An Incremental Hybrid System Diagnoser Automaton Enhanced by Discernibility Properties

Jorge Vento, Louise Travé-Massuyès, Vicenç Puig, Ramon Sarrate

An Incremental Hybrid System Diagnoser Automaton Enhanced by Discernibility Properties

Jorge Vento, Louise Travé-Massuyès, Vicenç Puig, and Ramon Sarrate

***Abstract*—This paper proposes a method to track the system mode and diagnose a hybrid system without building an entire diagnoser off-line. The method is supported by a hybrid automaton (HA) model that represents the hybrid system continuous and discrete behavioral dynamics. This model is built on request through parallel composition of the component HA models. Diagnosis is performed by interpreting the events and measurements issued by the physical system directly on the HA model. This interpretation allows us to construct the useful parts of the diagnoser developing only the branches that are required to explain the occurrence of incoming events. The resulting diagnoser adapts to the system operational life and is much less demanding in terms of memory storage than the entire diagnoser. In addition to this feature, the proposed framework subsumes previous works in that it copes with both structural and nonstructural faults. The method is validated by the application to a case study based on the sewer network of the city of Barcelona.**

***Index Terms*—Discrete event systems, fault diagnosis, hybrid automaton (HA), hybrid systems, sewer systems.**

I. Introduction

THE majority of real systems are controlled on-line and supervised by means of automatic computer-based control systems. The behavior of these systems arises from continuous plant dynamics that can be described by continuous state variables and supervisory control that generates actuator signals at discrete-time points to change regulator set-points or the plant configuration. Diagnosing these systems is a real issue as they are subject to faults that may appear in any of the plant components, in sensors or actuators [15], [27], [29], [30].

These complex systems are modeled using hybrid models that integrate continuous and discrete dynamics. These often take the form of hybrid automaton (HA) models [22] or hybrid bond graph models [16], [27]. Then, this model can support the monitoring of the system, fault diagnosis and control tasks. Model-based online diagnosis requires quick and robust reconfiguration processes when a mode change occurs, as well as the ability to keep the nominal behavior of the system on track during transient states [12].

A HA models the real behavior of the system through a set of operation modes and a set of transitions between modes which trigger upon discrete events or based on continuous state conditions. Continuous dynamics within each mode are described by a set of algebraic differential equations which constrain the continuous state, input and output variables. Input and output variables are measured. Discrete events may be observable or unobservable. Observable events may represent commands issued by the controller or changes in state variables recorded by sensors (i.e., when a state variable crosses a threshold). Unobservable events may represent failure events or other events that cause changes in the system state not directly recorded by sensors.

This paper focuses on the HA framework and proposes a method to track the system mode and diagnose hybrid systems. It takes advantage of the methods presented in [15], [29], and [30] to propose an enhanced diagnoser that is built incrementally on-line, avoiding the construction and storage of the entire diagnoser. Indeed, diagnosis is directly performed by interpreting the events and measurements issued by the physical system on the HA model. This interpretation allows us to build incrementally the useful parts of the diagnoser, developing only the branches that are required to explain the occurrence of incoming events. Generally, a hybrid system operates in a small region compared to the entire behavioral space defined by the HA states. A significant gain can hence be expected from the proposed approach. Moreover, the proposed framework subsumes previous works in the sense that structural and nonstructural faults are considered at the same time.

The structure of this paper is the following. In Section II, a review of the previous methods to diagnose hybrid systems is presented. In Section III, the hybrid model and an overview of the proposed method are provided. In Section IV, the principles of the proposed method to diagnose faults in hybrid systems is presented. Section V presents the method to build the hybrid system diagnoser incrementally as well as its implementation. In Section VI, an application case study based on the sewer network of Barcelona is used to assess the validity of the proposed approach. Finally, the conclusion is drawn in Section VII.

Manuscript received April 3, 2014; revised August 3, 2014; accepted September 25, 2014. This work was supported in part by the Spanish Ministry of Science and Technology through the CICYT Project WATMAN under Grant DPI2009-13744 and in part by the Spanish Ministry of Economy and Competitiveness through the CICYT Project SHERECS under Grant DPI2011-26243. This paper was recommended by Associate Editor W.-K. V. Chan.

J. Vento, V. Puig, and R. Sarrate are with the Automatic Control Department, Universitat Politècnica de Catalunya, Terrassa 08222, Spain (e-mail: vicenc.puig@upc.edu).

L. Travé-Massuyès is with Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique (LAAS-CNRS), Université de Toulouse, Toulouse F-31400, France.

Digital Object Identifier 10.1109/TSMC.2014.2375158

II. Related Work

The problem of hybrid system diagnosis has been studied by several researchers. Recently in the literature, model-based techniques have been proposed to diagnose hybrid systems by both the fault detection and isolation (FDI) and artificial intelligence diagnosis community (DX) communities.[1] In the FDI approach, diagnosis is based on the use of a HA to track the system mode [6], [15], [30], combining continuous and discrete techniques to detect and isolate faults. On the other hand, in the DX approach, Daigle [16] and Narasimhan and Biswas [27] have proposed alternative ways to diagnose hybrid systems such as the hybrid bond graph formalism where, unlike for hybrid automata models, preenumeration of all system modes is avoided by generating models at runtime as mode switches occur.

In the literature, many of the methods to diagnose hybrid systems are based on multiple model filtering methods [11], [21] and particle filtering methods [17], and HA models have long been restricted to hybrid estimation schemes exemplified by [8] and [22]. Only later, hybrid diagnosis approaches combining the discrete part of the hybrid model with parity-space residuals appeared [6], [15], [31]. The method presented in these works relies on building offline a finite state machine called diagnoser [28], which is built from the hybrid model. Residuals are generated for each mode as explained in [6] and [31].

In [15] and [30], the operation modes represent nominal behavior and diagnosis focuses on fault detection and isolation of nonstructural faults, i.e., faults that do not change the structure of the model. Additive faults like sensor and actuator faults are typical nonstructural faults. A set of analytical redundancy relations (ARR) are inferred from the set of equations in each mode and they are used to generate residuals. The impact of nonstructural faults on the residuals of every mode is assumed to be known and is captured by theoretical signatures generated using the sensitivity concept [26]. Tracking the system mode involves detecting that the set of residuals of the current mode are different from zero and the set of residuals of some successor mode are zero when evaluated from the measurements.

Later, Vento *et al.* [31] proposed a method to build the diagnoser based on the behavior automaton, which is obtained accounting for the discernibility property of pairs of modes. When there is an unobservable transition from one mode to another, the transition may turn observable if the pair of modes is discernible. The discernibility property can be verified through the set of residual expressions. On the other hand, nonstructural faults can be integrated in the HA as operation modes without continuous dynamics. The transition between them may turn observable if detectability and isolability properties are fulfilled. These properties are determined through the analysis of the fault signature matrix, which is based on the sensitivity concept [25]. A fault signature matrix per mode is generated and properties are verified analyzing its columns.

[1]The FDI and DX communities are the model-based diagnosis communities in the field of automatic control and artificial intelligence, respectively.

In [6], operation modes may be nominal or faulty, leading to the capability of detecting and isolating structural faults. The faulty behavior is represented by a dynamical continuous model as for the case of nominal modes. A valve in stuck position, opened or closed, is an example of a structural fault. As in the previous case, unobservable transitions between modes may turn observable by means of residuals based on the concept of mode signature. The set of all residuals for each mode are binarized and gathered in a vector. Discernibility is guaranteed as long as signatures are different. In both cases, the resulting behavior automaton is then used to build the diagnoser, following the methodology designed in [13] and [28]. The behavior automaton abstracts the information provided by discernibility, detectability and isolability properties. It is represented by a finite state automaton including the operation modes of the HA and adding new modes and events as long as these properties are satisfied.

Recently, extensions to these methods have been proposed to improve diagnosis performances. A method based on the parameter uncertainty using a passive robust strategy can be found in [33], where an adaptive threshold for residual evaluation is generated using the equivalence between the parity space approach and input/output models. Another method proposed in [32] allows to diagnose hybrid systems using a diagnoser that reasons on components, considering nonlinear models and including multiple fault detection hypotheses.

The main issue with these approaches is that the number of states of the diagnoser grows exponentially with the number of states of the HA and it may require too much memory storage. In addition, generating the set of residuals for every mode may also be a limiting factor, although some solutions have been proposed for specific cases [7].

The proposed incremental diagnoser tries to alleviate these complexity issues by the online adaptation of the diagnoser design methodology presented in [5] and [6]. Other alternative approaches to discrete event system (DES) diagnoser design are suggested in [2], [3], [9], [19], and [20]. In [3] and [19], an online Petri net diagnoser design methodology is proposed formulating the fault diagnosis problem in terms of mathematical programming. In [2], an incremental diagnosis methodology for a special class of DES, called active systems, is proposed. Timed DES are considered in [9] and [20], adapting the diagnoser automaton approach in [28].

III. Proposed Hybrid Diagnosis Method

A. Overview of the Method

Model-based diagnosis is based on the use of a model of the system to detect and isolate faults. The estimated system behavior described by the system model is compared with the real behavior available through sensor measurements [15], [31]. In particular, FDI algorithms for hybrid systems take into account the current operation mode to generate a set of residuals, used to build consistency indicators and to achieve the diagnosis task.

A scheme of the proposed method to diagnose hybrid systems is shown in Fig. 1. The figure shows several tasks involved in online diagnosis.

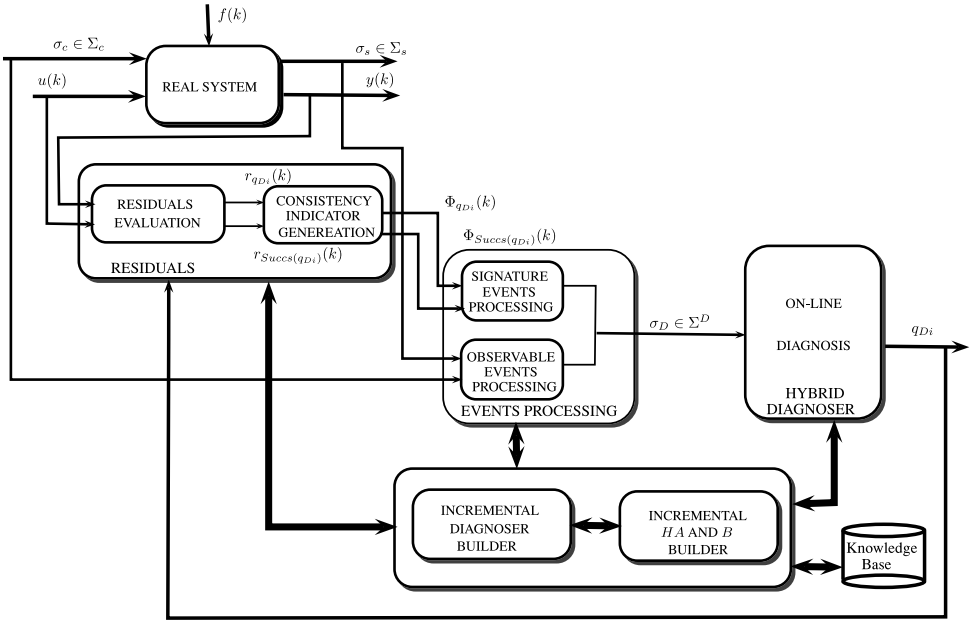

Fig. 1. Conceptual block diagram for the proposed method.

