine Subias, Louise Travé-Massuyès

Iterative Hybrid Causal Model Based Diagnosis: Application to Automotive Embedded Functions

R. Pons[a,c,*], A. Subias[a,b], L. Travé-Massuyès[a,c]

[a] *CNRS, LAAS, 7, avenue du Colonel Roche, F-31400 Toulouse, France*
[b] *Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France*
[c] *Univ de Toulouse, LAAS, F-31400 Toulouse, France*

Abstract

This paper addresses off-line diagnosis of embedded functions, such as that made in workshops by the technicians. The diagnosis problem expresses as the determination of a proper sequence of tests and measures at available control points, which would lead to greedily localize the fault quickly and at the lowest cost. Whereas anticipated discrete faults can be properly addressed by fault dictionary methods based on simulation, a consistency based method designed for hybrid systems is proposed to address parametric faults and non anticipated faults. This method uses those same inputs as the fault dictionary method and the only additional information is the structure of the reference models in the form of a causal graph and the interpretation of the simulation results into qualitative values and events. The consistency based diagnosis method is combined with a test selection procedure to produce an original iterative diagnosis method for hybrid systems that reduces diagnosis ambiguity at each iteration.

The method is illustrated in the automotive domain with a real case

[*] corresponding author
Email addresses: rpons@laas.fr (R. Pons), subias@laas.fr (A. Subias), louise@laas.fr (L. Travé-Massuyès)

study consisting in the electronic function commanding the rear windscreen wiper of a car.
Keywords: Consistency Based Diagnosis, Hybrid Systems, Causal Graphs, Test Selection, Automotive Embedded Functions

---

1. Introduction

Today embedded systems are found everywhere and form an integral part of the design of contemporary artefacts, in interaction with hardware components spanning multi-domain technologies, such as mechanical, hydraulic, etc. The intimate coupling of software and hardware capacities allows engineers to design systems for responding at the nearest of every anticipated situation. The resulting systems exhibit complex patterns of behavior and numerous nominal modes of operation to achieve high adaptability. Hardware components are inherently continuous but control is generally performed by supervisory controllers, also known as Electronic Control Units (ECUs), that impose discrete switching between the modes of operation (McIlraith et al., 2000a). Diagnosing and trouble-shooting such systems is a tedious task, which must not only account for the structural interconnection of components but also for the different configurations underlying behavioral modes.

In this paper, we are interested in off-line diagnosis, such as that made in workshops by the technicians. In practice, embedded systems are diagnosed from diagnosis trees built beforehand, often manually. They allow the technicians to find the faulty component(s) by performing a guided sequence of measurements. The diagnosis problem expresses as the determination of a proper sequence of tests and measures at available control

points, which would lead to greedily localize the fault quickly and at the lowest cost. This problem is also known as the Test Sequencing Problem (Pattipati and Dontamsetty, 1992a; Pattipati and Alexandridis, 1990) or Test Prioritization Problem in the software testing and debugging community (Li et al., 2007). It is generally approached by anticipating all the possible test results to generate a complete optimal diagnosis tree. Another option however is to iterate a diagnosis session starting with a few measurements and a test selection procedure proposing the next best test (Struss, 1994). This is the approach adopted in this paper.

In the automotive field, the use of electronic systems to control several functions, like fuel injection or ABS, has considerably increased during the last decade. These electronic systems are composed of voltage supplies, sensors and actuators linked to ECU by a wire harness. ECUs are equipped with an auto-diagnosis function delivering fault codes that reliably detect the failing electric circuits which are connected to this ECU, although they are unable to localize precisely the faulty components.

Diagnosis starts with a set of preliminary symptoms gathered by the garage mechanic. In addition to fault codes, these are client symptoms and other preliminary garage mechanic observations. Then, fault isolation is performed by successively applying the test that brings the best discrimination among the diagnostic hypotheses generated with the preliminary symptoms. One test is defined by the variable to be sensed and the configuration in which the system must operate. Previous works have proposed solutions to diagnose electric circuits (Faure et al., 1999; Faure, 2001; Olive et al., 2003; Price et al., 1996; Sachenbacher and Struss, 2001), among which only few of them fully account for the hybrid nature of the systems (Travé-massuyès et al., 2013). Most methods are based on a dictionary of fault

signatures supporting heuristic optimization techniques or the computation of the expected quantity of information for the tests. In (Ressencourt et al., 2006), hybrid system simulation techniques, based on the Modelica language, are used to build the dictionary of fault signatures from faulty models. Hierarchical multi-model strategies are then applied to structure the search space by articulating functional observations with low level signal measures, so that proposed tests best match expert human intuition. This method is very powerful to diagnose extreme faults (short-circuits, open circuits, etc.), which are easily anticipated. However, dictionary based methods are limited by the fact that, when the actual fault is out of the anticipated set, for instance a parametric fault corresponding to a parameter deviation, the generated tree does not allow the garage mechanic to reach a diagnosis conclusion.

This paper proposes a consistency based method designed for hybrid systems that can complement an available fault dictionary based method, in our case the method of (Ressencourt et al., 2006) based on Modelica models, and uses the same models and simulation results. The only additional information that is required is the structure of the reference models in the form of a causal graph that we are able to derive automatically and the interpretation of the simulation results obtained for continuous variables into qualitative values and events. The principle of consistency based diagnosis methods is to rely on a model of normal behavior that provides a reference. Any deviation from this reference indicates a fault that can be isolated by reasoning about the different parts of the model involved in the discrepancy (Hamscher et al., 1992; de Kleer and Kurien, 2003; Blanke et al., 2003). The hybrid model consistency based method is combined with a test selection procedure to produce an original iterative diagnosis method for

hybrid systems that reduces diagnosis ambiguity at each iteration.

The paper is organized as follows. A discussion on the position of the contribution and related work is given Section 8. Section 2 presents a global view of the iterative diagnosis method. Section 3 defines formally the hybrid causal model. Then Section 4 shows how the continuous behavior of the system is abstracted. The causal models and their use in a fault localization framework are presented in Section 5. Section 6 presents how these different concepts are used together and enhanced with the test selection method to perform iterative consistency based hybrid diagnosis. Finally Section 7 describes a real case study which is the function commanding the rear windscreen wiper of a car.

2. Global view of the iterative consistency based diagnosis method for hybrid systems

Figure 1 gives an overview of the proposed hybrid causal diagnosis method. The numbers appearing in the figure correspond to the different steps explained in the textual description below and that are presented in detail in the rest of the paper. These steps are divided in two main stages.

Design stage: modeling and generation of the partial diagnoser

The modeling step starts with the hybrid model of the system formalized in the form of a *hybrid automaton* ① and aims at abstracting the hybrid automaton into a pure discrete event model. From the hybrid model, we derive three types of mathematical objects that represent three different aspects of the system. The first one is the *underlying DES* ②. The second one refers to the *mode signatures* that capture the qualitative expected values of the observable continuous variables within each behavioral mode

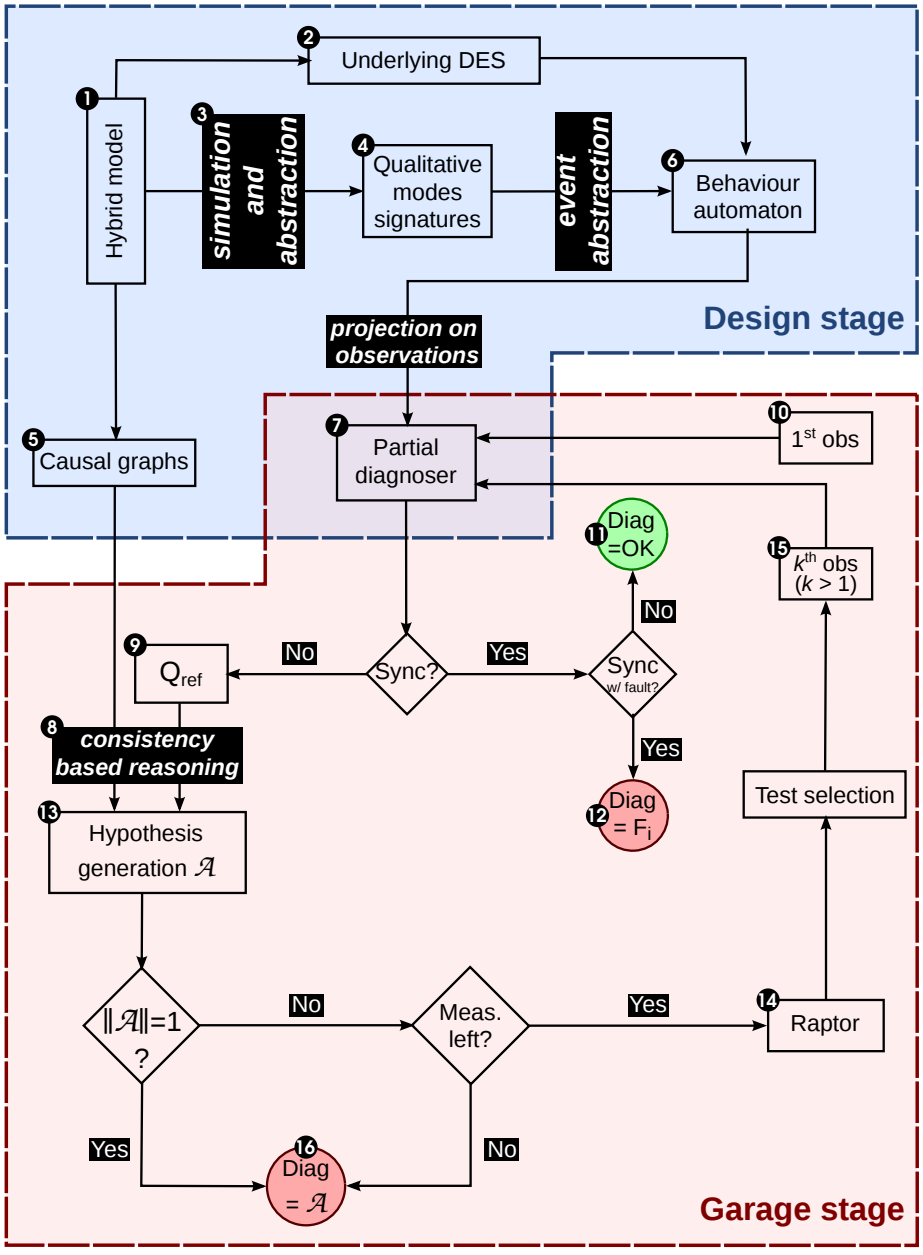

Figure 1: Modeling, detection and diagnosis algorithm.

; they are generated using a dedicated simulation tool for hybrid systems (Modelica) and an abstraction function ③ that computes qualitative mode signatures ④. The third one is a set of *causal graphs* (see Section 5) that describe the causal relationships among the variables for each mode of the system ⑤. The next step abstracts the continuous dynamics captured by qualitative mode signature changes into *signature-events*. Putting together the signature-events with the underlying DES model, we generate the so-called *behavior automaton* ⑥. This automaton provides a pure discrete event view of the hybrid system. The modeling step is completed by the generation of the causal graphs assigned to each mode of the system. The construction of a *partial diagnoser* ⑦ including normal trajectories and some fault trajectories is then performed like in (Sampath et al., 1995) by projecting the behavior automaton onto the observable space.

Garage stage: diagnosis hypotheses generation and test selection

Hypotheses generation is performed by applying *consistency based reasoning* ⑧ based on the causal graphs and on the knowledge of a set of possible reference behavioral modes Q_{ref} for checking the consistency. The set Q_{ref} ⑨ is obtained when the sequence of *observations* gathered by the car mechanic is not consistent with the expected discrete behaviours. In other cases, the fault (or normal situation) associated with the consistent behaviour is reported (⑪ and ⑫).

At the end of the hypotheses generation step a diagnosis *ambiguity set* $\mathcal{A}$ is generated ⑬. If $\mathcal{A}$ is not a singleton and if there are some measurements left, the test selection step takes place. The method chosen for selecting the tests is RAPTOR ⑭, proposed by (Gonzalez-Sanchez et al., 2011a) (see Section 6.3). With the new *selected test* ⑮, i.e. the variable to be measured and

the system configuration, the *partial diagnoser* is updated and the process is reiterated. If $\mathcal{A}$ is a singleton or if there are no measurements left, the *diagnosis* result ⑯ is given by $\mathcal{A}$.

3. The hybrid causal model

The hybrid system at hand is modeled by an extended transition system whose discrete states represent the modes of operation for which the continuous dynamics are characterized by a qualitative domain, called the *qualitative mode signature* defined in Section 4.1, and a causal model. The transition system in itself constitutes the *underlying DES* and constrains the possible transitions among modes. Formally, a hybrid causal system is defined as a tuple :

$$\Gamma = (\mathcal{X}, \mathcal{D}, Conf, Sig, T, \Sigma, CSD, Init) \tag{1}$$

where:

- $\mathcal{X} = \{x_i\}$ is a set of qualitative variables, obtained from continuous variables as explained in Section 4.1. They correspond to state and input/output variables and are functions of time t. The set of qualitative variables that are or can be measured is denoted by $\mathcal{X}_{OBS}$. [1]

- $\mathcal{D}$ is a set of discrete variables. $\mathcal{D} = Q \cup \mathcal{K} \cup \mathcal{H}$, where Q is the set of states q_i of the transition system, representing operation modes of the system. $\mathcal{K} = \{K_i, i = 1, \ldots, n_c\}$ is a set of auxiliary discrete variables used to represent the system configuration in each mode

[1] We assume that the set of system observable variables is the same in all system modes. This assumption is generally verified when the set of system's sensors is permanent.

q_i as defined below by $Conf(q_i)$. $\mathcal{H}$ is the set of discrete variables whose value changes trigger an event σ as defined below. Discrete variables may not be directly observable but their observability may be achieved through the observability of events.