Mode tracking and diagnoser building are carried out synchronously, considering the possible current modes of the system and their successors. The original idea is to incrementally build the hybrid diagnoser when events occur. This includes building the hybrid model incrementally through the composition of the automata (AC) describing the system component behaviors. The set of linear equations constituting the continuous model of the components are parameterized as a function of the mode. The behavior automaton includes so called signature-events, that abstract the residual behaviors. Transitions labeled by unobservable events in the HA may turn observable by means of the signature-events thanks to the discernibility property (see Section IV).

Both nonstructural and structural faults are included as states of the automaton of the hybrid model. Hence, the hybrid model includes nominal operation modes, structural faulty modes, and nonstructural faulty modes. Two possibilities exist to detect and isolate faults in the system. On one hand, structural faults correspond to faulty modes with their own continuous dynamical model. Therefore, the corresponding faulty mode is recognized when its consistency indicators are in agreement with measurements. On the other hand, nonstructural faults are characterized as disturbances on the models of the other hybrid system modes. Based on the fault sensitivity to faults, a fault signature matrix can be generated. Then, a consistency test is carried out, comparing the set of observed consistency indicators with the columns of the fault signature matrix.

As a consequence of including nonstructural faults as operation modes, those faults can be detected and isolated as a mode change. As in previous approaches, consistency indicators may increase transition observability. The discernibility property has been used to predict if a mode change can be detected and identified when the operation mode is described by a continuous dynamic model [6], [15], [26]. Regarding nonstructural faults, discernibility properties are related to detectability and isolability based on the fault signature matrix [26]. All these properties are now captured in a unique and consistent form thanks to a generalization of the concept of discernibility which allows us to predict whether a faulty mode change has occurred according to the nature of the mode.

The hybrid model and the behavior automaton are recalculated whenever the system reaches a new operation mode. The incremental diagnoser builder block builds the corresponding piece of the diagnoser. Once a piece of diagnoser is built, the set of events linking the current diagnoser state with their successors are taken into account to track the system mode. Assuming that the current mode is known, the set of residuals for the current mode and their successors are generated and used in the residuals block. Hence, the residuals block computes the consistency indicators needed by the event processing block. The event processing block detects the occurrence of an observable event: an input event of the system or a signature-event generated by the residuals.

Input events are identified instantaneously and signature-events are determined by looking for those successor modes whose consistency indicators are in agreement with measurements, or checking the consistency indicators against the fault signature matrix.

The on-line diagnosis block displays messages about the current diagnoser state and the possible occurrence of a fault. The number of system modes associated with a diagnoser state depends on the hybrid system diagnosability. After an event occurrence,[2] the hybrid diagnoser traces the possible mode changes and detects and isolates possible faults. Diagnosis is based on the single fault assumption during the detection

[2]It is assumed that simultaneous events cannot occur.

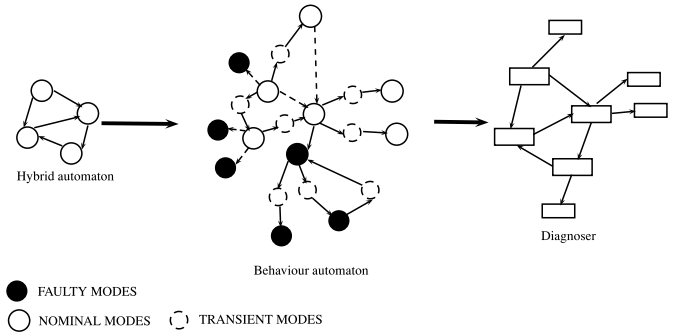

Fig. 2. Diagnoser building process.

phase. However, two faults can occur sequentially, as long as the first one corresponds to a structural fault and the second one to a nonstructural fault. Moreover, it is assumed that there is a minimal time between state transitions according to the dwell time of the HA.

Fig. 2 shows the proposed on-line diagnoser building procedure. The HA model is abstracted into a discrete-event automaton (behavior automaton) which involves the events from the model as well as signature-events built upon the continuous signals and residual generators (see the event processing block shown in Fig. 1). Next, the diagnoser is built based on the methodology proposed in [28].

Provided source and destination modes are discernible, signature-events turn observable a transition that is unobservable in the original model. In such case, a transient mode is inserted between the two discernible modes as shown in Fig. 2.

B. Hybrid Model

The system is composed by a set of components, denoted by $\mathcal{COMP}$, connected according to the system structure. We assume that the behavior of a component $\mathcal{C}_j \in \mathcal{COMP}$ is governed by linear affine equations (algebraic or differential) and parametrized with the mode. Model equations depend on a set of physical variables, which are divided in two subsets, unknown and known variables. The discrete event behavior of each component is represented by an automaton.

The HA model results from an adaptation of [5], [6], [24], and [31] but it is built incrementally. Hence, it is considered a dynamic object which is updated when a mode change is detected. The HA model results from the incremental parallel composition of the component automata and the parametrized linear equations of the system. The dependence on time is captured by indexing with the time instant k.

The incremental HA is given by $HA^k =< \mathcal{Q}^k, \mathcal{X}, \mathcal{U}, \mathcal{Y}, \mathcal{F}, \mathcal{G}^k, \mathcal{H}^k, \Sigma^k, \mathcal{T}^k >$, where:

1) $\mathcal{Q}^k$ is a set of modes. Each $q_i \in \mathcal{Q}^k$ with $|\mathcal{Q}^k| = n_q^k$ represents an operation mode, which may be a nominal mode or a structural or nonstructural faulty mode of the system i.e., $\mathcal{Q}^k = \mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s} \cup \mathcal{Q}^k_{\mathcal{F}_{\text{ns}}}$;
2) $q_0 \subseteq \mathcal{Q}^k$ is a set of initial modes;
3) $\mathcal{X} \subseteq \mathcal{R}^{n_x}$ defines the continuous state space. $\mathbf{x}(k) \in \mathcal{X}$ is the discrete-time state vector and $\mathbf{x}_0$ the initial state vector;
4) $\mathcal{U} \subseteq \mathcal{R}^{n_u}$ defines the continuous input space. $\mathbf{u}(k) \in \mathcal{U}$ is the discrete-time input vector;
5) $\mathcal{Y} \subseteq \mathcal{R}^{n_y}$ defines the continuous output space. $\mathbf{y}(k) \in \mathcal{Y}$ is the discrete-time output vector;
6) $\mathcal{F}$ is the set of faults that can be partitioned into structural and nonstructural faults, i.e., $\mathcal{F} = \mathcal{F}_s \cup \mathcal{F}_{\text{ns}}$. Every faulty mode $q_i \in \mathcal{Q}^k_{\mathcal{F}_s}$ or $q_i \in \mathcal{Q}^k_{\mathcal{F}_{\text{ns}}}$ has a corresponding fault $f_i \in \mathcal{F}_s$ or $f_i \in \mathcal{F}_{\text{ns}}$ and is associated with a fault event defined in the set $\Sigma^k_{\mathcal{F}}$. Modes associated with structural faults have a dynamic model specifying their continuous behavior, whereas those associated with nonstructural faults have not. These faults are captured by the modification of the system dynamics they imply. They are modeled by a vector $\mathbf{f}_{\text{ns}}$ impacting the equations of the other modes;
7) $\mathcal{G}^k$ defines a set of discrete-time state affine functions for each mode $q_i \in \mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$

$$\mathbf{x}(k+1) = \mathbf{A}_i\mathbf{x}(k) + \mathbf{B}_i\mathbf{u}(k) + \mathbf{F}_{xi}\mathbf{f}_{\text{ns}}(k) + \mathbf{E}_{xi} \quad (1)$$

where $\mathbf{A}_i \in \mathcal{R}^{n_x \times n_x}$, $\mathbf{B}_i \in \mathcal{R}^{n_x \times n_u}$ and $\mathbf{E}_{xi} \in \mathcal{R}^{n_x \times 1}$ are the state matrices in mode q_i, $\mathbf{f}_{\text{ns}}(k)$ is the vector representing nonstructural faults with $\mathbf{F}_{xi}$ being the fault distribution matrix. The case $\mathbf{f}_{\text{ns}}(k) = 0$ corresponds to a nominal or structural fault behavior;
8) $\mathcal{H}^k$ defines a set of discrete-time output affine functions for each mode $q_i \in \mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$

$$\mathbf{y}(k) = \mathbf{C}_i\mathbf{x}(k) + \mathbf{D}_i\mathbf{u}(k) + \mathbf{F}_{y_i}\mathbf{f}_{\text{ns}}(k) + \mathbf{E}_{y_i} \quad (2)$$

where $\mathbf{C}_i \in \mathcal{R}^{n_y \times n_x}$, $\mathbf{D}_i \in \mathcal{R}^{n_y \times n_u}$ and $\mathbf{E}_{y_i} \in \mathcal{R}^{n_y \times 1}$ are the output matrices in mode q_i and $\mathbf{F}_{y_i}$ is the fault distribution matrix;
9) $\Sigma^k = \Sigma^k_s \cup \Sigma^k_c \cup \Sigma^k_{\mathcal{F}}$ is a set of events. Spontaneous mode switching events (Σ^k_s), input events (Σ^k_c) and fault events ($\Sigma^k_{\mathcal{F}} = \Sigma^k_{\mathcal{F}_s} \cup \Sigma^k_{\mathcal{F}_{\text{ns}}}$) are considered. Σ^k can be partitioned into $\Sigma^k_o \cup \Sigma^k_{\text{uo}}$ where Σ^k_o represents a set of observable events and Σ^k_{uo} represents a set of unobservable events. $\Sigma^k_{\mathcal{F}} \subseteq \Sigma^k_{\text{uo}}$, $\Sigma^k_c \subseteq \Sigma^k_o$ and $\Sigma^k_s \subseteq \Sigma^k_{\text{uo}} \cup \Sigma^k_o$;
10) $\mathcal{T}^k : \mathcal{Q}^k \times \Sigma^k \rightarrow \mathcal{Q}^k$ is the transition function. The transition from mode q_i to mode q_j labeled with an event $\sigma \in \Sigma^k$ is denoted by $\mathcal{T}^k(q_i, \sigma) = q_j$ or by τ_{ij} when the event is of no interest.[3]

Alternatively, the model given by (1) and (2) can be expressed in input–output form using the delay operator which is denoted by p^{-1} and considering zero initial conditions, as follows:

$$\mathbf{y}(k) = \mathbf{M}_i\left(p^{-1}\right)\mathbf{u}(k) + \mathbf{\Upsilon}_i\left(p^{-1}\right)\mathbf{f}_{\text{ns}}(k) + \mathbf{E}_{mi}\left(p^{-1}\right) \quad (3)$$

where

$$\mathbf{M}_i\left(p^{-1}\right) = \mathbf{C}_i(p\mathbf{I} - \mathbf{A}_i)^{-1}\mathbf{B}_i + \mathbf{D}_i \quad (4)$$

$$\mathbf{\Upsilon}_i\left(p^{-1}\right) = \mathbf{C}_i(p\mathbf{I} - \mathbf{A}_i)^{-1}\mathbf{F}_{xi} + \mathbf{F}_{y_i} \quad (5)$$

$$\mathbf{E}_{mi}\left(p^{-1}\right) = \left(\mathbf{C}_i(p\mathbf{I} - \mathbf{A}_i)^{-1}\mathbf{E}_{xi} + \mathbf{E}_{y_i}\right)\frac{p}{p-1} \quad (6)$$

[3]It is assumed that there is only one transition from a given mode q_i to a given mode q_j.

TABLE I
TRANSITION FUNCTION DEFINED FOR THE *HA*

		Destination modes		
		$\mathcal{Q}^k_{\mathcal{N}}$	$\mathcal{Q}^k_{\mathcal{F}_s}$	$\mathcal{Q}^k_{\mathcal{F}_{ns}}$
Source modes	$\mathcal{Q}^k_{\mathcal{N}}$	$\Sigma^k_s \cup \Sigma^k_c$	$\Sigma^k_{\mathcal{F}_s}$	$\Sigma^k_{\mathcal{F}_{ns}}$
	$\mathcal{Q}^k_{\mathcal{F}_s}$	-	-	$\Sigma^k_{\mathcal{F}_{ns}}$
	$\mathcal{Q}^k_{\mathcal{F}_{ns}}$	-	-	-

where $\mathbf{M}_i(p^{-1})$ represents the system input/output transfer function, $\mathbf{\Upsilon}_i(p^{-1})$ is the nonstructural fault transfer function, and $\mathbf{E}_{mi}(p^{-1})$ is associated with the terms $\mathbf{E}_{xi}$ and $\mathbf{E}_{y_i}$ in the state space model.

The automaton for a component $\mathcal{C}$ is defined by $DA_{\mathcal{C}} =< \mathcal{Q}_{\mathcal{C}}, \Sigma_{\mathcal{C}}, \mathcal{T}_{\mathcal{C}}, \mathbf{\Gamma}_{\mathcal{C}} >$, where $\mathcal{Q}_{\mathcal{C}}$ is the set of discrete modes of the component, $\Sigma_{\mathcal{C}}$ is the set of events associated to the component automaton. These may be observable or unobservable like events corresponding to the occurrence of a structural fault. $\mathcal{T}_{\mathcal{C}}$ is the transition function and $\Gamma_{\mathcal{C}} : \mathcal{Q}_{\mathcal{C}} \to 2^{\Sigma_{\mathcal{C}}}$ is the active event function. It contains the set of all possible events $\sigma_{\mathcal{C}} \in \Sigma_{\mathcal{C}}$ such that $\mathcal{T}_{\mathcal{C}}(q_{\mathcal{C}}, \sigma_{\mathcal{C}})$ is defined.

In this paper, the hybrid model is built incrementally by combining the operation modes of the component models through the incremental parallel composition of their automata [13]. Given two automata $DA_{\mathcal{C}_1}$ and $DA_{\mathcal{C}_2}$, the parallel composition is defined as

$$DA_{\mathcal{C}_1} || DA_{\mathcal{C}_2} = Ac(\mathcal{Q}_{\mathcal{C}_1} \times \mathcal{Q}_{\mathcal{C}_2}, \Sigma_{\mathcal{C}_1} \cup \Sigma_{\mathcal{C}_2}, \mathcal{T}_{||}, \Gamma_{||}, (q_{0_1}, q_{0_2}))$$

$$\mathcal{T}_{||}((q_1, q_2), \sigma_{\mathcal{C}}) = \begin{cases} (\mathcal{T}_{\mathcal{C}_1}(q_1, \sigma_{\mathcal{C}}), \mathcal{T}_{\mathcal{C}_2}(q_2, \sigma_{\mathcal{C}})) & \text{if } \sigma_{\mathcal{C}} \in \Gamma_{\mathcal{C}_1}(q_1) \cap \Gamma_{\mathcal{C}_2}(q_2) \\ (\mathcal{T}_{\mathcal{C}_1}(q_1, \sigma_{\mathcal{C}}), q_2) & \text{if } \sigma_{\mathcal{C}} \in \Gamma_{\mathcal{C}_1}(q_1) \backslash \Sigma_{\mathcal{C}_2} \\ (q_1, \mathcal{T}_{\mathcal{C}_2}(q_2, \sigma_{\mathcal{C}})) & \text{if } \sigma_{\mathcal{C}} \in \Gamma_{\mathcal{C}_2}(q_2) \backslash \Sigma_{\mathcal{C}_1} \\ \text{undefined} & \text{otherwise} \end{cases} \tag{7}$$

where $Ac(G)$ is a unary operator that involves taking the accessible part of G from its initial state.

On the other hand, the system model is given by the sets of equations describing the component behaviors and their interconnections. The component equations are parametrized with the operation mode. The state space model of each mode in the incremental hybrid model is hence represented by (1) and (2), where state space matrices are instantiated depending on the modes obtained in the incremental composition [1], [7].

Table I summarizes when the transition function in HA^k is possibly defined. The symbol "–" indicates that the transition is not possible. Notice that transitions between nominal modes and transitions from structural faulty modes to nonstructural faulty modes are possible. Nevertheless, transitions from faulty modes to nominal modes are not possible neither any transition from nonstructural faulty modes.

Another aspect to consider is that the composition of component automata is done for operation modes that belong to $\mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$. Nonstructural faulty modes are added *a posteriori* to the resulting HA. Thus, the number of nonstructural modes associated with each mode in $\mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$ equals $|\mathcal{F}_{\text{ns}}|$.

IV. CONSISTENCY INDICATORS AND DISCERNIBILITY

A. Consistency Indicators for Diagnosis

In the hybrid framework, diagnosis is achieved both from reported observable discrete events Σ_o and continuous measurements $(\mathbf{y}(k), \mathbf{u}(k))$. Referring to the later, we adopt the common view of model-based diagnosis [10] and generate residuals for each mode associated with a dynamic model. These residuals are used to obtain consistency indicators.

Consider a mode $q_i \in \mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$ with dynamic model of (1) and (2), then the set of residuals is given by

$$\mathbf{r}_i(k) = \mathbf{y}(k) - \mathbf{G}_i\left(p^{-1}\right)\mathbf{u}(k) - \mathbf{H}_i\left(p^{-1}\right)\mathbf{y}(k) - \mathbf{E}_i\left(p^{-1}\right) \tag{8}$$

where $\mathbf{G}_i(p^{-1})$, $\mathbf{H}_i(p^{-1})$, and $\mathbf{E}_i(p^{-1})$ represent the input–output dynamic model for mode q_i. These transfer functions can be calculated using observers [26], for instance. Alternatively, the parity space approach can be also used[4] [14]. In fact, the equivalence between the two approaches has been proved under certain conditions [18]. The observer model is given by

$$\mathbf{G}_i\left(p^{-1}\right) = \mathbf{C}_i(p\mathbf{I} - \mathbf{A}_{oi})^{-1}\mathbf{B}_i + \mathbf{D}_i \tag{9}$$

$$\mathbf{H}_i\left(p^{-1}\right) = \mathbf{C}_i(p\mathbf{I} - \mathbf{A}_{oi})^{-1}\mathbf{L}_{oi} \tag{10}$$

$$\mathbf{E}_i\left(p^{-1}\right) = \left(\mathbf{C}_i(p\mathbf{I} - \mathbf{A}_{oi})^{-1}\mathbf{E}_{xi} + \mathbf{E}_{y_i}\right)\frac{p}{p-1} \tag{11}$$

where $\mathbf{A}_{oi} = \mathbf{A}_i - \mathbf{L}_{oi}\mathbf{C}_i$ and $\mathbf{L}_{oi}$ is the observer gain.

Once the residuals have been generated, they are evaluated with the measurements against a threshold, providing consistency indicators of the following form:

$$\varphi^l_i(k) = \begin{cases} 0 \text{ if } |r^l_i(k)| \le \tau^l_i \\ 1 \text{ if } |r^l_i(k)| > \tau^l_i \end{cases} \tag{12}$$

where $l \in \{1, \ldots, n_{r_i}\}$, n_{r_i} is the number of residuals for mode q_i and τ^l_i is the threshold[5] associated with residual $r^l_i(k)$. Consistency indicators are then gathered in a vector $\Phi_i(k) = [\varphi^1_i(k), \ldots, \varphi^{n_{r_i}}_i(k)]$.

To detect and isolate nonstructural faults, a theoretical fault signature matrix $\mathbf{FS}_i$ for mode q_i is generated using the concept of fault sensitivity, which is determined by the expression

$$\mathbf{\Lambda}_i\left(p^{-1}\right) = \left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{\Upsilon}_i\left(p^{-1}\right) \tag{13}$$

where $\mathbf{\Upsilon}_i$ is given by (5). Given the fault sensitivity of the jth residual with respect to the lth nonstructural fault denoted as $\Lambda_i(j, l)$ (i.e., the element (j, l) of the sensitivity matrix $\mathbf{\Lambda}_i$), the element (j, l) of $\mathbf{FS}_i$ is determined as follows:

$$\mathbf{FS}_i(j, l) = \begin{cases} 1 & \text{if } \mathbf{\Lambda}_i(j, l) \neq 0 \\ 0 & \text{if } \mathbf{\Lambda}_i(j, l) = 0. \end{cases} \tag{14}$$

$\mathbf{FS}_i(j, l)$ is 1 if the jth residual of mode q_i is sensitive to the lth fault, otherwise it is 0. For completeness one more column

[4]Any residual generation method available in the literature could be used (see [10], [23]).

[5]The thresholds can be decided using any of the standard FDI threshold generation approaches [10].

with zero signature is added representing the nonstructural fault free case. If f_l is the lth nonstructural fault, the theoretical fault signature of f_l, denoted as $\mathbf{FS}_i^{f_l}$, is then given by $\mathbf{FS}_i(\bullet, l)$.

B. Discernibility

Discernibility of two modes assesses whether these modes can be distinguished based on continuous measurements. This property is key for hybrid system mode tracking. In this section, we analyze discernibility for the general situation in which modes may be nominal or faulty and structurally or non structurally. Starting with the definition proposed by [15], we derive operational conditions based on the continuous dynamic models of the modes or on the deviations that they imply on the continuous dynamics of the hybrid system.

Definition 1: Two modes q_i and q_j are discernible iff there exists at least a couple of signals $(\mathbf{u}(k), \mathbf{y}(k))$ consistent with mode q_i that are not consistent with mode q_j and viceversa.

From the properties of residuals, we have the following result.

Proposition 1: Two modes q_i and q_j are nondiscernible iff the consistency indicators of the two modes satisfy $\Phi_i(k) = \Phi_j(k)$ for any $(\mathbf{u}(k), \mathbf{y}(k))$ and any time instant k.

Proof: According to Definition 1 two modes q_i and q_j are nondiscernible iff any couple of signals $(\mathbf{u}(k), \mathbf{y}(k))$ that are consistent with mode q_i are also consistent with mode q_j, and viceversa. Therefore, from the consistency indicator definition (12), it follows that their corresponding consistency indicator vectors $\Phi_i(k)$ and $\Phi_j(k)$ are equal. ■

We define the following function:

$$f_{\text{disc}} : \mathcal{Q}^k \times \mathcal{Q}^k \to \{0, 1\} \tag{15}$$

where $f_{\text{disc}}(q_i, q_j) = 1$ iff the two modes q_i and q_j are discernible, and $f_{\text{disc}}(q_i, q_j) = 0$ otherwise. If two modes q_i, q_j are discernible, we also say that the pair of modes (q_i, q_j) is discernible.

The following definitions are related to discernibility.

Definition 2: Considering HA^k, a mode change $q_i \to q_j$ is detectable at time instant k if q_i and q_j are discernible according to Definition 1.

Definition 3: Considering HA^k, two mode changes, $q_i \to q_j$ and $q_i \to q_l$, are isolable if the following conditions are satisfied at time instant k.

1) Both mode changes are detectable according to Definition 2, or equivalently both (q_i, q_j) and (q_i, q_l) are discernible.
2) The pair of modes (q_l, q_j) is discernible according to Definition 1.

The conditions guarantying discernibility depend on the pair of modes considered in HA^k.

Three cases can be outlined.