- $Conf$ and Sig are a couple of functions that define two domains for each mode, the *configuration* and the *qualitative mode signature*:
  - $Conf(q_i) : Q \rightarrow \otimes_i D(K_i)$[2], where $D(K_i)$ is the domain of $K_i \in \mathcal{K}$, provides the *configuration* associated to the mode, i.e. the modes of the underlying multimode components (typically, a switch has two normal modes, *open* and *closed*);
  - $Sig(q_i) : Q \rightarrow \otimes_i D(x_i)$, where $D(x_i)$ is the domain of $x_i \in \mathcal{X}$, provides the *qualitative signature* of the mode.
- Σ is a finite set of events, noted σ, associated to the transitions. There may be guards expressing boolean conditions depending on qualitative variables. Σ_o is the set of observable events and Σ_{uo} is the set of unobservable events.
- $T : Q \times \Sigma \rightarrow Q$ is the transition function. The transition from mode q_i to mode q_j with associated event σ is noted (q_i, σ, q_j) or $q_i \stackrel{\sigma}{\rightarrow} q_j$. Without loss of generality, we assume that the model is deterministic, i.e. whenever $q_i \stackrel{\sigma}{\rightarrow} q_j$ and $q_i \stackrel{\sigma}{\rightarrow} q_k$ then $q_j = q_k$ for each $(q_i, q_j, q_k) \in Q^3$ and each $\sigma \in \Sigma$.
- $CSD \supseteq \bigcup_i CSD_i$ is the *Causal System Description*, or causal model, used to represent the constraints underlying the continuous dynam-

[2] $\otimes$ is the Cartesian product.

ics of the hybrid system. Every CSD_i, associated to a mode q_i, is given by a graph $(\mathcal{V} = \mathcal{X} \cup \mathcal{K}, A_i)$. There is an edge $e(v_i, v_j) \in A_i$ from $v_i \in \mathcal{V}$ to $v_j \in \mathcal{V}$ if variable v_i influences variable v_j. Vertices represent variables, edges represent *influences* between variables and every edge is labelled by a component composing the system. The set of influences and the set of components are noted $\mathcal{I}$ and $COMP$, respectively (cf. Section 5).

- $Init \in \mathcal{X} \times \mathcal{D}$ is the initial condition of the hybrid system. $q_0 \in Q$ is the initial mode.

4. Abstractions of the continuous behavior

4.1. Qualitative abstraction of continuous behavior

Although the behavior of the systems that we consider involves continuous dynamics, diagnosis focuses on identifying the mode of operation. The idea of the qualitative abstraction is to partition the domain value of the continuous variables into a finite number of labels such that the label remains invariant when the system is operating within a given mode. Then, *qualitative signatures* can be defined for every mode.

The domain $D(x_i)$ of a qualitative variable $x_i \in \mathcal{X}$ is obtained through a function $f_{qual} : D(x_i^c) \rightarrow D(x_i)$ that maps the continuous domain $D(x_i^c) \subseteq \mathbb{R}$ of the original continuous variable x_i^c into a finite discrete domain, generally built from a partition of $D(x_i^c)$.

In choosing the qualitative abstraction functions, the transition guards must remain expressible. For instance, if a guard is given by the condition $x_i^c > k$, k being a constant, then k must be among the landmarks of the partition of $D(x_i^c)$ leading to $D(x_i)$.

The qualitative abstraction corresponds to the target function. As an example of qualitative abstraction, consider the input voltage of a motor. If one is interested in the motor running or being off, for this continuous variable, the abstract values 0 and 1 will be chosen to represent the real values $0\pm\epsilon$V (ground voltage) and $12\pm\epsilon$V (voltage delivered by the battery), respectively, where ϵ is an uncertainty parameter accounting for noise.

The possibility to address parametric faults through qualitative signatures depends on the sharpness of the partition of $D(x_i^c)$ and, of course, on the available sensors.

4.2. Qualitative mode signatures

Qualitative signatures characterize the expected values of the observable qualitative variables within a given mode. These values remain constant in a mode. Define as $[x_{OBS}]$ the vector composed of the qualitative variables $x_i \in \mathcal{X}_{OBS}$.

Definition 1 (Qualitative mode signature)**.** *The qualitative signature of a mode q_i noted $Sig(q_i)$ is the qualitative valuation of the vector $[x_{OBS}]$ in this mode:*

$$Sig(q_i) = [x_{OBS}]_{q_i} \tag{2}$$

It is important to notice that the qualitative abstraction function f_{qual} is defined such that a mode q_i has one and only one qualitative signature. This qualitative signature characterizes this mode with respect to the other modes.

2-lights circuit example

Let us consider the electric circuit shown in Figure 2, named the 2-lights circuit, that will be used as a running example.

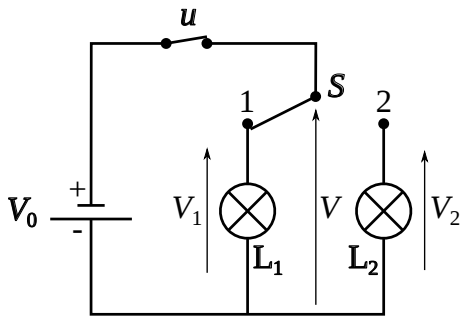

Figure 2: The 2-lights circuit.

The command u opens and closes the circuit and allows the voltage generator on the left branch to switch on the lights L_1 or L_2, depending on the position of the switch S, which is not observable. Since there are two switches, this system has $2^2 = 4$ configurations and operating modes q_1 to q_4. q_1 is the mode in which the switch u is closed and S is in position 1, therefore the light bulb L_1 is lit and L_2 is not. When S is switched to position 2, L_1 turns off, L_2 turns on and the mode q_2 is active. When u opens, both lights are unpowered and the circuit is in mode q_3. If S is switched back to position 1, both lights are still off because of u being opened and the system is in mode q_4.

The qualitative abstraction corresponds to the target function in the normal operating mode, which is the shining function of the light bulbs. If $V_i, i = 1, 2$ is not zero, then the light bulb L_i shines and it does not shine otherwise. The qualitative variables and mode signatures are given in Table 1.

Following the idea proposed in (Bayoudh et al., 2008; Bayoudh and Travé-Massuyès, 2012), qualitative mode signatures are used to generate *signature-events* that inform about the changes of operation mode based on

Table 1: Qualitative variables and mode signatures for the 2-lights circuit.

	q_1	q_2	q_3	q_4
u	1	1	0	0
V_1	1	0	0	1
V_2	0	1	1	0

the observed qualitative variables.

The corresponding event generator is defined by the abstraction function $f_{Sig\to\sigma}$ that maps qualitative signature changes to a set of discrete events Σ^{Sig} called *signature-events*. Σ^{Sig} is partitioned into observable (Σ_o^{Sig}) and unobservable (Σ_{uo}^{Sig}) signature-events, depending on whether the mode signature of the source mode is different from the mode signature of the destination mode or not .

$$f_{Sig\to\sigma} : Q \times T(Q,\Sigma) \longrightarrow \Sigma^{Sig}$$
$$(q_i, q_j) \longmapsto \begin{cases} r_{i,j}^{o} \in \Sigma_o^{Sig} \text{ if } Sig(q_i) \neq Sig(q_j) \\ r_{i,j}^{uo} \in \Sigma_{uo}^{Sig} \text{ if } Sig(q_i) = Sig(q_j) \end{cases} \tag{3}$$

2-lights circuit example

The signature events of the 2-lights circuit are given in Table 2.

Figure 3 gives the discrete part of the model, i.e. the underlying discrete event system $M = (Q, T, \Sigma, q_0)$.

The event $\sigma_{1,2}$ (resp. $\sigma_{2,1}$) represents the change of position of the switch S from position 1 to position 2 (resp. position 2 to position 1). The event $\sigma_{O,C}$ (resp. $\sigma_{C,O}$) represents the change of position of the command switch u from opened to closed (resp. closed to opened). All these events are

Table 2: Signature-events for the 2-lights circuit.

$Sig(q_1) \underset{r^o_{2,1}}{\overset{r^o_{1,2}}{\rightleftarrows}} Sig(q_2)$	$Sig(q_2) \underset{r^o_{3,2}}{\overset{r^o_{2,3}}{\rightleftarrows}} Sig(q_3)$
$Sig(q_3) \underset{r^{uo}_{4,3}}{\overset{r^{uo}_{3,4}}{\rightleftarrows}} Sig(q_4)$	$Sig(q_4) \underset{r^o_{1,4}}{\overset{r^o_{4,1}}{\rightleftarrows}} Sig(q_1)$

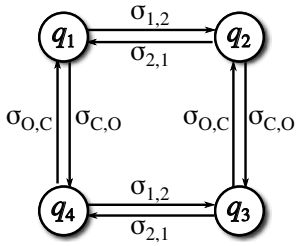

Figure 3: Underlying DES of the 2-lights circuit.

observable.

4.3. Behaviour automaton

The abstraction of the continuous dynamics in terms of discrete events allows us to define an abstract language to describe the behavior of the hybrid system. We denote by $\overline{\Sigma} = \Sigma \cup \Sigma^{Sig}$ the alphabet that contains "natural" discrete events and *signature-events*.

$\overline{\Sigma}$ can be partitioned into $\overline{\Sigma} = \overline{\Sigma}_o \cup \overline{\Sigma}_{uo}$ with $\overline{\Sigma}_o = \Sigma_o \cup \Sigma_o^{Sig}$ and $\overline{\Sigma}_{uo} = \Sigma_{uo} \cup \Sigma_{uo}^{Sig}$.

The behavior of the hybrid system is hence modeled by the prefix-closed language $L(\Gamma)$.

Definition 2 (Language of the hybrid system)**.** *The language generated by the system* Γ *is the set* $L(\Gamma) \triangleq \{s \in \overline{\Sigma}^* | q_0 \xrightarrow{s}\}$ *whose elements are called trajectories*

of Γ.

The set of finite sequences over $\overline{\Sigma}$ is denoted by $\overline{\Sigma}^*$, and ϵ is the empty sequence. The finite state generator (Ramadge and Wonham, 1989) of the language $L(\Gamma)$ is called the *behavior automaton* and denoted:

$$B_A(\Gamma) = (\overline{Q}, \overline{\Sigma}, \overline{T}, q_0). \tag{4}$$

The behavior automaton is obtained by defining a set of *transient* modes Q_t that model the continuous dynamics reaction to the occurrence of a mode change, and hence lead to the generation of a signature-event of Σ^{Sig}. We first define the bijective function f_t that associates a transient mode to each transition $t(q_i, \sigma_{i,j}, q_j) \in T$ of the underlying DES $M = (Q, \Sigma, T, q_0)$. The set of transient modes is obtained as follows :

$$\begin{aligned} f_t : T &\longrightarrow Q_t \\ t(q_i, \sigma_{i,j}, q_j) &\longmapsto q_{i,j} \end{aligned} \tag{5}$$

The set of modes of the behavior automaton is then given by $\overline{Q} = Q \cup Q_t$. The set of observable states of the behavior automaton is given by

$$\overline{Q}_o = \{q_0\} \cup \left\{ q \in \overline{Q}, \exists (q', \sigma) \in \overline{Q} \times \overline{\Sigma}_o \mid \overline{T}(q', \sigma) = q \right\} \tag{6}$$

The partial transition function $\overline{T} \subseteq (\overline{Q} \times \overline{\Sigma} \longrightarrow \overline{Q})$ is decomposed in two partial transition functions as follows:

$$\begin{aligned} &\overline{T} = \overline{T}^1 \cup \overline{T}^2 \\ &\text{with} \begin{cases} \overline{T}^1 \subseteq (Q \times \Sigma \longrightarrow Q_t) \\ \overline{T}^2 \subseteq (Q_t \times \Sigma^{Sig} \longrightarrow Q) \end{cases} \end{aligned} \tag{7}$$

The behavior automaton $B_A(\Gamma) = (\overline{Q}, \overline{\Sigma}, \overline{T}, q_0)$ is obtained by replacing every transition $t(q_i, \sigma_{i,j}, q_j)$ in $M = (Q, \Sigma, T, q_0)$ by two transitions in

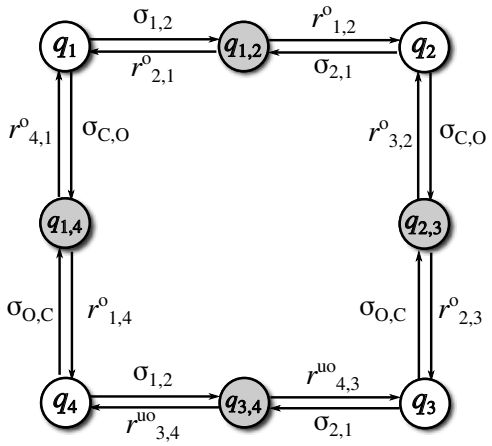

Figure 4: Behaviour automaton of the 2-lights circuit.

sequence $t_1(q_i, \sigma_{i,j}, q_{i,j}) \in \overline{T}^1$ and $t_2(q_{i,j}, r_{i,j}, q_j) \in \overline{T}^2$, the transient mode $q_{i,j} \in Q_t$ hence coming in between q_i and q_j.

Informally, this means that on the occurrence of an event $\sigma_{i,j} \in \Sigma$ that triggers a transition from mode q_i to mode q_j, the system goes through a transient mode $q_{i,j}$ in which the transition is not yet effective. The transition to mode q_j is confirmed by the occurrence of the corresponding signature-event $r_{i,j}^{o/uo}$, providing evidence (when observable) of the response of the continuous dynamics [3].

2-lights example

Figure 4 shows the automaton behavior of the 2-lights circuit.