Case 1: Let us consider a pair of modes that have an associated continuous dynamic model of (1) and (2), represented in input–output form (3). We have the following result.

Proposition 2: Two modes $\{q_i, q_j\} \subseteq \mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$ are nondiscernible if the following conditions are fulfilled:

$$\mathbf{M}_i\left(p^{-1}\right) = \mathbf{M}_j\left(p^{-1}\right) \tag{16}$$

$$\mathbf{E}_{mi}\left(p^{-1}\right) = \mathbf{E}_{mj}\left(p^{-1}\right) \tag{17}$$

where $\mathbf{M}_i$, $\mathbf{E}_{mi}$, $\mathbf{M}_j$, and $\mathbf{E}_{mj}$ correspond to the input/output model matrices given by (4) and (6), respectively.

As derived in the following proof, (16) and (17) guarantee that consistency indicators of the two modes satisfy $\Phi_i(k) = 0$ and $\Phi_j(k) = 0$ for any $(\mathbf{u}(k), \mathbf{y}(k))$ and any time instant k, hence proving nondiscernibility of the two modes with reference to Proposition 1.

Proof: For mode i, the residual expression is given by

$$\mathbf{r}_i(k) = \left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{y}(k) - \mathbf{G}_i\left(p^{-1}\right)\mathbf{u}(k) - \mathbf{E}_i\left(p^{-1}\right). \tag{18}$$

Under no fault condition,[6] $\mathbf{r}_i(k) = 0$ if measurements $(\mathbf{u}(k), \mathbf{y}(k))$ are consistent with mode i. Therefore, the following equation holds:

$$\mathbf{y}(k) = \mathbf{M}_i\left(p^{-1}\right)\mathbf{u}(k) + \mathbf{E}_{mi}\left(p^{-1}\right). \tag{19}$$

For mode j, the residual expression is given by

$$\mathbf{r}_j(k) = \left(\mathbf{I} - \mathbf{H}_j\left(p^{-1}\right)\right)\mathbf{y}(k) - \mathbf{G}_j\left(p^{-1}\right)\mathbf{u}(k) - \mathbf{E}_j\left(p^{-1}\right). \tag{20}$$

Replacing (19) into (20) leads to

$$\begin{aligned}\mathbf{r}_{j/i}(k) = &\left(\left(\mathbf{I} - \mathbf{H}_j\left(p^{-1}\right)\right)\mathbf{M}_i\left(p^{-1}\right) - \mathbf{G}_j\left(p^{-1}\right)\right)\mathbf{u}(k) \\ &+ \left(\mathbf{I} - \mathbf{H}_j\left(p^{-1}\right)\right)\mathbf{E}_{mi}\left(p^{-1}\right) - \mathbf{E}_j\left(p^{-1}\right)\end{aligned} \tag{21}$$

which corresponds to the residual expression of mode j evaluated with measurements corresponding to mode i. Therefore, the following equalities must be satisfied in order to have a zero residual:

$$\left(\mathbf{I} - \mathbf{H}_j\left(p^{-1}\right)\right)\mathbf{M}_i\left(p^{-1}\right) = \mathbf{G}_j\left(p^{-1}\right) \tag{22}$$

$$\left(\mathbf{I} - \mathbf{H}_j\left(p^{-1}\right)\right)\mathbf{E}_{mi}\left(p^{-1}\right) = \mathbf{E}_j\left(p^{-1}\right). \tag{23}$$

Symmetrically, the residual expression of mode i evaluated with measurements corresponding to mode j leads to

$$\begin{aligned}\mathbf{r}_{i/j}(k) = &\left(\left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{M}_j\left(p^{-1}\right) - \mathbf{G}_i\left(p^{-1}\right)\right)\mathbf{u}(k) \\ &+ \left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{E}_{mj}\left(p^{-1}\right) - \mathbf{E}_i\left(p^{-1}\right)\end{aligned} \tag{24}$$

where the following equalities must be satisfied in order to have a zero residual:

$$\left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{M}_j\left(p^{-1}\right) = \mathbf{G}_i\left(p^{-1}\right) \tag{25}$$

$$\left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{E}_{mj}\left(p^{-1}\right) = \mathbf{E}_i\left(p^{-1}\right). \tag{26}$$

Therefore, the equalities (22), (23), (25), and (26) are simultaneously satisfied if the following conditions hold: $\mathbf{M}_i(p^{-1}) = \mathbf{M}_j(p^{-1})$ and $\mathbf{E}_{mi}(p^{-1}) = \mathbf{E}_{mj}(p^{-1})$. ■

The discernibility function can be evaluated using conditions (16) and (17), which rely on the system model (1) and (2) represented in input–output form (3).

[6] Without the effect of a nonstructural fault.

Case 2: Let us consider a pair of modes corresponding to nonstructural faults, that have a common predecessor mode. This mode does not have a continuous dynamic model but faults have a signature in the fault signature matrix.

The discernibility property involves comparing their corresponding fault signatures.

Proposition 3: Two modes $\{q_{i_1}, q_{i_2}\} \subseteq \mathcal{Q}^k_{\mathcal{F}_{\text{ns}}}$ associated to nonstructural faults f_{ns_1} and f_{ns_2} respectively, such that $\mathcal{T}^k(q_i, \sigma_{f_{\text{ns}_1}}) = q_{i_1}$ and $\mathcal{T}^k(q_i, \sigma_{f_{\text{ns}_2}}) = q_{i_2}$ for a given mode $q_i \in \mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$ and $\sigma_{f_{\text{ns}_1}}, \sigma_{f_{\text{ns}_2}} \in \Sigma^k_{\mathcal{F}_{\text{ns}}}$, are nondiscernible if their residual fault sensitivities satisfy

$$\Lambda_i^{f_{\text{ns}_1}}\left(p^{-1}\right) = \Lambda_i^{f_{\text{ns}_2}}\left(p^{-1}\right) \neq \mathbf{0}. \tag{27}$$

Proof: According to (14), the sensitivity to a nonstructural fault is given by a binary signature. For the two modes q_{i_1} and q_{i_2}, the signatures are $\mathbf{FS}_i^{f_{\text{ns}_1}}$ and $\mathbf{FS}_i^{f_{\text{ns}_2}}$, respectively. If the fault sensitivities are the same, then their signatures are the same too. Then, the modes q_{i_1} and q_{i_2} are nondiscernible if the following condition holds for any q_i:

$$\mathbf{FS}_i^{f_{\text{ns}_1}} \neq \mathbf{FS}_i^{f_{\text{ns}_2}} \neq \mathbf{0}. \tag{28}$$

■

Case 3: Let us consider a mode that has a continuous dynamic model and another one which has not continuous dynamic model, with a common predecessor mode. We have the following result.

Proposition 4: A mode $q_j \in \mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$ and a mode $q_{i_\alpha} \in \mathcal{Q}^k_{\mathcal{F}_{\text{ns}}}$ associated with **a** nonstructural fault f_{ns_α}, such that $\mathcal{T}^k(q_i, \sigma) = q_j$ and $\mathcal{T}^k(q_i, \sigma_{f_{\text{ns}_\alpha}}) = q_{i_\alpha}$ for a given mode $q_i \in \mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$, $\sigma \in \Sigma^k_s \cup \Sigma^k_c \cup \Sigma^k_{\mathcal{F}_s}$ and $\sigma \in \Sigma^k_{\mathcal{F}_{\text{ns}}}$, are nondiscernible if the following conditions are fulfilled:

$$\mathbf{M}_j\left(p^{-1}\right) - \mathbf{M}_i\left(p^{-1}\right) = \Lambda_i^{f_{\text{ns}_\alpha}}\left(p^{-1}\right) \tag{29}$$

$$\mathbf{E}_{mi}\left(p^{-1}\right) = \mathbf{E}_{mj}\left(p^{-1}\right) \tag{30}$$

$$\mathbf{u}(k) = \mathbf{f}_{\text{ns}_\alpha}(k). \tag{31}$$

Notice that the discernibility condition makes use of the sensitivity function of the nonstructural faulty mode calculated through the dynamic model of its predecessor mode.

Proof: This case can be deduced from cases 1 and 2. Consider the following residual expressions:

$$\begin{aligned}\mathbf{r}_{i/j}(k) = &\left(\left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{M}_j\left(p^{-1}\right) - \mathbf{G}_i\left(p^{-1}\right)\right)\mathbf{u}(k)\\ &+ \left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{E}_{mj}\left(p^{-1}\right) - \mathbf{E}_i\left(p^{-1}\right)\end{aligned} \tag{32}$$

that corresponds to the residual of mode q_i evaluated with measurements corresponding to mode q_l and

$$\begin{aligned}\mathbf{r}_{i/i_\alpha}(k) = &\left(\left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{M}_i\left(p^{-1}\right) - \mathbf{G}_i\left(p^{-1}\right)\right)\mathbf{u}(k)\\ &- \mathbf{E}_i\left(p^{-1}\right) + \left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{E}_{mi}\left(p^{-1}\right)\\ &+ \left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\boldsymbol{\Upsilon}_i\left(p^{-1}\right)\mathbf{f}_{\text{ns}_\alpha}(k)\end{aligned} \tag{33}$$

that corresponds to the residual expression of mode q_i evaluated with measurement corresponding to mode q_j under the nonstructural fault effect. Evaluating the difference $\mathbf{r}_{i/i_\alpha}(k) - \mathbf{r}_{i/j}(k)$, we obtain

$$\begin{aligned}&\left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{M}_i\left(p^{-1}\right)\mathbf{u}(k) + \left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{E}_{mi}\left(p^{-1}\right)\\ &\left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\boldsymbol{\Upsilon}_i\left(p^{-1}\right)\mathbf{f}_{\text{ns}_\alpha}(k)\\ &= \left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{M}_j\left(p^{-1}\right)\mathbf{u}(k) + \left(\mathbf{I} - \mathbf{H}_i\left(p^{-1}\right)\right)\mathbf{E}_{mj}\left(p^{-1}\right).\end{aligned}$$

Hence, the pair of modes is nondiscernible if the following conditions are satisfied:

$$\mathbf{M}_j\left(p^{-1}\right) - \mathbf{M}_i\left(p^{-1}\right) = \Lambda_i^{f_{\text{ns}_\alpha}}\left(p^{-1}\right)$$

$$\mathbf{E}_{mj}\left(p^{-1}\right) = \mathbf{E}_{mi}\left(p^{-1}\right)$$

assuming that $\mathbf{u}(k)$ and $\mathbf{f}_{\text{ns}_\alpha}(k)$ are unitary steps. ■

Algorithm 1 Incremental_HA_Builder($q_D(k)$)

```
L_h = ∅
for all q_i ∈ q_D(k) such that q_i ∈ Q^k_N ∪ Q^k_{F_s} do
    L_h = L_h ∪ {q_i}
end for
while L_h ≠ ∅ do
    L_h = L_h \ {q_i}
    for all f_w ∈ F_ns do
        Q^k := {q_{f_w i}} ∪ Q^{k-1}.
        T(q_i, σ_{f_w}) = q_{f_w i}.
    end for
    Update the model by incremental parallel composition.
    for all σ_M ∈ Γ_||(q_i) do
        T^k(q_i, σ_M) := T_||(q_i, σ_M).
        if σ_M ∉ Σ^{k-1} then
            Σ^k := σ_M ∪ Σ^{k-1}.
        end if
        if q_j ∉ Q^k then
            Q^k := {q_j} ∪ Q^{k-1}.
            Instantiate equations for this mode.
            Compute residual expression for r_j(•).
            Classify q_j into Q_disc.
            if q_j creates a new group ν_j in Q_disc then
                Compute FS_{ν_j}(•).
                Update and store in knowledge-base.
            end if
            if σ_M ∈ Σ_uo then
                if (q_i, q_j) are non-discernible according to (15)
                then
                    L_h = L_h ∪ {q_j}
                end if
            end if
        end if
    end for
end while
```

V. Hybrid Diagnosis

The incremental hybrid system diagnoser is a finite state machine built from the behavior automaton which is obtained from the incremental hybrid model HA^k, as explained in

the following subsections. It is used to achieve online diagnosis.

A. Incremental Hybrid Model Building

At any instant k, the system can be operating in one of the modes of the set called the belief mode and denoted by $q_D(k)$. Algorithm 1 takes $q_D(k)$ as input and incrementally builds the hybrid model whenever there is a change in the system, i.e., when the consistency indicators of one of the modes in the belief state change value or when an observable event occurs. HA^k is built by the parallel composition of component automata from (7) along with parametrized equations which allow one to obtain the model equations in (1) and (2) for the modes that are introduced.

As can be seen in Algorithm 1, the branches generation of HA^k depends on the discernibility property between the current mode and their successors. If some of them are nondiscernible implies that the event between them is unobservable. The iterations of the algorithm stop when HA^k is such that all branches end with an observable event avoiding uncertainty in the model. In the first iteration, HA^k initially must contain at least the initial mode and their successors, assuming they are discernible.

Line 11 of Algorithm 1 updates the discrete part of HA^k using parallel composition. The parallel composition given by (7) is adapted to generate only the successor modes of a given mode q_i. The function provides the set of successor modes, the set of events and the transition function of this iteration. The elements generated in every parallel composition are gathered in HA^k. It is assumed that the incremental initial mode (HA_{init}) is known and it is generated before the diagnosis process starts.

In Algorithm 1, lines 7–10 add the successor nonstructural faulty modes, whereas lines 13–16 add the successor nominal and structural faulty modes using the information provided by the incremental parallel composition. Lines 17–25 update the knowledge-base whenever a new mode is generated. In order to verify whether the branches of HA^k should be extended one more level further, the discernibility concerning the current mode and its successors is analyzed (see lines 26–30).

Algorithm 1 also examines conditions to recognize whether the current node has been previously considered (see line 17). Since the states of the HA have a finite number of successor states, this algorithm is guaranteed to terminate in a finite number of steps.

The system model parameterized as a function of the operation mode is composed from the whole set of equations of the components and their interconnections (see line 19 of Algorithm 1). The state space model of each mode can be represented by (34) and (35). State space matrices depend on system parameters and they are instantiated for the modes obtained in the incremental composition

$$\begin{aligned}\mathbf{x}(k+1) = {} & \mathbf{A}_i\mathbf{x}(k) + \mathbf{B}_i\mathbf{u}(k) + \mathbf{F}_{xi}\mathbf{f}(k) + \mathbf{E}_{xi} \\ & + \sum_{j=1}^{n_{S_i}} \mu_{x_i}^j S_i^j(\mathbf{x}(k),\mathbf{u}(k)) + \sum_{j=1}^{n_{D_i}} \psi_{x_i}^j D_i^j(\mathbf{x}(k),\mathbf{u}(k))\end{aligned} \tag{34}$$

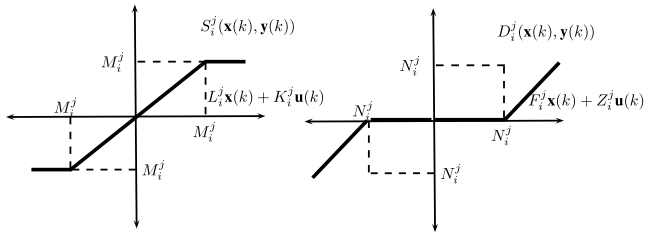

Fig. 3. Saturation and dead zone representation.

$$\begin{aligned}\mathbf{y}(k) = {} & \mathbf{C}_i\mathbf{x}(k) + \mathbf{D}_i\mathbf{u}(k) + \mathbf{F}_{y_i}\mathbf{f}(k) + \mathbf{E}_{y_i} \\ & + \sum_{j=1}^{n_{S_i}} \mu_{y_i}^j S_i^j(\mathbf{x}(k),\mathbf{u}(k)) + \sum_{j=1}^{n_{D_i}} \psi_{y_i}^j D_i^j(\mathbf{x}(k),\mathbf{u}(k)).\end{aligned} \tag{35}$$

The S_i^j and D_i^j functions model the saturation and dead zone nonlinearities that appear in the evolution and observation equations following the methodology in [7] (see Fig. 3). n_{S_i} and n_{D_i} denote the number of saturation and dead zone nonlinearities introduced by a subset of components, $\mu_{y_i}^j$ and $\psi_{y_i}^j \in \mathcal{R}^{n_y} \times \mathcal{R}$, $\mu_{x_i}^j$ and $\psi_{x_i}^j \in \mathcal{R}^{n_x} \times \mathcal{R}$

$$S_i^j(\mathbf{x}(k),\mathbf{u}(k)) = \begin{cases} -M_i^j & \text{if } L_i^j\mathbf{x}(k) + K_i^j\mathbf{u}(k) < -M_i^j \\ L_i^j\mathbf{x}(k) + K_i^j\mathbf{u}(k) & \text{if } |L_i^j\mathbf{x}(k) + K_i^j\mathbf{u}(k)| \le M_i^j \\ M_i^j & \text{if } L_i^j\mathbf{x}(k) + K_i^j\mathbf{u}(k) > M_i^j \end{cases} \tag{36}$$

where $M_i^j \in \mathcal{R}$ is a threshold, $L_i^j \in \mathcal{R} \times \mathcal{R}^{n_x}$ and $K_i^j \in \mathcal{R} \times \mathcal{R}^{n_u}$ are constant matrices

$$D_i^j(\mathbf{x}(k),\mathbf{u}(k)) = \begin{cases} F_i^j\mathbf{x}(k) + Z_i^j\mathbf{u}(k) & \text{if } \left|F_i^j\mathbf{x}(k) + Z_i^j\mathbf{u}(k)\right| \le N_i^j \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

where $N_i^j \in \mathcal{R}$ is a threshold, $F_i^j \in \mathcal{R} \times \mathcal{R}^{n_x}$ and $Z_i^j \in \mathcal{R} \times \mathcal{R}^{n_u}$ are constant matrices.

B. Incremental Behavior Automaton

The behavior automaton is a finite state generator of the language $L(HA^k)$ resulting from abstracting the continuous dynamics in terms of discrete signature-events [4], [6]. The behavior automaton is defined by $B^k =< \overline{\mathcal{Q}}^k, \overline{\Sigma}^k, \overline{T}^k, \overline{q}_0 >$.

1) $\overline{\mathcal{Q}}^k = \mathcal{Q}^k \cup \mathcal{Q}'^k$ is a set of discrete states where:
   a) $\mathcal{Q}^k$ is a set of system modes;
   b) $\mathcal{Q}'^k$ is a set of transient modes.
2) $\overline{q}_0$ is the initial state.
3) $\overline{\Sigma}^k = \Sigma^k \cup \Sigma^{\text{Sig}^k}$ is the set of events where:
   a) Σ^k is a set of system events;
   b) Σ^{Sig^k} is a set of signature-events generated when two modes are discernible according to (15).
4) $\overline{T}^k : \overline{\mathcal{Q}}^k \times \overline{\Sigma}^k \mapsto \overline{\mathcal{Q}}^k$ is the partial transition function of the behavior automaton.

In this paper, it is proposed to build B^k following Algorithm 2, which is an adaptation of the previous approach proposed in [31]. In particular, it is shown that B^k is built based

Algorithm 2 B_Builder($q_D(k)$)

1: $\mathcal{L}_h = \emptyset$.
2: **for all** $q_i \in q_D$ **do**
3: $\mathcal{L}_h = \mathcal{L}_h \cup \{q_i\}$
4: **end for**
5: **while** $\mathcal{L} \neq \emptyset$ **do**
6: $\mathcal{L}_h = \mathcal{L}_h \setminus \{q_i\}$
7: **for all** $q_j \in \text{Succs}_{HA}(q_i)$ **do**
8: **if** $q_j \notin \mathcal{Q}^k \cap \overline{\mathcal{Q}}^k$ **then**
9: $\overline{\mathcal{Q}}^k = \{q_j\} \cup \overline{\mathcal{Q}}^{k-1}$
10: **end if**
11: Let σ is such as $\mathcal{T}(q_i, \sigma) = q_j$:
12: **switch** (σ)
13: **case** $\sigma \in \Sigma_o^k$**:**
14: $\overline{\mathcal{T}}^k(q_i, \sigma) := q_j$.
15: **case** $\sigma \in \Sigma_{\text{uo}}^k$**:**
16: **if** q_i and q_j are discernible according to (15) **then**
17: $\mathcal{Q}^{tk} = \{q_{i-j}^t\} \cup \mathcal{Q}^{tk-1}$.
18: $\delta := f_{\text{Sig\_}ev}(q_i, q_j)$ according to (38).
19: **if** $\delta \notin \overline{\Sigma}^{k-1}$ **then**
20: $\overline{\Sigma}^k = \{\delta\} \cup \overline{\Sigma}^{k-1}$
21: **end if**
22: $\overline{\mathcal{T}}^k(q_i, \sigma) := q_{i-j}^t$.
23: $\overline{\mathcal{T}}^k(q_{i-j}^t, \delta) := q_j$.
24: **else**
25: **if** $q_j \in \mathcal{Q}_{\mathcal{N}}^k \cup \mathcal{Q}_{\mathcal{F}_s}^k$ **then**
26: $\mathcal{L}_h = \mathcal{L}_h \cup \{q_j\}$
27: **end if**
28: $\overline{\mathcal{T}}^k(q_i, \sigma) := q_j$.
29: **end if**
30: **end switch**
31: **end for**
32: **end while**

on the discernibility properties presented in Section IV-B. The algorithm explores HA^k taking into account only the modes in which the real system is possibly operating at time instant k.

B^k is built assuming that the system can be operating in one of the modes q_i of the belief mode $q_D(k)$. Then, an exploration of each successor mode $q_j \in \text{Succs}_{HA}(q_i)$, $q_i \in q_D$ is carried o, where $\text{Succs}_{HA}(q_i) = \{q_j \in \mathcal{Q}^k : \exists \sigma \in \Sigma^k, T^k(q_i, \sigma) = q_j\} u$.

The transitions outgoing the system modes belonging to $q_D(k)$ in HA^k are integrated into B^k and the discernibility between the source and destination modes is studied whenever necessary (see Section IV). If a transition in HA^k is labeled by an observable event the transition is kept in B^k (see lines 13–14). Otherwise, the discernibility property is evaluated between the pair of modes (q_i, q_j) (see lines 15, 16). If the two modes are discernible then a transient mode[7] is added between these modes (see line 17). The outgoing transition of the transient mode is associated with a signature-event δ (see line 18) indicating that the mode change can be observed by means of consistency indicators. Otherwise, if the two modes are nondiscernible the original transition is kept in B^k labeled with its corresponding unobservable event (see line 28).

As in the case of Algorithm 1, Algorithm 2 is guaranteed to terminate in a finite number of steps, given that the number of successor states of the behavior automaton states is finite.

The set of modes generated by the composition is partitioned into subsets of nondiscernible modes i.e., $\mathcal{Q}_{\text{disc}}^k = \mathcal{Q}_{\nu_1}^k \cup \cdots \cup \mathcal{Q}_{\nu_N}^k$. This information is stored in a knowledge base used by Algorithm 2.