[3]Notice that, by construction, mode signatures cannot change while being in the same mode.

4.4. The partial diagnoser

Once the hybrid system behavior Γ has been abstracted in the form of a behavior automaton $B_A(\Gamma)$, the diagnoser approach of (Sampath et al., 1995) is applicable. The obtained diagnoser can be viewed as a *partial diagnoser* because it does not account for all the fault trajectories but only those anticipated in the functional specifications. We define:

- $L_o(\Gamma, q)$ the set of all strings that originate from the state q and end at the first observable event:

$$L_o(\Gamma, q) = \{s \in L(\Gamma, q) \mid s = u\sigma, u \in \overline{\Sigma}_{uo}^*, \sigma \in \overline{\Sigma}_o\}; \tag{8}$$

- $L_\sigma(\Gamma, q)$ those strings in $L_o(\Gamma, q)$ that end at the particular observable event σ:

$$L_\sigma(\Gamma, q) = \{s \in L_o(\Gamma, q) \mid s_f = \sigma\}; \tag{9}$$

  with s_f the final event of a string s.

- the set of event labels $\mathcal{L}_E = \{E_1, E_2, \ldots, E_\lambda\}$, where λ is the number of different unobservable events in the system, $\lambda = |(\overline{\Sigma}_{uo})|$. The set of possible labels is defined as $\mathcal{L} = 2^{\mathcal{L}_E}$.

- The label propagation function $LP : \overline{Q}_o \times \mathcal{L} \times \overline{\Sigma}^* \rightarrow \mathcal{L}$. Given $q \in \overline{Q}_o$, $l \in \mathcal{L}$ and $s \in L_o(\Gamma, q)$, LP propagates the label l over s, starting from q and following the dynamics of Γ, i.e. according to $L(\Gamma, q)$:

$$LP(q, l, s) = \begin{cases} \emptyset \text{ if } l = \emptyset \text{ and } \forall \sigma \in s, \sigma \in \overline{\Sigma}_o \\ \{E_k | E_k \in l\} \cup \{E_i | \exists \sigma \in \overline{\Sigma}_{uo}, \sigma \in s\} \text{ otherwise.} \end{cases} \tag{10}$$

The diagnoser of the hybrid system is a deterministic finite state machine built from the behavior automaton, $PDiag(B_A(\Gamma)) = (Q_{PD}, \Sigma_{PD}, T_{PD}, q_{PD_0})$ where:

- $q_{PD_0} = \{(q_0, \emptyset)\}$ is the initial state of the partial diagnoser (assuming Γ is normal to start with);
- $\Sigma_{PD} = \overline{\Sigma}_o$ is the set of all observable events of the system;
- $Q_{PD} \subseteq 2^{\overline{Q}_o \times \mathcal{L}}$ is the set of states of the partial diagnoser where $\mathcal{L} = 2^{\overline{\Sigma}_{uo}}$. The states of the partial diagnoser provide a set of couples whose first element refers to the state of the behavior automaton and the second is a label providing the unobservable events on the path leading to this state. In other words, an element $q_{PD} \in Q_{PD}$ is a set $q_{PD} = \{(q_1, l_1), (q_2, l_2), \ldots, (q_n, l_n)\}$, where $q_i \in \overline{Q}_o$ and $l_i \in \mathcal{L}$. l_i is of the form $l_i = \emptyset$ or $l_i = \{E_{i_1}, E_{i_2}, \ldots, E_{i_k}\}$ where $\{i_1, i_2, \ldots, i_k\} \subseteq \{1, 2, \ldots, \lambda\}$.
- $T_{PD} \subseteq Q_{PD} \times \overline{\Sigma}_o \rightarrow Q_{PD}$ is the partial transition function of the diagnoser defined as follows:

$$T_{PD}(q_{PD}, \sigma) = \bigcup_{\substack{(q,l) \in q_{PD} \\ s \in L_\sigma(\Gamma, q)}} \{(\overline{\mathcal{T}}(q, s), LP(q, l, s))\}. \tag{11}$$

  $\overline{\mathcal{T}}(q, s)$ is the recursive application of $\overline{T}$ along the string $s = s_1 s_2 \ldots s_n \sigma$ of events defined as $\overline{T}(q, s) = \overline{T}(\ldots \overline{T}(\overline{T}(q, s_1), s_2), \ldots, s_n), \sigma)$.

The reader must notice that the diagnoser (Sampath et al., 1995) has been fully quoted in the previous paragraph. However the method proposed in this paper does not make use of the labels $l_i \in \mathcal{L}$. They are not computed by the software DIADES that we use and will not appear in the rest of the paper.

2-lights example

The partial diagnoser is automatically built using the DIADES software (Pencolé, 2013). DIADES takes as input a text file that describes the behavior automaton of the system and lists the observable and non-observable events. Applying the diagnoser approach to the 2-lights circuit example leads to the partial diagnoser shown in Figure 5.

5. Causal models

Causal models have been shown to be suitable for diagnosis in several pieces of work (Venkatasubramanian et al., 2003). In (Narasimhan and Biswas, 2007a; Travé-Massuyès et al., 2001; Travé-Massuyès, 2014), the causal model is proposed as a substitute of dependency recording mechanisms like the ATMS (de Kleer and Williams, 1987). Also, the generation of analytical redundancy relations from causal models is addressed in (Svard and Nyberg, 2010).

Causal models are supported by an oriented graph, also called *causal graph*. The causal graph is used for explanations purposes and not for simulations ones. In this case the causal graph may contain cycles and there is no need to tackle time related issues. In such a graph, vertices represent variables and edges represent influences from variable to variable. An oriented edge from variable v_i to variable v_j exists if v_i has an influence on v_j, i.e. if a value change on variable v_i affects the value of variable v_j. v_i and v_j are called the *cause* and the *effect* variables of the influence, respectively. As explained in Section 5.1, influences represent the causal structure of the underlying equational model (Travé-Massuyès and Pons, 1997) but they may also capture behavioral information when adequately labelled (Gentil et al.,

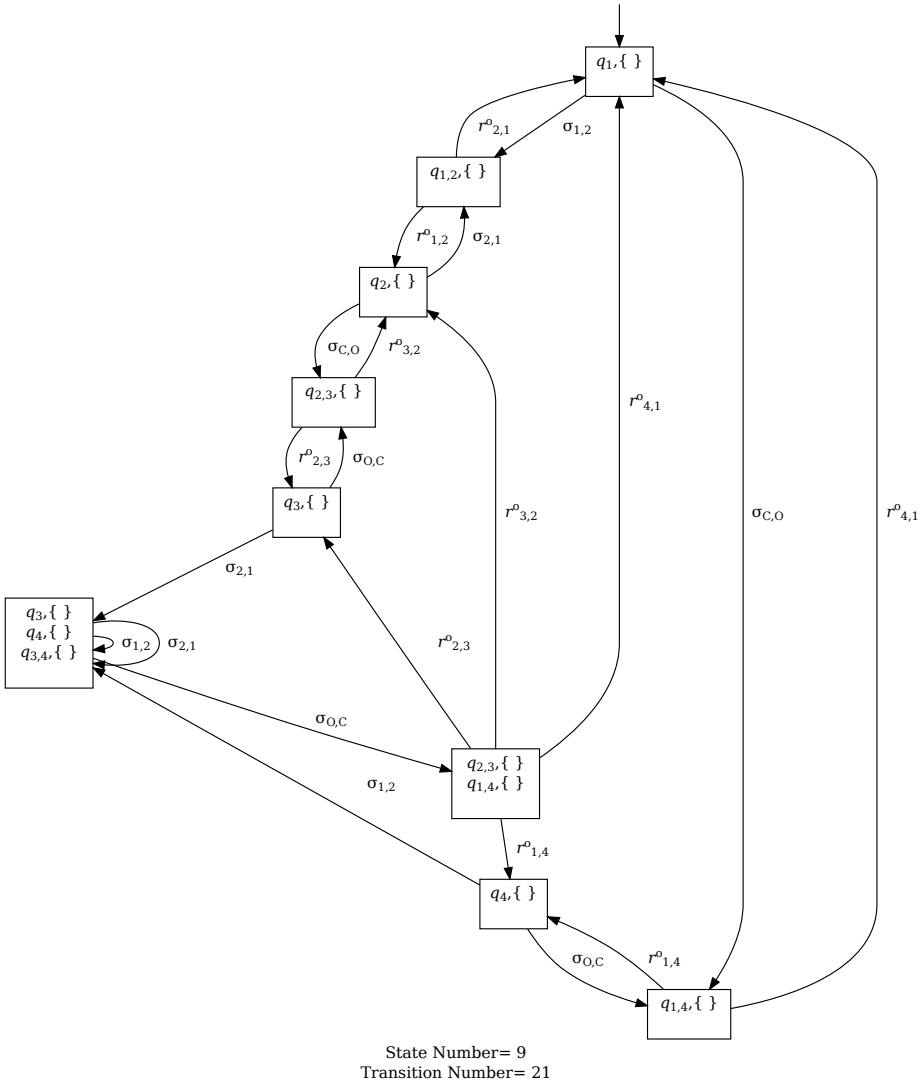

Figure 5: Partial diagnoser of the 2-lights circuit.

2004; Heim et al., 2003; Travé-Massuyès and Calderon-Espinoza, 2007).

5.1. Automatic derivation of the causal model

The theory of *causal ordering* issued from the Qualitative Reasoning community can be advantageously applied to derive automatically the causal structure associated to a set of equations (Iwasaki and Simon, 1986, 1994). Deriving the causal model of a system in a given operating mode implies to gather the equations that represent the behavior of the system in this mode. Their structure is then represented in the form of a bipartite graph, composed of two disjoint vertices sets: the equation vertices on one side and the variable vertices on the other side. There exists an edge between an equation vertex and a variable vertex if the variable appears in the equation. The idea of causal ordering is to match a variable to every equation, then interpret every equation as a causal mechanism that can be used to solve for the matched variable. This step is performed by searching for a perfect matching in the bipartite graph (Hopcroft and Karp, 1973). An equation hence gives rise to a bunch of influences starting at non matched variable vertices and all pointing to the matched variable vertex. This graph pattern, called an *equation bunch*, is the primary pattern of the Causal System Description.

(Travé-Massuyès and Pons, 1997) extended causal ordering to systems with several operating modes, by associating activation conditions to the equations. The influences of the resulting graph consequently carry activation conditions as well. The proposed algorithm, implemented in the

Causalito software[4], makes use of conditions that avoid recomputing a totally new perfect matching for every operating mode, thus reducing the computational cost.

In this work, the variables at hand belong to the set $\mathcal{V} = \mathcal{X} \cup \mathcal{K} \cup \mathcal{H}$ (cf. Section 3) where $\mathcal{V}_{OBS} \subseteq \mathcal{V}$ denotes the subset of observed, i.e. measured, variables, and we call $\mathcal{I}$ the set of influences. The Causal System Description is hence given by $CSD = (\mathcal{V}, \mathcal{I})$, where each influence is labeled with:

- an activation condition stating the modes in which it is active (or no label if it is active in all modes),
- the corresponding equation,
- the component whose behavior is represented by the equation.

2-lights circuit example

In the 2-lights circuit example of Figure 2, we define the boolean conditions C_u, which is true when the circuit is closed by the command u, $\neg C_u$ which is true when the circuit is open, and C_1, which is true when the switch S is connected to the light L_1, and $\neg C_1$ which is true when the switch S to the light L_2. V_1 and V_2 are the voltages of the lights L_1 and L_2.

The equations of the 2-lights circuit are

$$V = V_0 \text{ if } C_u \tag{12}$$

$$V = 0 \text{ if } \neg C_u \tag{13}$$

$$V_1 = V \text{ if } C_1 \tag{14}$$

[4]Available on the PLUME project website https://www.projet-plume.org/relier/causalito

$$V_1 = 0 \text{ if } \neg C_1 \tag{15}$$

$$V_2 = 0 \text{ if } C_1 \tag{16}$$

$$V_2 = V \text{ if } \neg C_1 \tag{17}$$

The causal graphs of the system modes are given in Figure 6.

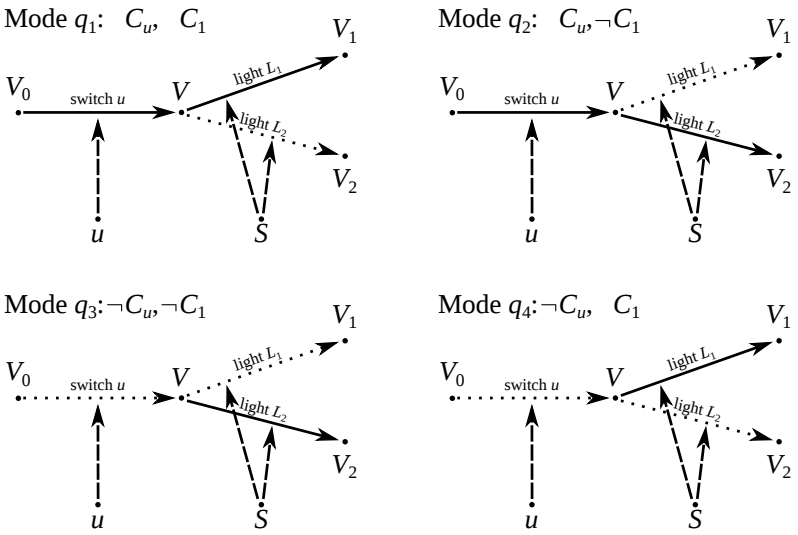

Figure 6: Causal graphs of the 2-lights circuit in the four modes.

Let us notice that in this simple example, the switch related to the command u as well as the switch S are activated exogeneously.