The signature-event δ labeling the transition from a mode q_i to a mode q_j is indexed according to the case of discernibility of the two modes (see line 18 of Algorithm 2), according to the following function:

$$\delta = f_{\text{Sig\_}ev} : \overline{\mathcal{Q}} \times \overline{\mathcal{Q}} \to \Sigma^{\text{Sig}} \tag{38}$$

$$f_{\text{Sig\_}ev} \mapsto \begin{cases} \delta_{\nu_i - \nu_j} & \text{if } f_{\text{disc}}(q_i, q_j) = 1 \text{ according to Proposition 2, where } q_i \in \mathcal{Q}_{\nu_i}^k \text{ and } q_j \in \mathcal{Q}_{\nu_j}^k \text{ with } \mathcal{Q}_{\nu_i}^k, \mathcal{Q}_{\nu_j}^k \subseteq \mathcal{Q}_{\text{disc}}^k \\ \delta_{\mathcal{F}_{\nu_i}^\iota} & \text{if } f_{\text{disc}}(q_i, q_j) = 1 \text{ according to Proposition 3, where } \delta_{\mathcal{F}_\nu^\iota} \text{ is associated to a nonstructural fault } f_l \text{ belonging to a subset } \mathcal{F}_{\nu_i}^\iota \text{ with } \iota \in \mathcal{Z}^+ \\ \delta & \text{if } f_{\text{disc}}(q_i, q_j) = 1 \text{ according to Proposition 4.} \end{cases}$$

The event label allows for distinguishing between the discernibility cases analyzed in Section IV, so that the diagnoser can be properly built.

C. Incremental Hybrid Diagnoser

The diagnoser is a finite state machine $D^k = < \mathcal{Q}_D^k, \Sigma_D^k, T_D^k, q_{D_0} >$, where:

1) $q_{D_0} = \{q_0, \emptyset\}$ is the initial state of the diagnoser, which is assumed to correspond to a nominal system mode;
2) $\mathcal{Q}_D^k$ is a subset of the diagnoser states. An element $q_D \in \mathcal{Q}_D^k$ is a set of the form $q_D(k) = \{(q_1, l_1), (q_2, l_2), \cdots (q_n, l_n)\}$, where $q_i \in \overline{\mathcal{Q}}^k$ and $l_i \in \Delta_{\mathcal{F}}$ where $\Delta_{\mathcal{F}}$ defines the power set of fault labels $\Delta_{\mathcal{F}} = \Delta_{\mathcal{F}_s} \cup \Delta_{\mathcal{F}_{\text{ns}}}$ with $\Delta_{\mathcal{F}_s} = \{f_1, \ldots, f_\gamma\}$, and $\Delta_{\mathcal{F}_{\text{ns}}} = \{f_1^*, \ldots, f_\mu^*\}$, respectively, $\gamma + \mu$ is the total number of fault combinations and $\gamma, \mu \in \mathcal{Z}^+$. In $\Delta_{\mathcal{F}}$, $\emptyset$ represents the nominal behavior;
3) $\Sigma_D^k = \overline{\Sigma}_o^k$ is the set of all observable events in B^k;
4) $\mathcal{T}_D^k : \mathcal{Q}_D^k \times \overline{\Sigma}_o^k \mapsto \mathcal{Q}_D^k$ is a partial transition function of the diagnoser.

The transition function $\mathcal{T}_D^k$ can be calculated according to procedure described in [28] and [13], from the incremental behavior automaton B^k. According to this procedure, a diagnoser automaton is built like an observer automaton with the difference that labels reporting whether fault events have occurred are attached to the diagnoser states. The algorithm to build the transition function is executed after the occurrence of an observable event whenever there are behavior automaton

[7]The transient mode is the way to account for the HA HA^k dwell time requirement [7].

states to be introduced that have not been previously visited. The part of the diagnoser obtained takes into account only the possible successor states and hence transitions that may occur next.

D. Mode Tracking Logic

Given a set of observations of the system, a mode change can be expected if consistency indicators of the current mode have changed. The minimal time to observe that change is given by the dwell time requirement, which guarantees that residuals, and hence consistency indicators, can be properly computed [5].

The following results provide conditions for transition detection and transition identification.

Proposition 5: If $\Phi_i(k-1) = 0$ and $\Phi_i(k) \neq 0$, then a transition from $q_i \in \mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$ to another mode is suspected at time instant k.

Proof: According to the residual definition (8), $\Phi_i(k) = 0$ indicates that the system is not in mode q_i anymore, hence a transition is suspected. ■

Proposition 5 is used to decide if a mode change has occurred by monitoring the set of consistency indicators of the possible current modes, i.e., modes in the belief mode.

Proposition 6: Assuming that HA^k is in mode q_i and a transition has been suspected at time instant k according to Proposition 5, then:

1) if $\Phi_i(k) = \mathbf{FS}_i(\bullet, f_j)$ then a transition to $q_j \in \mathcal{Q}^k_{\mathcal{F}_{\text{ns}}}$ is detected at time instant k;
2) if $\Phi_j(k) = 0$ and $\mathcal{T}^k(q_i, \tau_{ij}) = q_j$, then a transition to $q_j \in \mathcal{Q}^k_{\mathcal{N}} \cup \mathcal{Q}^k_{\mathcal{F}_s}$ is detected at time instant k.

Proof: Consider the following.

1) If $\mathbf{FS}(\bullet, f_j) = \Phi_i(k)$, then the system is in mode $q_j \in \mathcal{Q}^k_{\mathcal{F}_{\text{ns}}}$ from (12)–(14).
2) If a transition from q_i has been suspected and q_j is a successor of q_i, then by the definition of residual, $\Phi_j(k) = 0$ indicates that the system has possibly transitioned to q_j.

■

Let us notice that Proposition 6 does not necessarily identify a unique mode q_j. In particular, condition 1) or 2) of Proposition 6 may be satisfied for more than one index, which corresponds to the cases of ambiguous nonstructural faulty modes and ambiguous structural faulty modes, respectively. This logic is used to identify the set of possible mode changes through Algorithm 3. Since a diagnoser state may refer to a set of modes of HA^k, in the algorithm the consistency indicators to be monitored are gathered and denoted by $\Phi_{q_{D_i}}(k)$.

E. Complexity Analysis

During the online diagnosis process, all blocks shown in Fig. 1 cooperate. A mode change triggers the occurrence of an observable or unobservable discrete event of HA^k. The diagnoser is in some state of the belief state $q_D(k)$ and waits for the occurrence of an event. If the event is identified, HA^k and B^k are updated following Algorithms 1 and 2, and D^k is extended accordingly.

The complexity of our incremental hybrid diagnosis method can be analyzed with respect to time, i.e., referring to the computation time that is required to provide a diagnosis whenever an event occurs, and with respect to space, i.e., referring to the space required to store the manipulated objects, in particular the incremental diagnoser. Generating the diagnoser in an incremental way is expected to decrease spatial complexity but increase temporal complexity (we show below that this is not actually the case).

It is well known that the standard diagnoser's spatial complexity is exponential in $O(2^{n_q})$, where n_q is the number of states of the behavior automaton. This complexity refers to the worst case, which assumes that all events are observable and that, in a given behavior automaton state, there is a set of outgoing transitions toward every other state, all labeled by the same event, for every possible event. In our incremental method, the number of modes of the behavior automaton n^k_q at some iteration k depends on the events that have been issued so far. Hence the spatial complexity of the diagnoser at iteration k is $O(2^{n^k_q})$, which is still exponential but with lower exponent since $n^k_q \leq n_q$. Assuming that the mean number of successor modes to be introduced at each iteration is s, at iteration $k+1$ the spatial complexity of the diagnoser is increased to $O(2^{n^{k+1}_q}) = O(2^{n^k_q + s})$ in the worst case, i.e., when no successor

Algorithm 3 Event_Processing($q_D(k)$)

```
loop
  wait until Φ_{q_Di}(k) ≠ 0 or σ_o ∈ Σ_o occurs
  if σ_o occurs then
    σ_D := σ_o
  else
    for all q_Dj ∈ Succs(q_Di) do
      if Φ_{q_Dj}(k) = 0 then
        COND1 := true
        break
      end if
    end for
    for all q_Dj ∈ Succs(q_Di) do
      if Φ_{q_Di}(k) = FS_{ν_i}(•, F^ι_{ν_i}) then
        COND2 := true
        break
      end if
    end for
    if COND1 = false and COND2 = false then
      print Unknown event
    else
      if COND1 and COND2 then
        σ_D := δ
      else
        if COND1 then
          σ_D := δ_{ν_i − ν_j}
        else
          σ_D := δ_{F^ι_{ν_i}}
        end if
      end if
    end if
    return
  end if
end loop
```

mode has already been visited at previous iterations. Hence, if the system was to issue all possible events and all the modes of the behavior automaton were to be visited over the successive iterations, it is clear that the spatial complexity would tend to the complexity of the global diagnoser $O(2^{n_q^k})$.

Nevertheless, our method stands on the fact that a real system never undergoes all the possible faulty modes that are anticipated in the hybrid model. From a practical point of view, controlled systems are generally designed so that the control compensates for the faults and reconfiguration policies are applied, allowing the system to run only under the presence of a very limited number of faults. This means that, although the number of possible faulty modes is theoretically high, the system really operates in a limited subset of modes and n_q^k remains of the same order of magnitude as the number of nominal operation modes n_q^{nom} even when $k \to +\infty$. This means that after a few iterations all the successors tend to have already been visited and $s \to 0$, resulting in spatial complexity remaining constant and comparable to $O(2^{n_q^{\text{nom}}})$. For instance, in the case study presented in Section VI, there are just 16 nominal modes versus 458 752 modes in total.

It is also important to notice that practical cases are generally far from the worse case and closer to the best case in which, in a given behavior automaton state, the outgoing transitions are all labeled by a different observable event (it is true for our case study). Notice that in our hybrid framework, this is related to mode discernibility. In this case, the spatial complexity of the diagnoser is $O(n_q^k)$, i.e., linear in the number of states of the behavior automaton. Furthermore, according to the above considerations, this complexity comes back to linear in the number of nominal modes, i.e., $O(n_q^{\text{nom}})$, and remains constant over iterations.

The above analysis implies that the spatial complexity of the average case $\bar{\mathcal{C}}_s$ is between the two bounds determined above, i.e., $O(n_q^{\text{nom}}) \leq \bar{\mathcal{C}}_s \leq O(2^{n_q^{\text{nom}}})$.

From the point of view of temporal complexity, it is important to notice that we are neither interested in the time required to build the (entire) diagnoser over the iterations (our method is just based on the idea that such machine does not need to be built) nor in the total time required to build the incremental diagnoser D^k over a sequence of iterations from 0 to k. For real-time purposes, we are actually interested in the time required to update the diagnosis when a new event occurs, i.e., the time required in the worst case by one iteration. In the standard case, updating the diagnosis is $O(1)$ since the diagnosis is returned in constant time by tracing the last incoming event issued by the system in the global diagnoser. In the incremental case, temporal complexity can be estimated from the number of behavioral automaton states that must be processed to extend the diagnoser, their own connectivity and their connectivity with respect to the states that have been visited in previous iterations. Time complexity is hence, in the worst case, in the order of $s(s-1)/2 + s \times n_q^k + 1$. In practice, the fact that s rapidly tends to 0 over iterations hence implies that time complexity is also rapidly $O(1)$ and that diagnosis computation is as fast as for the standard method (the same reasoning holds for the best case, which is linear).