5.2. The causal diagnosis problem

The causal model is used for explanatory purposes, based on a preliminary labeling process of the vertices corresponding to observed variables. An observed variable is qualified as *normal*/*misbehaving* at some time point when there is a match/discrepancy between the measured value and the predicted value obtained from the Modelica simulations. A discrepancy

for variable $v_i \in \mathcal{V}_{OBS}$ is noted with a predicate as $KO(v_i)$, otherwise the variable is qualified as $OK(v_i)$. The corresponding vertices are labelled accordingly. Exogenous input variables are unconditionally qualified as $OK(v_i)$ and so are the labels of the vertices with no in-going influences in the causal graph. The causal graph is explored backwards to determine the cause(s) of discrepancies using the results of the logical theory of Model-Based Diagnosis (MBD) (Reiter, 1987). The main concepts and results of MBD are summarized and interpreted in the causal modeling framework below.

Diagnosis must explain the detected discrepancies by providing the *health status* of each component $C_i \in COMP$, i.e. $AB(C_i)$ if abnormal and $\neg AB(C_i)$ if normal. A component is qualified AB if and only if at least one of its underlying influences is AB[5]. The set of observations *OBS* is defined as follows:

Definition 3 (Observations)**.** *A set of observations OBS is given by the tuple that qualifies every observed variable $v_i \in \mathcal{V}_{OBS}$ as $KO(v_i)$ or $OK(v_i)$.*

Definition 4 (Diagnosis problem)**.** *A diagnosis problem is defined as the triple (CSD, COMP, OBS) where CSD is the Causal System Description, COMP the set of components and OBS a set of observations.*

The set of observations OBS defines a partial labeling of the vertices of CSD. When one or several vertices are labeled KO, the diagnosis system must derive all sets of faulty components of *COMP*, or equivalently all health status assignments, that are consistent with the observations OBS.

[5]The AB (resp.$\neg AB$) predicate is used to qualify abnormal (resp. normal) components while the OK (resp.KO) predicate is used for variables.

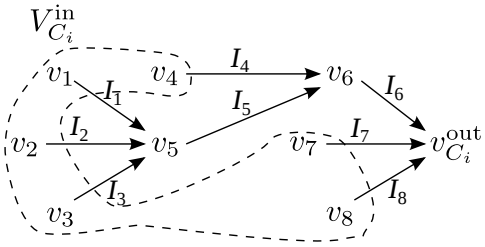

Figure 7: Subgraph $\mathcal{G}_{C_i}$ of a component C_i.

Consistency is defined with respect to a model as explained below.

Without loss of generality, consider that CSD corresponds to one single operating mode. In CSD, all the in-going influences of any vertex with non zero in-degree have the same supporting component as they correspond to an equation bunch (cf. Section 5.1). If the behavior of a component C_i is represented by several equations, then the causal representation of the behavior of C_i corresponds to a connected subgraph $\mathcal{G}_{C_i}$ composed of equation bunches. Consider a component C_i and its associated subgraph $\mathcal{G}_{C_i}$ as exemplified in Figure 7, then the set of input vertices is noted $V_{C_i}^{\text{in}}$, the output vertex is noted $v_{C_i}^{\text{out}}$and the set of influences from $V_{C_i}^{\text{in}}$ to $v_{C_i}^{\text{out}}$ is noted $\mathcal{I}_{C_i}$. In this example, $V_{C_i}^{\text{in}} = \{v_1, v_2, v_3, v_4, v_7, v_8\}$ and $\mathcal{I}_{C_i} = \{\mathcal{I}_1, ..., \mathcal{I}_8\}$.

The component C_i is AB if and only if at least one of the influences in $\mathcal{I}_{C_i}$ is AB, it is qualified OK otherwise. In the same way, $V_{C_i}^{\text{in}}$ is qualified as KO if and only if at least one of its vertex elements is KO, it is qualified OK otherwise. The labels of $V_{C_i}^{\text{in}}$ and $v_{C_i}^{\text{out}}$ are constrained by the consistency model shown in Table 3, which is generic to all components.

It should be noticed that this consistency model is free of assumptions, i.e. it does not assume single fault neither exoneration [6]. The only true con-

[6]The exoneration assumption states that if $v_{C_i}^{\text{out}}$ is OK then C_i is $\neg AB$ and $V_{C_i}^{\text{in}}$ is OK .

Table 3: Component consistency model.

C_i	$V_{C_i}^{\text{in}}$	$v_{C_i}^{\text{out}}$
$AB/\neg AB$	OK	OK
$AB/\neg AB$	KO	OK
$AB/\neg AB$	KO	KO
AB	OK	KO

straint is given by the last row expressing the fact that if the inputs are all OK and the output is KO, the component is necessarily AB (in the table, "$AB/\neg AB$" means that AB or $\neg AB$ are consistent). However, this constraint can only be checked when all the inputs and the output are labeled, i.e. observed. This is why the constraint is generalized later on and used to define the notion of *R-conflict*.

5.3. Conflict generation and diagnoses computation

Definition 5 (Diagnoses and minimal diagnoses)**.** *A diagnosis for (CSD, COMP, OBS) is a set of components $\Delta \subseteq COMP$ such that the assignment $AB(C_i)$ for $C_i \in \Delta$ and $\neg AB(C_i)$ for $C_i \in COMP - \Delta$ is consistent with CSD and OBS. A minimal diagnosis is a diagnosis Δ such that $\forall \Delta' \subset \Delta$, Δ' is not a diagnosis.*

The notion of *R-conflict*, in the sense of (Reiter, 1987), plays an important role for computing the diagnoses.

Definition 6 (Reiter conflict and minimal conflict)**.** *A conflict in the sense of (Reiter, 1987), or R-conflict, for (CSD, COMP, OBS) is a set of components $S = \{C_1, \ldots, C_k\} \subseteq COMP$ such that the assignment of $\neg AB$ to all $C_i \in S$ is*

inconsistent. A minimal R-conflict is an R-conflict which does not strictly include (in the sense of set inclusion) any R-conflict.

Interpreting the notion of *R-conflict* in our causal framework requires to define the notions of *Observed Macro-Component (OMC)* and *test*.

Definition 7 (Observed Macro-Component (OMC))**.** *An OMC $\mathbb{C}_i$ is defined by a non-zero in-degree output vertex $v_{\mathbb{C}_i}^{\text{out}} \in \mathcal{V}_{OBS}$ and a set of input vertices $v_i \in V_{\mathbb{C}_i}^{\text{in}}$ defined as a set of observed predecessors of (predecessors of) $v_{\mathbb{C}_i}^{\text{out}}$. The behavior of $\mathbb{C}_i$ is represented by the subgraph of CSD $\mathcal{G}_{\mathbb{C}_i}$ given by the in-tree in which only $v_{\mathbb{C}_i}^{\text{out}}$ is reachable from every other vertex.*

Similar to a component (cf. Figure 7), an OMC $\mathbb{C}_i$ has one single output vertex $v_{\mathbb{C}_i}^{\text{out}}$ and a set of input vertices noted $V_{\mathbb{C}_i}^{\text{in}}$ and its subgraph $\mathcal{G}_{\mathbb{C}_i}$ is composed of equation bunches. The difference is that $v_{\mathbb{C}_i}^{\text{out}} \in \mathcal{V}_{OBS}$ and $v_i \in V_{\mathbb{C}_i}^{\text{in}} \Leftrightarrow v_i \in \mathcal{V}_{OBS}$. In other words, an OMC determines a subgraph whose input and output vertices are observed.

Definition 8 (Test and covered components)**.** *The labeling associated to an OMC $\mathbb{C}_i$ is defined as a* test *T_i. The test T_i is said to be based on $v_{\mathbb{C}_i}^{\text{out}}$: if $v_{\mathbb{C}_i}^{\text{out}}$ is labelled KO, than the test is said to* fail *and if it is labelled OK, the test is said to* pass*. The components $C_{j_1}, \ldots, C_{j_{K_i}}$ labeling the influences of $\mathcal{G}_{\mathbb{C}_i}$ are called the* components covered *by T_i, or the* coverage *of T_i.*

The consistency model given in Table 3 for a component extends to OMCs.

Proposition 1 (Potential R-conflict and R-conflict)**.** *The set of components $\{C_{j_1}, \ldots, C_{j_{K_i}}\}$ covered by a test T_i define a* potential R-conflict *in the sense of (Cordier et al., 2004). $\{C_{j_1}, \ldots, C_{j_{K_i}}\}$ is a* R-conflict *if and only if the health status of $\mathbb{C}_i$ is AB.*

Proof. A test T_i is associated to an OMC $\mathbb{C}_i$. Because the inputs and output of $\mathbb{C}_i$ are observed, its health status can be assessed from Table 3. If $\mathbb{C}_i$ is AB, then at least one component underlying $\mathbb{C}_i$ is AB, hence the set of components $\{C_{j_1}, \ldots, C_{j_{K_i}}\}$ covered by T_i is an R-conflict. □

In the causal framework, R-conflicts can be identified according to the following result:

Corollary 1 (Conflict identification)**.** *The set of components* $\{C_{j_1}, \ldots, C_{j_{K_i}}\}$ *covered by a test* T_i *that fails and whose input label is* OK*, i.e.* $V^{\text{in}}_{\mathbb{C}_i}$ *is* OK*, define an R-conflict.*

Proof. From the last row of Table 3, the health status of $\mathbb{C}_i$ is AB if T_i fails and the input label is OK. □

(Reiter, 1987) proved that diagnoses can be computed from R-conflicts.

Proposition 2 (Diagnosis)**.** $\Delta \subseteq COMP$ *is a (minimal) diagnosis for* (CSD, COMP, OBS) *if and only if* Δ *is a (minimal) hitting set for the collection of (minimal) R-conflict sets of* (CSD, 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 R-conflicts was originally proposed by (Reiter, 1987), then corrected by (Greiner et al., 1989).

The set of diagnoses defines the *ambiguity set* $\mathcal{A}$. Every element of the ambiguity set is a *diagnosis hypothesis*.

2-lights circuit example

Let's assume that the 2-lights circuit is in mode q_2 (C_u and $\neg C_1$ are true) and the OMC $\mathbb{C}^*$ defined by $v^{\text{out}}_{\mathbb{C}^*} = V_2$ and $V^{\text{in}}_{\mathbb{C}^*} = \{V_0\}$. V_0 is an input to

the circuit and is always labelled OK. We measure V_2 to be 0, hence the test T_{V_2} fails and is labelled KO, so the health status of $\mathbb{C}^*$ is AB. The coverage of T_{V_2} is the set $\{switch\ u, light\ L_2\}$, which is hence an R-conflict by Corollary 1 (see Figure 8). With this only conflict, the minimal diagnoses are $\{switch\ u\}$ and $\{light\ L_2\}$ and the ambiguity set is $\mathcal{A} = \{switch\ u, light\ L_2\}$.

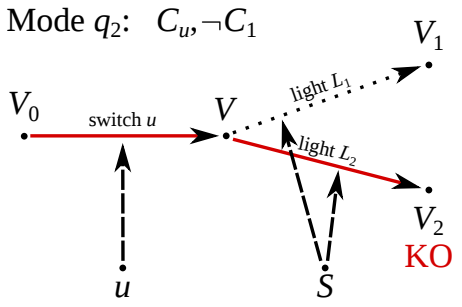

Figure 8: Example of a conflict for the 2-lights circuit

6. Consistency based hybrid diagnosis

A diagnosis consistency based-approach relies on the use of a reference model and on the observation of the real behavior of the monitored system. In our case the model represents a hybrid system is constituted, on one hand by the partial diagnoser $PDiag(B_A(\Gamma))$ and, on the other hand by the Causal System Description CSD. Observations takes the form of a sequence of events $s_{obs} \in L_{obs}(\Gamma, q) = \{s \in L(\Gamma, q) \mid s = u\sigma, u \in \Sigma^*_{hyb_o}, \sigma \in \overline{\Sigma}_o\}$, i.e. s_{obs} is a word of $L_{obs}(\Gamma, q)$. The events may be natural discrete events or signature-events coming from the continuous dynamics. Basing consistency-based diagnosis reasoning on our model requires to interface event-based and variable-based diagnosis reasoning, which is explained in

the next section.

6.1. Interfacing event-based and variable-based diagnosis reasoning

Our diagnosis method interlinks event-based and variable-based reasoning thanks to the addition, in CSD, of a set of vertices corresponding to discrete variables $\mathcal{K}$. These variables represent the switch positions that materialize configurations and whose switches trigger mode changes. The influences outgoing these vertices are different from standard influences since they act on the causal graph structure. The variables in $\mathcal{K}$ are labelled OK/KO by the set of natural discrete events that are actually observed compared to those that are expected from the model.

6.2. The iterative diagnosis process

The diagnosis process iterates diagnosis hypotheses generation and test selection. The set of observed variables hence monotonically increases along the process, implying that qualitative mode signatures must be updated accordingly and so must be the set of signature-events. The behavior automaton and the resulting partial diagnoser must also be updated. In practice, it may be more efficient to build the partial diagnoser for the full measurement situation, i.e. when all the continuous variables are measured, and to derive the successive partial diagnosers by removing the states and events that correspond to non-measured variables. The dependence upon the iteration is indicated by superscripting the corresponding symbols by k : $\mathcal{X}^k_{OBS}$, $Sig^k(q_i)$, $PDiag(B^k_A(\Gamma))$, etc.

At each iteration k, the consistency based hybrid diagnosis algorithm is structured along the following steps:

- *Step 1: Fault detection and reference mode hypotheses generation* – This is achieved by synchronizing $PDiag(B_A^k(\Gamma))$ with s_{obs}. If none synchronized trajectory corresponds to a complete trajectory, i.e. a trajectory that ends with a receptor state or with a cycle, then a fault is detected. The last state of each synchronized trajectory indicates a possible reference mode. These modes are put in the set Q_{ref}^k and the ambiguity set is initialized to $\mathcal{A}^k = COMP$.

- *Step 2: Diagnosis hypotheses generation* – Every hypothesized reference mode $q_i \in Q_{ref}^k$ is likely to provide evidence about the faulty situation of the system. For every $q_i \in Q_{ref}^k$, we consider the corresponding CSD_i, and apply the consistency-based causal diagnosis approach presented in Section 5 to obtain an ambiguity set $\mathcal{A}_i^k$. The global *ambiguity set* $\mathcal{A}$ is then updated as $\mathcal{A}^k = COMP \cap \bigcap_i \{\mathcal{A}_i^k\}$.

- *Step 3: Test selection* – This step determines the best next variable $x_i \in \mathcal{X}_{OBS} - \mathcal{X}_{OBS}^k$ to be tested to maximize ambiguity reduction. It is detailed below in Section 6.3.

- *Step 4: Hypothesis discrimination* – The current ambiguity set $\mathcal{A}^k$ is reduced by going to step 1.

6.3. Test selection

Given an ambiguity set $\mathcal{A}^k$ resulting from processing a subset of tests based on the qualitative variables of $\mathcal{X}_{OBS}^k$, the goal of the test selection step is to determine the best next test T_i based on a variable $x_i \in \mathcal{X}_{OBS} - \mathcal{X}_{OBS}^k$. This test should maximize diagnostic information while minimizing the overall testing cost C_T.

For this purpose, we use the ideas of the RAPTOR (gReedy diAgnostic Prioritization by ambiguiTy Reduction) method (Gonzalez-Sanchez et al., 2011a) in which test selection is based on maximizing diagnosis ambiguity reduction. Diagnostic performance is expressed in terms of a cost metric C_d that measures the excess effort incurred in finding the faulty component. C_d represents the number of inspected components that are not the faulty one or, in other words, the wasted effort.

Tests are characterized by their *coverage*, i.e. set of covered components, as defined in Definition 8. In our method, this information is provided by the CSD. Two components that are covered by the same tests cannot be discriminated. In (Gonzalez-Sanchez et al., 2011a), sets of such undiscriminable components are defined as *ambiguity groups* and RAPTOR is presented to be used off-line, taking iteratively as input a set of *ambiguity groups* and determining the best next test. At the end of the algorithm, a sequence of tests reducing diagnosis ambiguity at best is available. RAPTOR is not theoretically optimal but has been shown to be quite competitive (Gonzalez-Sanchez et al., 2011b).

In our method, we use the test selection on-line based on one iteration of RAPTOR to determine the best next test and we apply the test. The ambiguity set is updated accordingly and constitutes the only ambiguity group given as input for the next iteration. Each test T_i breaks the ambiguity set $\mathcal{A}^k$ into two ambiguity groups $\mathcal{A}^k_1(T_i)$ and $\mathcal{A}^k_2(T_i)$, one corresponding to the components covered by the test, and one corresponding to the components that are not covered.

The expected diagnostic effort if components were picked randomly in $\mathcal{A}^k$ for inspection is:

$$E_D(\mathcal{A}^k) = \frac{|\mathcal{A}^k| - 1}{2} \tag{18}$$

Considering the test T_i and the corresponding set of ambiguity groups $AG(T_i) = \{\mathcal{A}_1^k(T_i), \mathcal{A}_2^k(T_i)\}$, we want to estimate diagnosis quality $Q(AG(T_i))$ which can be seen as an estimation of the residual diagnosis effort.

The expected diagnostic effort if components were picked randomly in each ambiguity group for inspection is $E_D(\mathcal{A}_1^k(T_i))$ and $E_D(\mathcal{A}_2^k(T_i))$. Considering that faults are distributed uniformly in the ambiguity set, we have $Pr(\mathcal{A}_j^k(T_i)) = \frac{|\mathcal{A}_j^k(T_i)|}{|\mathcal{A}^k|}$. Averaging the effort in each group by its probability, the residual diagnosis effort can hence be estimated by :

$$\begin{aligned} Q(AG(T_i)) &= Q(\mathcal{A}_1^k(T_i)) + Q(\mathcal{A}_2^k(T_i)) \\ &= \sum_{j=1}^{2} Pr(\mathcal{A}_j^k(T_i)) \times E_D(\mathcal{A}_j^k(T_i)) \\ &= \sum_{j=1}^{2} \frac{\left|\mathcal{A}_j^k(t_i)\right|}{|\mathcal{A}^k|} \times \frac{|\mathcal{A}_j^k(T_i)| - 1}{2}, \end{aligned} \tag{19}$$

The ambiguity reduction heuristic is defined as the difference in residual diagnosis effort, i.e. in ambiguity, caused by considering test T_i :

$$AR(T_i) = Q(\mathcal{A}^k) - Q(AG(T_i)), \tag{20}$$

where $Q(\mathcal{A}^k) = E_D(\mathcal{A}^k)$ because $Pr(\mathcal{A}^k) = 1$.

If the costs of the tests were to be accounted for, one should consider the ratios $\frac{Q(\mathcal{A}^k)}{C_T^k}$ and $\frac{Q(AG(T_i))}{C_T^k + C_{T_i}}$, where C_T^k is the cost of the executed tests and C_{T_i} is the cost of text T_i. The test T_i which maximizes $AR(T_i)$ is chosen as the best next test. Let us notice that discriminability based on test coverage comes back to adopt the exoneration assumption. The test selection method

is actually based on that a test that passes exonerates the components of its coverage and a test that fails incriminates them. Since exoneration is not an assumption of the diagnosis method presented in Section 5.3, if the selected test is applied and passes, the ambiguity reduction may hence be lower than expected. This means that RAPTOR selects the test that *may eventually* lead to the highest diagnosis ambiguity reduction.

2-lights circuit example

Although this simple example is quite trivial to illustrate test selection, let's follow on with the 2-lights circuit in mode q_2 (assumed to be the reference mode). After the test based on V_2, the ambiguity set is $\{\mathcal{A}^1 = \{switch\ u, light\ L_2\}\}$ (cf. Figure 8) and $Q(\mathcal{A}^1)) = E_D(\mathcal{A}^1)) = \frac{2-1}{2} = 0.5$. If we select the test T_V (V is actually the only unmeasured variable), we obtain two ambiguity groups and we have $Q(AG(T_V)) = \frac{1}{2} \times \frac{1-1}{2} + \frac{1}{2} \times \frac{1-1}{2} = 0$. The ambiguity reduction is obviously maximized since ambiguity is completely resorbed :

$$AR(T_V) = Q(\mathcal{A}^1) - Q(AG(T_V)) = 0.5 - 0 = 0.5 \tag{21}$$

Indeed, if T_V fails, the diagnosis is $\{switch\ u\}$ and if T_V passes, the diagnosis is $\{light\ L_2\}$.

In our hybrid framework, one has to deal with all the reference modes in Q^k_{ref}, accounting for the different test coverages. The test selection strategy presented above is applied for each reference mode and the overall best test is chosen. The ambiguity sets resulting from the different reference modes are intersected to obtain a unique ambiguity set at each iteration (cf. step 2 of the diagnosis algorithm).

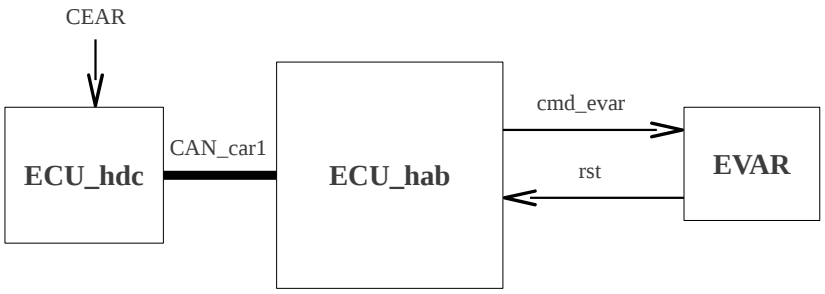

Figure 9: Synoptic of the rear windscreen wiper.

7. Case study

The case study consists in the function of the rear windscreen wiper of a C4 Citroën car whose synoptic is given on Figure 9. The technical material refering to this case study, in particular sketches, was kindly provided by the ACTIA® Group, based in Toulouse, France, that designed the system in the framework of the collaborative OSEO-ISI project AMIC-TCP [7]. Many acronyms are in French. The glossary of the Table 4 provides the acronyms' meaning in French and English.

The system is modeled in the Modelica® language with the sofware Dymola® and is composed of:

- Two electronic control units (*ECU_hdc* and *ECU_hab*) communicating through the Controller Area Network *CAN_car1*. *ECU_hdc* is dedicated to the control of the steering wheel. It acquires and filters information on the position of the rear windscreen wiper switch (*CEAR*) and transmits it to the *ECU_hab* via the CAN. *ECU_hab* is assigned to the passenger compartment. It manages the global operation of the

[7] Available on the ACTIA company website AMIC-TCP program.

Table 4: Glossary

Glossary

CAN: *fr* bus système série de type Controller Area Network, *en* Controller Area Network serial system bus

CAN_car1: *fr* CAN du véhicule *car1*, *en* CAN for vehicle *car1*

CEAR: *fr* commande essuie-vitre arrière, *en* rear windscreen wiper switch

cmd_evar: *fr* signal de la commande essuie-vitre arrière, *en* rear windscreen wiper signal

ECU_hab: *fr* calculateur de l'habitacle, *en* electronic control unit of the passenger compartment

ECU_hdc: *fr* calculateur des commandes au volant, *en* electronic control unit of the steering wheel switches

EVAR: *fr* essuie-vitre arrière, *en* rear windscreen wiper

Hard_ECU_hab: *fr* composant matériel de *ECU_hab*, *en* hardware part of *ECU_hab*

Soft_ECU_hab: *fr* composant logiciel de *ECU_hab*, *en* software part of *ECU_hab*

system and controls the motor of the rear windscreen wiper through the *cmd_evar* request. It acquires and filters the signal rst indicating that the wiper is in the rest position. *ECU_hab* manages also other systems interacting with the rear windscreen wiper such as the front wiper.

- A switch rear windscreen wiper (*CEAR*).
- A rear windscreen wiper module (*EVAR*).

7.1. System description

The *ECU_hdc* is composed of a software part whereas the *ECU_hab* includes both a software and a hardware part (see Figure 10 and Figure 11). The software part *Soft_ECU_hab* receives the wiper switch position signal *CEAR* from *CANcar1* and the wiper rest sensor signal from the hardware part *Hard_ECU_hab*. *Soft_ECU_hab* implements the discrete event system (see Figure 12) that controls the activation of the switch Switch_{K_1} in *Hard_ECU_hab* via *cmd_evar*.

The rear windscreen wiper module is given on Figure 13. This module is composed of an electrical motor, a wiper linkage system and a wiper rest switch Switch_{K_2}.

The rear windscreen wiper can be activated in two ways:

1. By the driver acting on the rear windscreen actuator: In this case the wiping is intermittent and the wiper movement is periodic with a period that includes a forward and backward movement and a stop in rest position.
2. Automatically, when the front wiper is activated and when the driver puts the car into reverse.

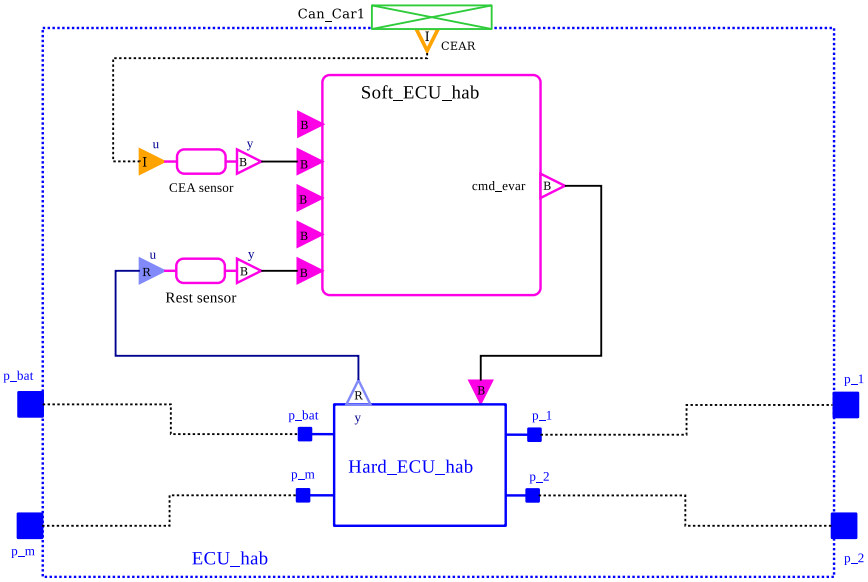

Figure 10: Decomposition of the *ECU_hab*.

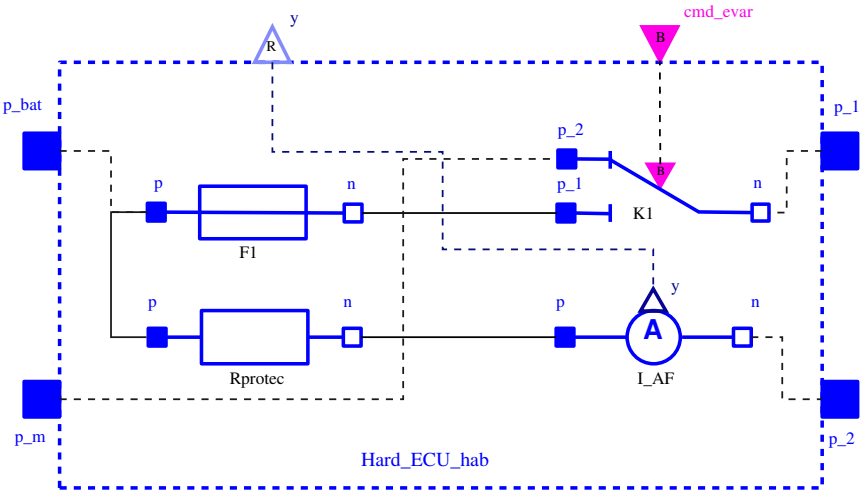

Figure 11: Hardware part of the *ECU_hab*.