Summarizing, the incremental method significantly reduces the online memory requirement keeping the online execution time negligible. It adapts to the actual operational life of the system and does not waste resources in considering all the theoretical mode space. In the case in which there are memory storage limitations, the incremental approach is definitively a good option.

F. Incremental Method Assessment

This section qualifies the incremental hybrid diagnosis method by assessing a set of properties. It also discusses the method in relation with diagnosability analysis.

The correctness of the method relies on the correctness of the incremental diagnoser, which comes quite obviously. Indeed, the way the incremental diagnoser is constructed for a given incoming event is the same as the one that would be used for this event if the diagnoser was completely built beforehand (off-line). Its correctness hence relies on the standard diagnoser correctness, which has been proved in [28]. More specifically, Sampath *et al.* [28] proved that the occurrence of failures in the system can be detected with a finite delay by inspecting the states of the diagnoser.

On reception of an incoming event, one run of the algorithms composing the method updates HA^k and B^k by including the set of relevant successor modes and extends the incremental diagnoser D^k accordingly. Algorithm 1 includes conditions to recognize whether the introduced modes have been previously considered (see line 17). Above all, the number of successors is finite. Hence the algorithms are guaranteed to terminate in a finite number of steps.

To be applicable, the incremental method must meet real-time requirements, which relies on the fact that the maximum time to compute the transition function of the diagnoser should not be greater than the minimum time delay between two observable events. Given the discrete-time implementation, the minimum time delay between two events, hence the maximum time to compute the transition function, is one period of the sampling time used by the computer. With our implementation of the hybrid diagnoser, the real-time constraint is met for the case study presented in the next section, where the sampling time is 300 s. However, in an application with faster dynamics, the applicability of the method must be carefully analyzed.

Let us notice that the incremental method has been devised for on-line diagnosis applications. However, it is well-known that the diagnoser can also be used at the design stage for analyzing the diagnosability of the system [28]. So, one may ask to what extend diagnosability analysis is still possible with the incremental diagnoser. Diagnosability is the ability to detect and isolate all anticipated faulty situations without ambiguity from the measurements acquired on a bounded time window. In the hybrid framework, measurements consist in continuous signals as well as discrete events. In [5], diagnosability has been formalized and studied for hybrid systems based on the same HA framework as used in this paper. This paper takes advantage of the abstraction of the continuous dynamics by a set of signature-events that preserve mode discernibility. The conditions that indicate that some faults are

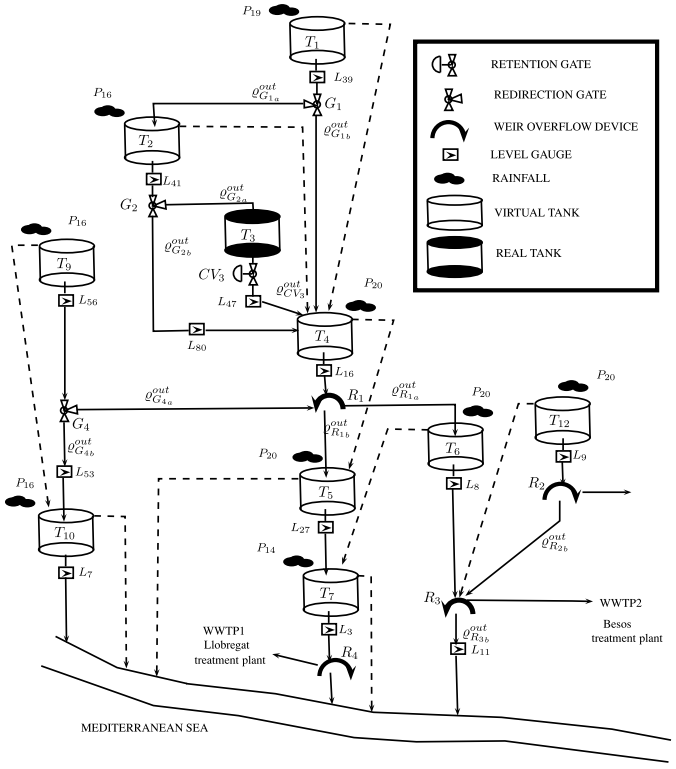

Fig. 4. Representative part of the sewer network.

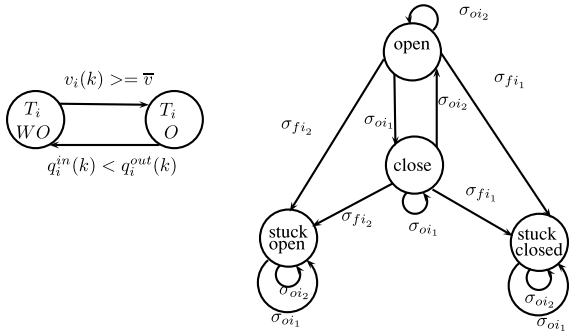

Fig. 5. Component automata.

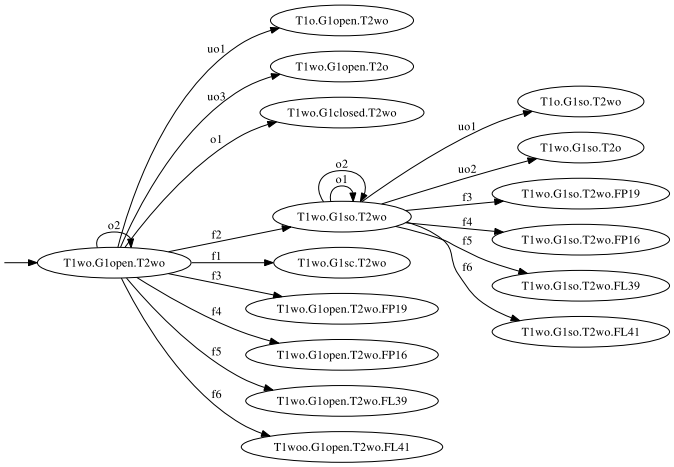

Fig. 6. Initial incremental HA HA_{init}.

nondiagnosable are then similar to the conditions for DES, i.e., the diagnoser includes an undeterminate cycle as defined in [5] and [28].

The incremental method uses a partial diagnoser at each iteration. If the system is nondiagnosable, the partial diagnoser may as well exhibit the conditions that indicate that some faults are nondiagnosable, i.e., include an undeterminate cycle. This will happen if the faulty modes already explored over previous iterations during the on-line operation are nondiagnosable. This means that the incremental diagnoser allows for local diagnosability analysis. However, full diagnosability analysis requires the global diagnoser and can hence only be achieved if all the behaviors represented in the model happen to be explored on-line, but this is just what the incremental method wants to avoid.

VI. Application Case Study

To illustrate the method, a representative part of the sewer network of Barcelona presented in [25] is used. Sewer networks present several elements exhibiting numerous operating modes depending on the sewer flows. Sewer networks may be modeled using the virtual tank modeling approach. Therefore, the decomposition of the sewer network in catchments looks like what is shown in Fig. 4. The elements that appear in the sewer are: 1) nine virtual tanks; 2) one real tank; 3) three redirection gates; 4) one retention gate; and 5) one four rain gauges to measure the rain intensity and ten limnimeters to measure the sewer level. The control gates are commanded by a controller where actions are open gate or close depending on the flow in the sewer.

A. Hybrid Modeling

A HA model can be obtained to represent the hybrid phenomena present in the network associated with the virtual tanks and the control gates. As proposed by our incremental method, the hybrid model is obtained incrementally from the automata for each component. The general automaton for a virtual tank is given by two discrete states: 1) overflow (o) and 2) nonoverflow (wo) as shown in Fig. 5 (left side). Regarding the control gates, there are four discrete states, the nominal behaviors (open or closed) and the faulty behaviors [stuck open (so) or stuck closed sc] such as shown in Fig. 5 (right side).

The elements of the sewer can be described by the set of equations below according to the component configuration. The dynamic model of the virtual tank is given by the following discrete-time equation representing the water volume:

$$T_i: \quad v_i(k+1) = v_i(k) + \Delta t(\varrho_i^{\text{in}}(k) - \varrho_i^{\text{out}}(k) - \varrho_i^{des}(k))$$

with $i \in \{0, 1\}$. The overflow is given by

$$\varrho_i^{des}(k) = \begin{cases} \varrho_i^{\text{in}}(k) - \varrho_i^{\text{out}}(k) & \text{if } v_i(k) \geq \overline{v}_i \\ 0 & \text{otherwise.} \end{cases} \tag{39}$$

The input flow associated with a virtual tank is given by

$$\varrho_i^{\text{in}} = \varrho_i^{\text{pluv}}(k) + \sum_{h=1}^{H} \varrho_i^{\text{out}_h}(k) + \sum_{l=1}^{L} \varrho_i^{des_l}(k) \tag{40}$$

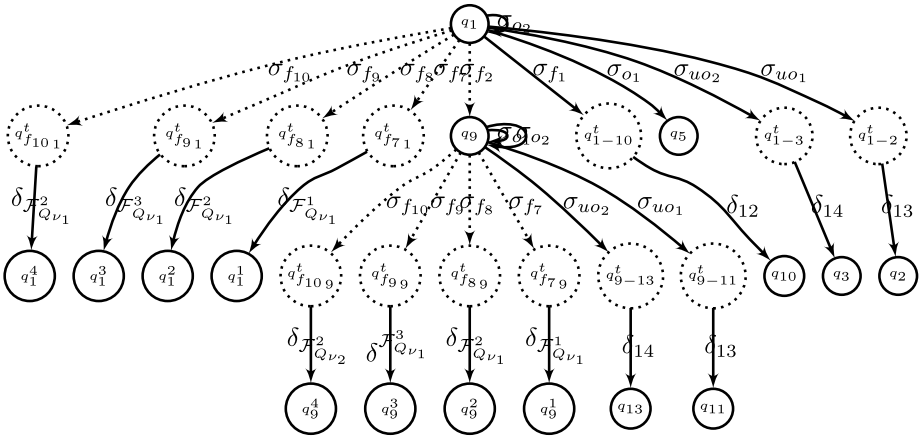

Fig. 7. Initial incremental behavior automaton B_{init}.

where $\varrho_i^{\text{pluv}}(k) = S_i\phi_i u_i(k)$ is associated with the rain intensity, $\varrho_i^{\text{out}_h}(k)$ corresponds to all the output flows of the other tanks pouring into the tank T_i and $\varrho_i^{des_l}(k)$ corresponds to all overflows pouring into the tank T_i and $h, l \in \mathcal{Z}^+$.

The output flow for every tank is given by

$$\varrho_i^{\text{out}}(k) = \begin{cases} \beta_i v_i(k) \text{ if } \varrho_i^{\text{in}}(k) < \varrho_i^{\text{out}}(k) \\ \beta_i \overline{v}_i \quad \text{if } v_i(k) \geq \overline{v}_i. \end{cases} \tag{41}$$

The relation between level and volume and the measurements provided by the sensors are described by the equations below

$$L_i(k) = \frac{\beta_i}{M_i} v_i(k). \tag{42}$$

The input flow to a control gate is divided into two output flows where the values depend on the position: 1) open ($\alpha_j = 0$) or 2) close ($\alpha_j = 1$)

$$\varrho_{G_j}^{\text{out}}(k) = \begin{cases} \varrho_{aG_j}(k) = (1-\alpha_j)\varrho_{G_j}^{\text{in}}(k) \\ \varrho_{bG_j}(k) = \alpha_j \varrho_{G_j}^{\text{in}}(k). \end{cases} \tag{43}$$

The composition is based on the automata of virtual tanks and control gates.