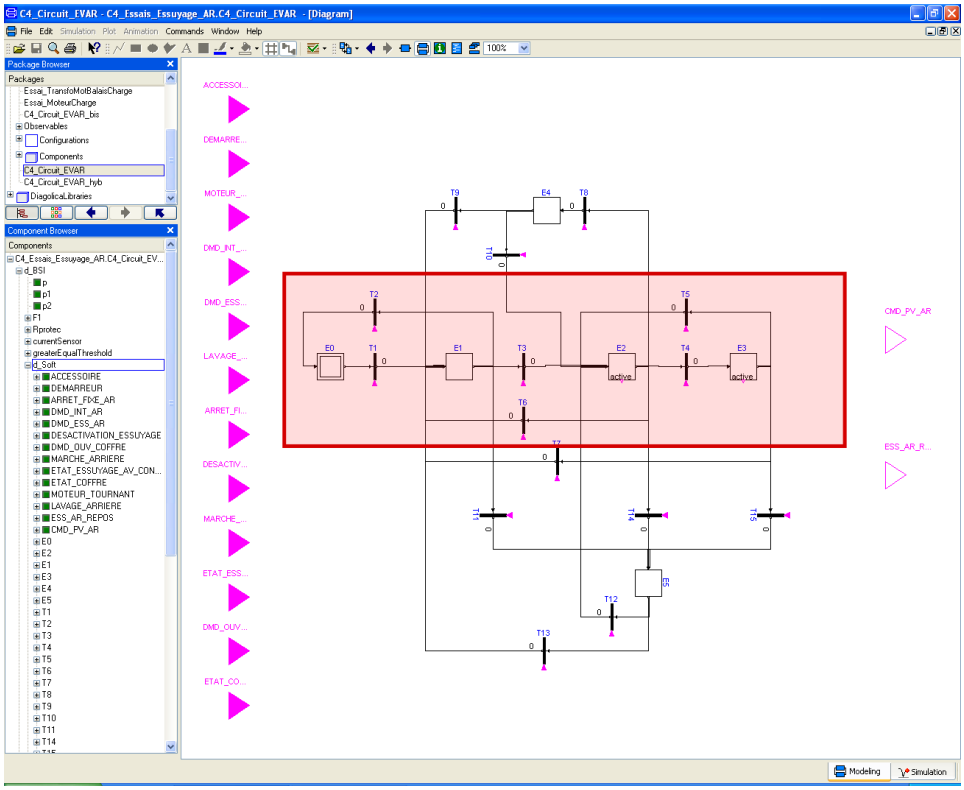

Figure 12: Dymola® software screenshot: discrete event control implemented in *Soft_ECU_hab*.

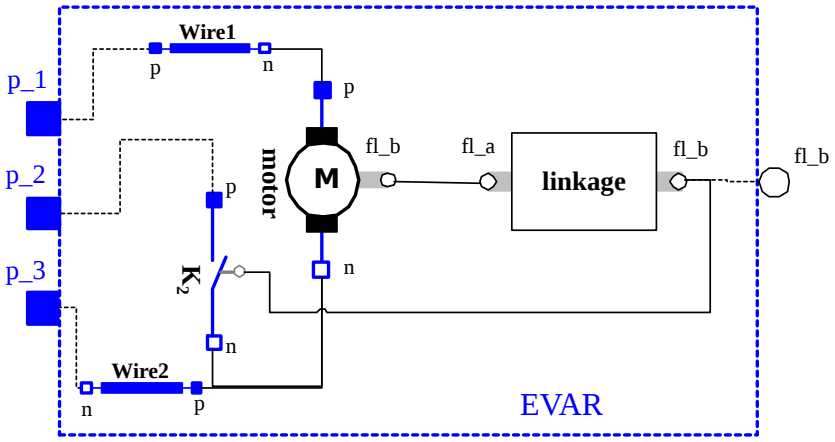

Figure 13: Rear windscreen wiper module.

In the following we consider the simplified description of the system shown in Figure 14 and focus on the intermittent movement of the wiper controlled by the switch position (framed part of Figure 12).

When the driver acts on the actuator *act*, the ECU *ECU_hab* closes the switch Switch_{K_1} and then supplies electrical power from the battery *bat* to the wiper motor M. The rotational move of the motor flange is transformed into an alternative straight move via the wiper linkage that allows the wiper to wipe the screen. After the wiper has moved forward and backward on the screen, it opens the switch Switch_{K_2}, supplying electrical power to the wiper rest sensor that issues the rest position signal *rst*. The sensor sends the signal *rst* back to the *ECU_hab* to indicate that the wiper is in rest position. The *ECU_hab* then opens the switch Switch_{K_1} for a given timeout: the wiper motor is no longer supplied with power and stays in rest position until the timeout expires, then the *ECU_hab* closes the switch Switch_{K_1} again. Hence, the wiper moves forward and backward on the screen, stops during the timeout, etc. until the driver switches the actuator

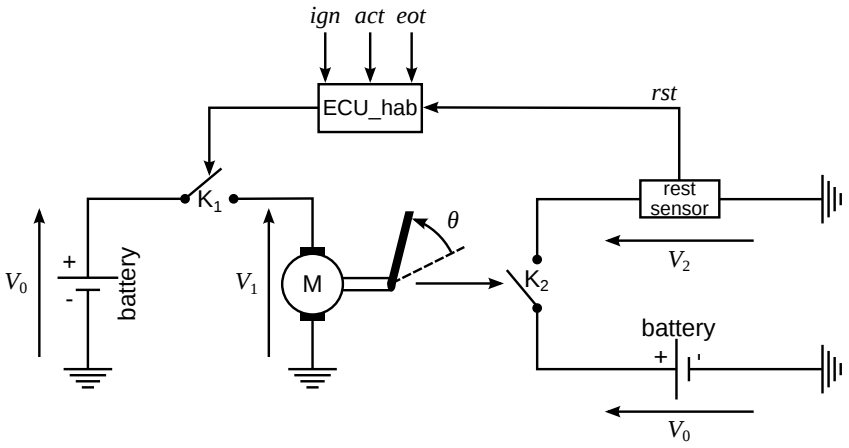

Figure 14: Rear windscreen wiper simplified synoptic.

act off.

7.2. Hybrid features of the windscreen wiper

The windscreen wiper system obviously involves hybrid behavior. Since there are two switches Switch_{K_1} and Switch_{K_2} in the circuit, the underlying system has four modes. In addition, modeling the behavior of this system involves a set of discrete variables and a set of continuous variables.

The discrete variables are *ign* that gives the ignition status, *act* that reflects the actuator position controlled by the driver, *CEAR* that is the wiper switch position signal, *rst* that is the wiper rest position signal, and the end of timeout *eot*. The position (open or closed) of the two switches Switch_{K_1} and Switch_{K_2} come into play as two auxiliary discrete (Boolean) variables K_1 and K_2 that indicate the system *configuration* associated to an operation mode. $K_i = 0/1$ when Switch_{K_i} is opened/closed, $i = 1, 2$. The system has thus four different operation modes.

The continuous variables are the battery voltage V_0, the wiper motor

Table 5: Qualitative variables and mode signatures for the rear windscreen wiper.

	q_0 Off	q_1 On	q_2 Wiping	q_3 Timeout
V_0	1	1	1	1
V_1	0	0	1	0
V_2	1	1	1	0
Ω	0	0	1	0
θ	0	0	1	0

Table 6: Signature-events for the rear windscreen wiper in the full measurement situation.

$Sig(q_0) \xrightarrow{r^{uo}_{0,1}} Sig(q_1)$	$Sig(q_1) \xrightarrow{r^{o}_{1,2}} Sig(q_2)$
$Sig(q_2) \xrightarrow{r^{o}_{2,3}} Sig(q_3)$	$Sig(q_3) \xrightarrow{r^{o}_{3,2}} Sig(q_2)$

input voltage V_1, the rest position sensor output voltage V_2, the wiper angular velocity Ω, and its angular position θ. As explained in Section 4.1, continuous behaviors are captured qualitatively. Hence the domain value of continuous variables is partitioned into an appropriate number of qualitative labels (cf. Section 4.1).

7.3. Design stage

The qualitative variables and their values providing the qualitative mode signatures in each mode are shown in Table 5. Signature-events for the full measurement situation, i.e. when all the continuous variables V_0, V_1,

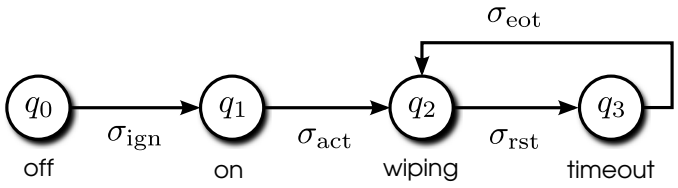

Figure 15: Underlying DES of the rear windscreen wiper.

V_2, Ω, θ are measured, are given in Table 6. Qualitative mode signatures are obtained from the Modelica simulation underlying the fault dictionary method.

In the case study, continuous variable domain partitioning has been chosen to tackle extreme faults such as a broken coil. It must be noticed that incipient faults could be addressed by means of a finer partition. The ability to discriminate incipient faults also relies on the system's instrumentation. As an example, there is no way to detect a change in the motor resistance since the speed of the wiper cannot be evaluated: there is indeed no speed sensor and no way to measure the required time for the wiper to make a forward and backward move.

Figure 15 gives the discrete part of the model, i.e. the underlying DES $M = (Q, T, \Sigma, q_0)$. This model is obtained from the function specification data and corresponds to a simplified version of the control embedded in the software part of the *ECU_hab*. All the events appearing in Figure 15 are observable. Figure 16 gives the behavior automaton for the full measurement situation. The corresponding partial diagnoser is given in Figure 17.

Among the tasks of the design stage, one has to derive the CSD of the system from the structure of its equations. The different components of

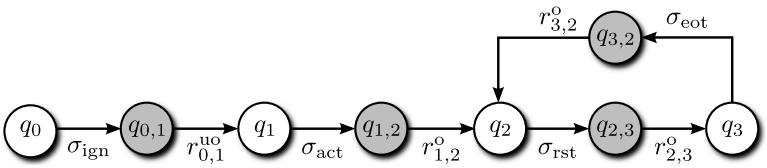

Figure 16: Behavior automaton of the rear windscreen wiper for the full measurement situation.

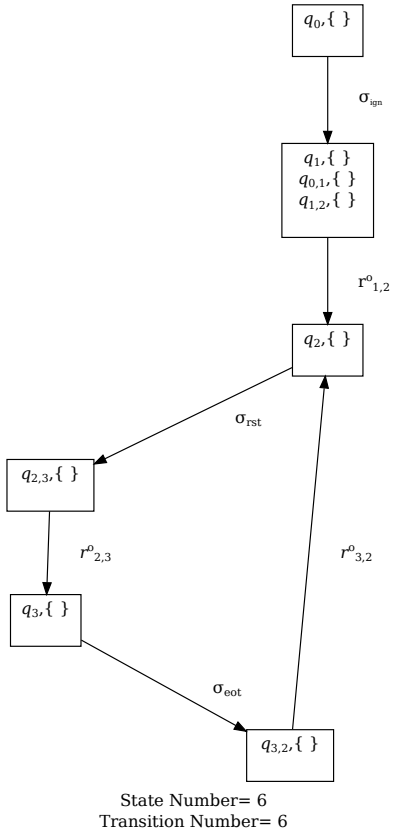

Figure 17: Partial diagnoser $PDiag(B_A(\Gamma))$ for the full measurement situation.

the system (see Figure 14) and their corresponding structural equations are described below.

Structural model of the system. Only structural knowledge is required, i.e. the knowledge of the relations linking the variables and which variables are involved in which relations.

The battery voltage V_0 is considered as an exogenous variable. It must be explicitely specified by the equation:

$$f_0(V_0) = 0. \tag{22}$$

The wiper motor voltage V_1 is related to V_0 and K_1:

$$f_1(V_1, K_1, V_0) = 0. \tag{23}$$

The wiper angular speed Ω is related to V_1:

$$f_2(\Omega, V_1) = 0. \tag{24}$$

and we have the following relation for the wiper position Θ:

$$f_3(\Theta, \Omega) = 0. \tag{25}$$

The wiper position opens or closes the second switch Switch_{K_2}:

$$f_4(\Theta, K_2) = 0. \tag{26}$$

The rest sensor receives the V_2 voltage signal depending on the position K_2 of the Switch_{K_2}:

$$f_5(V_2, K_2, V_0) = 0, \tag{27}$$

and accordingly sends the event σ_{rst} to the *ECU_hab*:

$$f_6(rst, V_2) = 0. \tag{28}$$

The *ECU_hab* opens or closes Switch_{K_1} depending on the values of the ignition signal ign, the actuator position act, the timeout eot and the rest position signal rst:

$$f_7(K_1, ign, act, eot, rst) = 0. \tag{29}$$

7.4. Fault scenario

The fault scenario is the following: the wiper motor coil is broken (opened circuit), the command to move the wiper is sent by the *ECU_hab* (σ_{act} is issued) and the motor is powered but the wiper does not move. Obviously, the wiper never gets to the rest position and the *ECU_hab* never receives the rest position event σ_{rst}. We assume that neither the battery nor its wired connexion are faulty.

7.5. Garage stage

The diagnosis is within the initial ambiguity set:

$$\begin{aligned} \mathcal{A}^0 &= \{ECU\_hab,\ motor,\ wiper,\ rest\ sensor,\ Switch_{K_1},\ Switch_{K_2}\} \\ &= COMP. \end{aligned}$$

The discrete events are always observable as they are linked to the state of the ECUs and can be obtained by a reading of the ECU parameters, i.e. $\Sigma_o = \{\sigma_{\mathrm{ign}}, \sigma_{\mathrm{act}}, \sigma_{\mathrm{rst}}, \sigma_{\mathrm{eot}}\}$.

The garage mechanic starts the diagnosis session by measuring the input voltage of the wiper motor: the set of observable variables at iteration $k = 1$ is hence $\mathcal{X}^1_{\mathrm{obs}} = \{V_1\}$ (cf. Figure 18). Note that $\mathcal{X}^1_{\mathrm{obs}}$ induces the behavior automaton $PDiag(B^1_A(\Gamma))$ of Figure 19[8]. The system issues the

[8] As stated previously, the labels of the diagnoser approach are not used and remain empty.

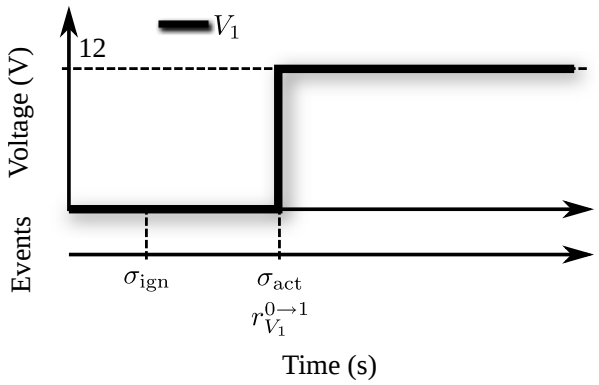

Figure 18: V_1 observed behavior.

signature-events and the observed event sequence $s_{\text{obs}} = \{\sigma_{\text{ign}}, \sigma_{\text{act}}\}$. In particular, the occurrence of the event $r_{V_1}^{0\to 1}$ (V_1 value transitioning from 0 to 1) in the signal of V_1 is a sufficient condition for $r_{1,2}^{o}$ to be issued.

Step 1: Fault detection and reference mode hypotheses generation. The synchronization of the observed event sequence with $PDiag(B_A^1(\Gamma))$ (Figure 19) indicates that the system can be synchronized along the sub-trajectory

$$[\{(q_0, .\}, \{(q_{0,1}, .), (q_1, .)\}, \{(q_{1,2}, .)\}, \{(q_2, .)\}] .$$

This is not a complete trajectory, hence a fault is detected and $Q_{ref}^1 = \{q_2\}$.

Step 2: Diagnosis hypotheses generation. As described in Section 2, this step relies on the four causal models associated to the system modes. The causal models of the modes q_1 and q_2 are given in Figure 20. The dashed arrows show that the position of the switches Switch_{K_1} and Switch_{K_2} activate or desactivate the influences they point to (e.g. if Switch_{K_1} is on, i.e. $K_1 = 1$, then there is an influence from V_0 to V_1, which happens in mode q_2). The dotted arrows represent influences that are not active in the current mode.

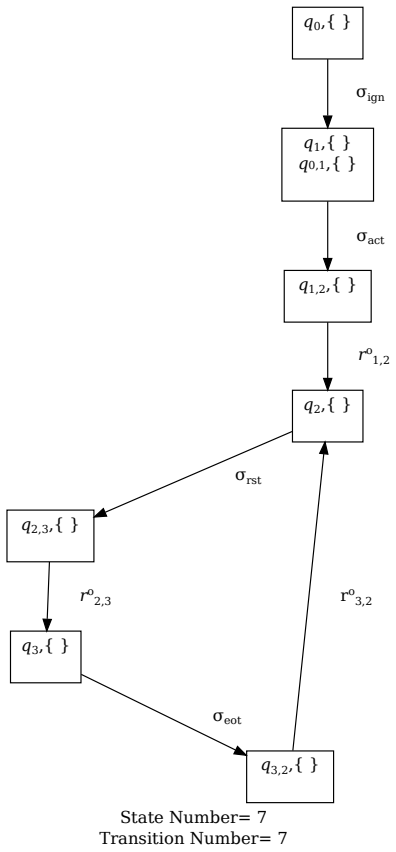

Figure 19: Partial diagnoser $PDiag(B_A^1(\Gamma))$ for $\mathcal{X}_{\text{obs}}^1 = \{V_1\}$.

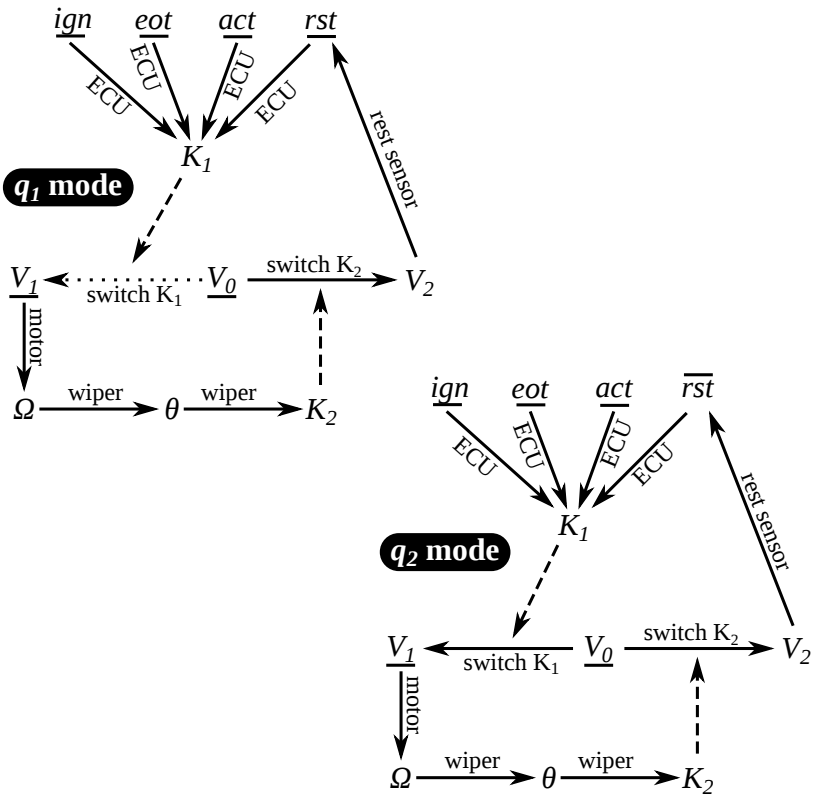

Figure 20: Causal models of q_1 and q_2 ($\underline{X}$: OK, $\overline{X}$: KO).

s_{obs} does not include σ_{rst}, which is supposed to be emitted in q_2 by the rest sensor when Switch_{K_2} switches, hence rst is labelled KO in q_2, which means that there is a conflict $\{Switch_{K_2},\ wiper,\ motor,\ rest\ sensor\}$.

Interlinking temporally the observed events σ_{ign}, σ_{act} and $r_{V_1}^{0\to 1}$ as shown in Figure 18 provides the observed (partial) signatures for every synchronized state of $PDiag(B_A^1(\Gamma))$ (cf. Figure 21). These signatures must be compared to the theoretical partial signatures of every mode as given in Table 5, from which one obtains the labeling of the corresponding vertices of the two causal models. $Q_{ref}^1 = q_2$ and in q_2, V_1 is labelled OK so the cor-

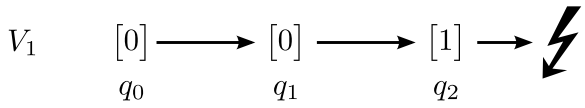

Figure 21: Observed qualitative signatures based on V_1 at iteration $k = 1$.

responding test T_{V_1} passes and does not indicate any additional R-conflict.

The R-conflict indicated by the label of rst in q_2 leads to:

$$\mathcal{A}^1 = \{Switch_{K_2},\ wiper,\ motor,\ rest\ sensor\}.$$

Step 3: Test selection. This step relies on the coverage of the tests. These are given for mode q_2 at the iteration 1, i.e. after the test T_{V_1} has passed, in Table 7. We consider the subtable defined by the components of $\mathcal{A}^1 = \{Switch_{K_2},\ wiper,\ motor,\ rest\ sensor\}$, which is the ambiguity set. We have $Q(\mathcal{A}^1) = \frac{4-1}{2} = 1.5$ and consider the possible next tests T_{V_2}, T_θ and T_Ω. These break the ambiguity set in two ambiguity groups and we have:

- $AG(T_{V_2}) = \{\mathcal{A}_1^1(T_{V_2}) = \{rest\ sensor\},$
  $\mathcal{A}_2^1(T_{V_2}) = \{Switch_{K_2}, wiper, motor\}\},$
  $Q(AG(T_{V_2}) = \frac{1}{4}\frac{1-1}{2} + \frac{3}{4}\frac{3-1}{2} = 0.75,$
  $AR(T_{V_2}) = Q(\mathcal{A}^1) - Q(AG(T_{V_2}) = 1.5 - 0.75 = 0.75;$

- $AG(T_\theta) = \{\mathcal{A}_1^1(T_\theta) = \{rest\ sensor, Switch_{K_2}, wiper\},$
  $\mathcal{A}_2^1(T_\theta) = \{wiper, motor\}\},$
  $Q(AG(T_\theta) = \frac{3}{4}\frac{2-1}{2} + \frac{2}{4}\frac{2-1}{2} = 0.625,$
  $AR(T_\theta) = Q(\mathcal{A}^1) - Q(AG(T_\theta) = 1.5 - 0.625 = 0.875;$

- $AG(T_\Omega) = \{\mathcal{A}_1^1(T_\Omega) = \{rest\ sensor, Switch_{K_2}, wiper\},$
  $\mathcal{A}_2^1(T_\Omega) = \{motor\}\},$

Table 7: Tests and their coverage for mode q_2 at iteration 1 (T_{V_1} has passed).

	T_{V_1}	T_{V_2}	T_θ	T_Ω
motor	passed	1	1	1
wiper	passed	1	1	0
switch K_2	passed	1	0	0
rest sensor	passed	0	0	0

$Q(AG(T_\Omega) = \frac{3}{4}\frac{3-1}{2} + \frac{1}{4}\frac{1-1}{2} = 0.75$,
$AR(T_\Omega) = Q(\mathcal{A}^1) - Q(AG(T_\Omega) = 1.5 - 0.75 = 0.75$.

The test that maximizes ambiguity reduction is T_θ, then the set of observed variables at iteration 2 is: $\mathcal{X}^2_{\text{obs}} = \{V_1, \theta\}$. According to Table 5, the signature-events and the partial diagnoser remain unchanged, as well as the set of possible reference modes: $PDiag(B^2_A(\Gamma)) = PDiag(B^1_A(\Gamma))$, $Q^2_{ref} = Q^1_{ref} = \{q_2\}$. The signal for θ remains flat at $\theta = 0$, which indicates that θ is KO. The new conflict in mode q_2 implies that the ambiguity set is reduced to $\mathcal{A}^2 = \{motor,\ wiper\}$.

The test selection procedure then obviously proposes the test T_Ω. Ω is also found KO, providing another conflict, and leading to $\mathcal{A}^3 = \{motor\}$, leaving the final single component diagnosis $\Delta = \{motor\}$.

8. Position of the contribution and related work

The iterative hybrid model based diagnosis method proposed in this paper refers to two research domains: hybrid model based diagnosis and test prioritization.

Hybrid models, in particular the specific class known as switched systems, represent systems that undergo abrupt changes of dynamics upon switches. They provide a particularly relevant framework for diagnosis because discrete fault occurrence can be represented by switching dynamics. Many methods have hence addressed discrete faults by adding fault modes to the nominal system model for each discrete fault (Hofbaur and Williams, 2004; Wang et al., 2007; Bayoudh et al., 2008; Benazera and Travé-Massuyès, 2009; Rienmüller et al., 2013). Other methods have addressed parametric faults like (McIlraith et al., 2000b; Narasimhan and Biswas, 2007b).

In this work, we are interested in those methods that handle the two kinds of faults in an integrated framework like (Cocquempot et al., 2004; Daigle et al., 2010; Arogeti et al., 2010; Yu et al., 2012). All these works make use of fault signatures which differ in the way they are obtained. The framework of (Cocquempot et al., 2004) is based on a hybrid automaton and makes use of analytical redundancy relations (ARRs) obtained for every automaton state, i.e. every behavioral mode. In a faulty situation, the satisfaction or violation of the set of ARRs generates a boolean indicator vector which is the signature of the fault. Mode identification relies on the concept of discernability, which uses ARRs evaluated with the continuous measured signals.

(Arogeti et al., 2010) and (Yu et al., 2012) use extensions of ARRs called Global ARRs (GARRs) (for parametric faults) and Augmented GARRs (for sensor/actuator faults) that provide a compact representation of ARRs valid for all modes. They are able to distinguish mode changes from fault occurrence thanks to a Mode-Change Signature Matrix (MCSM) which represents cause-effect relations between mode changes and GARRs.

ARRs, GARRs and AGARRs require the knowledge of the analytical ex-

pressions describing the continuous behavior of the system in every behavioral mode. In our case, we do not have this knowledge and are constrained by the fact that only the Modelica simulation models underlying the fault dictionary method and their structure, i.e. which variables are dependent and which are not, are available for use. This is why, in our method, the fault signatures are obtained from the Modelica simulation by interpreting the output signals qualitatively. This is quite similar to the qualitative signatures of Hybrid Transcend (Daigle et al., 2010). However, our method differs from (Daigle et al., 2010) in the way mode changes are addressed. (Daigle et al., 2010) extends signatures with the variables, i.e. effort and flow, associated to the junctions of the Hybrid Bond Graph used to represent the system. Like (Cocquempot et al., 2004), rather we use a hybrid automaton to represent the behavior of the hybrid system and the discrete automaton is obtained automatically from the ECU specifications. Then following (Bayoudh et al., 2008; Bayoudh and Travé-Massuyès, 2012), we enrich this automaton with so-called signature-events that signify signature changes to finally build a *partial diagnoser* following diagnosis methods for discrete-event systems (Sampath et al., 1995). Our diagnoser is *partial* because it does not account for all fault modes. It captures the set of faulty behaviors that are anticipated in the ECU specifications and for which some alerting or reaction mechanism is implemented. The partial diagnoser tracks mode changes and either identifies directly a fault mode or points at the mode(s) whose model(s) must be taken as reference to localize the fault. Fault localization is achieved by an additional procedure relying on the structure of the model in the form of a causal graph, which presents similarities with the Temporal Causal Graph of (Daigle et al., 2010) but does not require a Hybrid Bond Graph. What is also different is that the

procedure relies on the logical theory of diagnosis (Reiter, 1987) interpreted in the causal modeling framework, which allows us to address single and multiple faults in a unified way (Cordier et al., 2004).

The other domain to which our work refers is Test Prioritization that can be formulated as the problem of selecting tests or determining a proper sequence of tests. This domain has also received a lot of attention (Pattipati and Dontamsetty, 1992a,b; Dick and Faivre, 1993). Solutions to this problem can be found through heuristic optimization techniques (Li et al., 2007; Raghavan et al., 1999). Among the standard heuristics is the Information Gain, relying on entropy, which is based on a theoretical measurement of the quality of the current diagnosis, the probability of the test passing or failing, and the quality of the diagnosis if the test passes or fails (de Kleer and Williams, 1987; Pattipati and Alexandridis, 1990). The performance of the Information Gain heuristic heavily depends on the precise estimation of several parameters that are difficult to obtain and are often erroneous, like false negative test rate, and it requires costly on-line calculations based on the actual fail/pass results of previously executed tests. Based on the Information Gain heuristic, some methods, known as Diagnostic Test Prioritization Techniques, have been proposed to maximize the diagnostic information gain per test and increase the rate at which diagnosis quality improves (Gonzalez-Sanchez et al., 2011b) but they are still limited by their complexity. The gReedy diAgnostic Prioritization by ambiguiTy Reduction (RAPTOR) method (Gonzalez-Sanchez et al., 2011a) stands because it is low complexity and can be used off-line. It relies on a quite intuitive diagnosis ambiguity heuristic presented in Section 6.3. In addition to this, RAPTOR is based on concepts, in particular Tests and Test Coverage, that are easily matched to the concepts of our causal diagnosis reasoning framework.

For all these reasons, we selected RAPTOR to implement the test selection procedure.

9. Conclusion

This paper addresses the problem of diagnosing embedded functions by a test and diagnose method. The diagnosis method is designed for hybrid systems and relies on a theoretical framework that merges ideas from discrete event and continuous systems diagnosis. It does not require the availability of fault models and can be viewed as a consistency based method that complements an available fault dictionary method based on simulation (Modelica model based simulation in our case). The required additional information is the discrete dynamics automaton (underlying DES automaton), the causal structure of the reference continuous models and the abstraction of continuous signals available from the simulation of the models into qualitative values and events. The diagnosis method is coupled to a test selection procedure that determines the test expected to bring the best information for discriminating among the diagnostic hypotheses. The diagnosis ambiguity set is hence reduced in an iterative way towards the localization of the faulty component(s).

The paper reports the results of experimenting this method on the electronic function commanding the rear windscreen wiper of a car.

One of the questions often raised about the DES diagnoser method used in this paper is its complexity, as the number of states is exponential with respect to the number of states of the underlying DES automaton. In the proposed approach, the diagnoser as well as the causal graphs are generated in the design stage and used in the garage. In this context, compu-

tational or resource constraints are not an issue. On the other hand, the formalism that we use to represent hybrid systems is not a standard DES formalism as it proceeds of an aggregation in terms of modes. Particular to embedded functions is the fact that the functional decomposition generally leads to quite elementary functions whose associated subsystems' behavior shows a quite limited number of operating modes. Hence the diagnoser's complexity is drastically reduced. This does not solve however the problem of modeling all the faults and particularly multiple faults. This is just what the consistency based diagnosis approach avoids to do and this is why, among other reasons, it is interesting.

Future work could however consolidate some aspects of the method. In particular, the abstraction of the continuous signals into qualitative values that remain invariant when the system is operating within a given operating mode may not be easy in the general case. In the rear windscreen wiper case study, this operation is quite obvious and the abstraction is rather intuitive. We expect it to be the same for many embedded electronic functions, however the automatization of this step would be beneficial to the method.

Acknowledgements

This work has been developed in collaboration with the ACTIA® Group in the framework of the French OSEO-ISI AMIC-TCP. The authors want to thank Jérôme Thomas and Hervé Poulard from the ACTIA® Group for their inputs and valuable comments.

Arogeti, S. A., Wang, D., Low, C. B., 2010. Mode identification of hybrid systems in the presence of fault. Industrial Electronics, IEEE Transactions on 57 (4), 1452–1467.

Bayoudh, M., Travé-Massuyès, L., 2012. Diagnosability analysis of hybrid systems cast in a discrete-event framework. Journal of Discrete Event Dynamic Systems (JDEDS), DOI : 10.1007/s10626-012-0153-z.

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

Benazera, E., Travé-Massuyès, L., Oct. 2009. Set-theoretic estimation of hybrid system configurations. IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society 39 (5), 1277–1291.

Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., 2003. Diagnosis and fault-tolerant control. Springer Verlag.

Cocquempot, V., El Mezyani, T., Staroswiecki, M., 2004. Fault detection and isolation for hybrid systems using structured parity residuals. In: Control Conference, 2004. 5th Asian. Vol. 2. IEEE, pp. 1204–1212.

Cordier, M., Dague, P., Lévy, F., Montmain, J., Staroswiecki, M., Travé-Massuyès, L., 2004. 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 Systems, Man, and Cybernetics, Part B 34 (5), 2163–2177.

Daigle, M. J., Koutsoukos, X. D., Biswas, G., 2010. An event-based approach to integrated parametric and discrete fault diagnosis in hybrid systems. Transactions of the Institute of Measurement and Control 32 (5), 487–510.

de Kleer, J., Kurien, J., 2003. Fundamentals of model-based diagnosis. In: Proceedings of the fifth IFAC symposium on Fault Detection, Supervision, and Safety of technical Processes (Safeprocess). pp. 25–36.

de Kleer, J., Williams, B. C., 1987. Diagnosing multiple faults. Artificial Intelligence 32 (1), 97–130.

Dick, J., Faivre, A., 1993. Automating the generation and sequencing of test cases from model-based specifications. In: FME'93: Industrial-Strength Formal Methods: First International Symposium of Formal Methods Europe, Odense, Denmark, April 19-23, 1993. Proceedings. Vol. 670. Springer, p. 268.

Faure, P., Trave-Massuyes, L., Poulard, H., 1999. An interval model-based approach for optimal diagnosis tree generation. In: Proc. DX-99, 10th International Workshop on Principles of Diagnosis, Loch Awe, Scotland, 8-11 June. pp. 78–89.

Faure, P.-P., Jun. 2001. An interval model-based approach for optimal diagnosis tree generation: application to the automotive domain. Ph.D. thesis, Université Paris 13, Paris, France.

Gentil, S., Montmain, J., Combastel, C., 2004. Combining fdi and ai approaches within causal model-based diagnosis. IEEE Transactions on Systems, Man and Cybernetics 34 (5), 2207–2201.

Gonzalez-Sanchez, A., Abreu, R., Gross, H., Gemund, A. V., 2011a. Raptor: Greedy diagnostic prioritization by ambiguity group reduction. In: Proceedings of the 22nd International Workshop on Principles of Diagnosis, DX2011. pp. 84–91.

Gonzalez-Sanchez, A., E.Piel, Abreu, R., Gross, H., van Gemund, A. J., Sept 2011b. Prioritizing tests for software fault localization. Software: Practice and Experience 41 (10), 1105–1129.

Greiner, R., Smith, B. A., Wilkerson, W., 1989. A correction to the algorithm in reiter's theory of diagnosis. Artificial Intelligence 41, 79–88.

Hamscher, W., Console, L., de Kleer, J. (Eds.), 1992. Readings in Model-Based Diagnosis. Morgan Kaufmann.

Heim, B., Gentil, S., Celse, B., Cauvin, S., Travé-Massuyès, L., 2003. Fcc diagnosis using several causal and knowledge based models. In: IFAC Symposium Safeprocess. Washington, USA.

Hofbaur, M. W., Williams, B. C., October 2004. Hybrid estimation of complex systems. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 34 (5), 2178–2191.

Hopcroft, J. E., Karp, R. M., Dec. 1973. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM Journal on Computing 2 (4), 225–231.

Iwasaki, Y., Simon, H., 1986. Causality in device behaviour. Artificial intelligence 29 (1–3), 63–67.

Iwasaki, Y., Simon, H., 1994. Causality and model abstraction. Artificial intelligence 67 (1), 143–194.

Li, Z., harman, M., Hierons, R., 2007. Search algorithms for regression test case prioritization. IEEE Transactions on software engineering 33 (4), 225–237.

McIlraith, S., Biswas, G., Clancy, D., Gupta, V., 2000a. Hybrid systems diagnosis. In: N. Lynch and B. Krogh, editors, Hybrid Systems: Computation and Control, Lecture Notes in Computer Science. Springer-Verlag, pp. 282–295.

McIlraith, S., Biswas, G., Clancy, D., Gupta, V., 2000b. Hybrid systems diagnosis. In: Hybrid Systems: Computation and Control. Springer, pp. 282–295.

Narasimhan, S., Biswas, G., 2007a. Model-based diagnosis of hybrid systems. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 37 (3), 348–361.

Narasimhan, S., Biswas, G., 2007b. Model-based diagnosis of hybrid systems. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 37 (3), 348–361.

Olive, X., Travé-Massuyes, L., Thomas, J., 2003. Complementing an interval based diagnosis method with sign reasoning in the automotive domain. In: 5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS. pp. 615–620.

Pattipati, K., Alexandridis, M., 1990. Application of heuristic search and information theory to sequential fault diagnosis. IEEE Transactions on systems, man and cybernetics 20 (4), 872–887.

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

Pattipati, K., Dontamsetty, M., 1992b. On a generalized test sequencing problem. Systems, Man and Cybernetics, IEEE Transactions on 22 (2), 392–396.

Pencolé, Y., 2013. Diades: DIAgnosis of discrete-event systems. http://homepages.laas.fr/ypencole/diades/html/index.html.

Price, C. J., Snooke, N., Landry, J., 1996. Automated sneak identification. Engineering applications of artificial intelligence 9 (4), 423–427.

Raghavan, V., Shakeri, M., Pattipati, K., 1999. Optimal and near-optimal test sequencing algorithms with realistic test models. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 29 (1), 11–26.

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

Reiter, R., 1987. A theory of diagnosis from first principles. Artificial Intelligence 32 (1), 57–95.

Ressencourt, H., Travé-Massuyès, L., Thomas, J., 2006. Hierarchical modelling and diagnosis for embedded systems. In: 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'2006). Beijing, China, pp. 553–558.

Rienmüller, T., Hofbaur, M., Travé-Massuyès, L., Bayoudh, M., 2013. Mode set focused hybrid estimation. International Journal of Applied Mathematics and Computer Science 23 (1), 131–144.

Sachenbacher, M., Struss, P., 2001. Aqua: A framework for automated qualitative abstraction. In: 15th International Workshop on Qualitative Reasoning (QR-01). San Antonio, USA, pp. 971–984.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D., Sep. 1995. Disgnosability of discrete-event systems. IEEE Transactions on Automatic Control 40 (9), 1555–1575.

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

Svard, C., Nyberg, M., 2010. Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. IEEE Transactions on Systems, Man, and Cybernetics – Part A 40 (6), 1310–1328.

Travé-Massuyès, L., 2014. Bridging control and artificial intelligence theories for diagnosis: A survey. Engineering Applications of Artificial Intelligence 27, 1–16.

Travé-Massuyès, L., Calderon-Espinoza, G., jul 2007. Timed fault diagnosis. In: European Control Conference (ECC-07), Kos, Greece.

Travé-Massuyès, L., Escobet, T., Pons, R., Tornil, S., 2001. The Ca˜En diagnosis system and its automatic modelling method. Computacion i Sistemas Journal 5 (2), 128–143.

Travé-Massuyès, L., Pons, R., 1997. Causal ordering for multiple mode systems. In: 11th International Workshop on Qualitative Reasoning. Cortona, Italy, pp. 203–214.

Travé-massuyès, L., Ressencourt, H., Poulard, H., Thomas, J., aug 2013. Method for diagnosing a malfunction of a mechatronic system. http://www.freepatentsonline.com/EP2339318B1.html.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., Mar 2003. A review of process fault detection and diagnosis part II: Qualitative models and search strategies. Computers and Chemical Engineering 27 (3), 313–326.

Wang, W., Li, L., Zhou, D., Liu, K., 2007. Robust state estimation and fault diagnosis for uncertain hybrid nonlinear systems. Nonlinear analysis: Hybrid systems 1 (1), 2–15.

Yu, M., Wang, D., Luo, M., Zhang, D., Chen, Q., 2012. Fault detection, isolation and identification for hybrid systems with unknown mode changes and fault patterns. Expert Systems with Applications 39 (11), 9955–9965.