Notice that the tank overflow nonlinearity can be represented by a dead zone function (36), whereas the tank output flow equation can be described by a saturation function (37). In this case, the dead zone and saturation nonlinearities only depend on the tank volume. Given the system configuration, through this parametrization, a general model is obtained such that when a mode change is detected, new modes are generated, and the model is properly instantiated.

The set $\Sigma_s = \{\sigma_{\text{uo1}}, \sigma_{\text{uo2}}, \sigma_{\text{uo3}}, \sigma_{\text{uo4}}, \ldots, \sigma_{\text{uo}_{17}}, \sigma_{\text{uo}_{18}}\}$ represents the unobservable spontaneous events. Event σ_{uo1} corresponds to the volume in tank T_1 reaching its maximum, i.e., $v_1 \geq \overline{v}_1$. Event σ_{uo2} corresponds to the case in which the input flow is less than the output flow from T_1, i.e., $q_1^{\text{in}} < q_1^{\text{out}}$. The other events are related to the other virtual tanks. The set $\Sigma_{\mathcal{F}_s} = \{\sigma_{f1}, \sigma_{f2}, \sigma_{f3}, \sigma_{f4}, \sigma_{f5}, \sigma_{f6}\}$ represents the fault events related to structural faulty modes (faults in the control gates) and $\Sigma_{\mathcal{F}_{\text{ns}}} = \{\sigma_{f7}, \ldots, \sigma_{f16}, \sigma_{f17}, \ldots, \sigma_{f20}\}$ the fault events related to nonstructural faulty modes (faults in the sensors). The set $\Sigma_c = \{\sigma_{o1}, \sigma_{o2}, \sigma_{o3}, \sigma_{o4}, \sigma_{o5}, \sigma_{o6}\}$ gathers the input

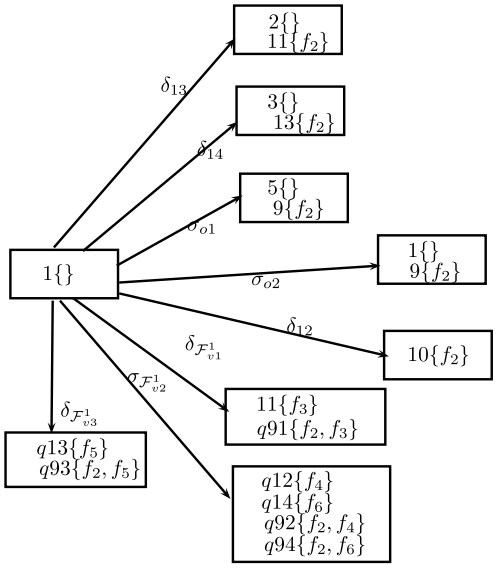

Fig. 8. Initial incremental diagnoser D_{init}.

events issued by a controller corresponding to closing or opening the valves.

In order to graphically illustrate Algorithms 1 and 2, let us consider two virtual tanks (T_1, T_2) and one control gate G_1 in the representative part of the sewer network. Applying Algorithm 1, the initial incremental hybrid model HA_{init} is obtained assuming that initially no tank is overflowing and G_1 is open (see Fig. 6). Notice that transitions labeled with f_3, f_4, f_5, and f_6 in the figure correspond to nonstructural fault events (generated by lines 7–10 of Algorithm 1). Transitions labeled with uo1, uo3, $o1$, $o2$, $f1$, and $f2$ lead to the successors modes generated by the parallel composition function at line 11 of Algorithm 1 for the initial mode. In this case, the mode labeled with $T1wo.G_1\text{so}.T1wo$ and the initial mode are nondiscernible. As mentioned before, the deepness of the HA model exploration depends on the discernibility property between the successor modes and the current mode, hence HA^k must be extended one level further. The same procedure is repeated for this successor mode.

Applying Algorithm 2, the initial incremental behavior automaton B_{init} is obtained (see Fig. 7). Modes in dashed

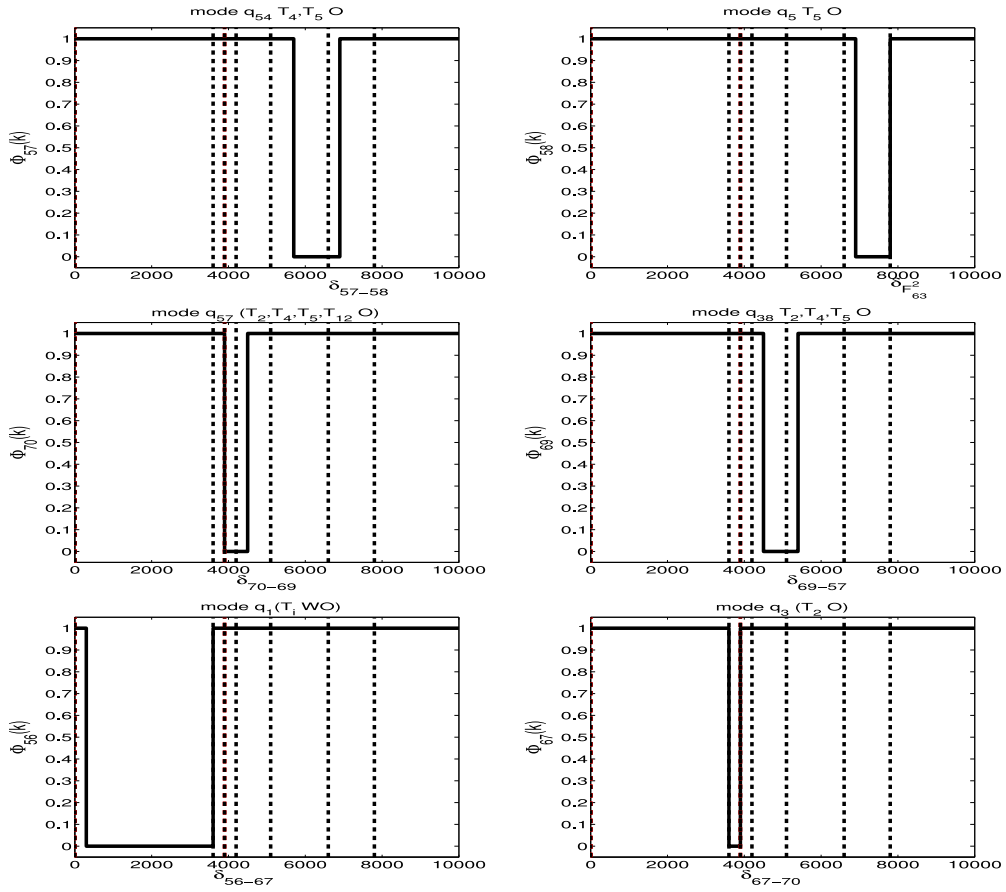

Fig. 9. Binary residuals.

line correspond to the transient modes generated evaluating the discernibility property (see lines 15–23). The generated signature-events are δ_{13}, δ_{14}, δ_{12}, $\delta_{\mathcal{F}^1_{\nu_1}}$, $\delta_{\mathcal{F}^2_{\nu_1}}$, and $\delta_{\mathcal{F}^3_{\nu_1}}$ (see line 18). Transitions in dashed line show that the destination mode is a faulty mode. Modes q_1 and q_5 are given by an observable event, therefore the transition is kept (see lines 13, 14). Modes q_1 and q_9 do not have a transient mode between them because they are nondiscernible, hence the transition is kept (see line 28). Modes labeled as q^j_i correspond to those nonstructural faulty modes where i represents its path in HA^k and j the considered nonstructural fault.

Notice that B_{init} includes the possible events that may occur. The initial diagnoser (see Fig. 8) is obtained applying the procedure mentioned in Section V-C to B_{init}.

B. Simulation Results

Considering the whole sewer, assume that system follows the mode sequence $q_1 \rightarrow q_3 \rightarrow q_{57} \rightarrow q_{38} \rightarrow q_{54} \rightarrow q_5 \rightarrow q_{585}$ with a sample time of $\Delta t = 300\ s$.

Mode q_1 refers to the situation in which no tank is in overflow. Mode q_3 refers to T_1 being in overflow. q_{57} refers to T_2, T_4, T_5, and T_{12} being in overflow. q_{38} refers to T_2, T_4, and T_5 being in overflow. Mode q_{54} refers to T_5 and T_4 being in overflow and mode q_5 refers to T_5 being in overflow. The diagnoser must track the right mode sequence and detect and isolate the possible faults from an incrementally built behavior automaton B^k.

The set of residuals are only generated for modes that are visited in HA^k. In this way, the efficient use of memory is guaranteed. There is a set of ten residuals per group using the expression given by (8).

Fig. 9 shows the set of residuals for the concerned modes in the sequence. Remark that the residuals of a given mode are consistent with measurements whenever the system remains in this mode. The signature-events identified during the simulation are shown in black vertical dashed lines in Fig. 9. Events correspond to a virtual tank reaching an overflow situation, a virtual tank leaving an overflow situation and a nonstructural fault in a sensor. These events are reported in Table II. Notice for instance that when the system is in mode q_3, $\Phi_{67}(k) = \mathbf{0}$ during the time interval [3600, 3900s] whereas the remaining consistency relations differ from zero.

Next, a nonstructural fault occurs at 7800 s, that is detected by the diagnoser. The set of consistency indicators of mode q_5 are used to isolate the fault. The observed signature is $[\,0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\,]^t$ which, according to $\mathbf{FS}_{58}$, corresponds to a fault in sensor L_{41} (see Fig. 10). Finally, the hybrid diagnoser stops and reports the diagnosis. Indeed, a nonstructural faults needs to be repaired before the diagnoser can resume.

The report given by the hybrid diagnoser is shown in Table II. The first column represents mode changes in HA^k, the second one, the identified events. The third column corresponds to the diagnoser state information and total number of states generated, the fourth one shows the total number of residuals generated. The last two columns show

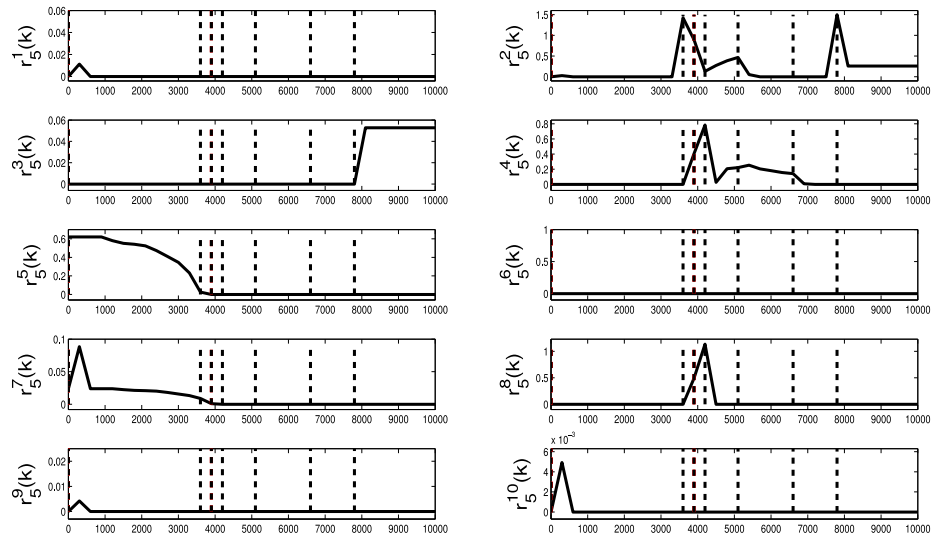

Fig. 10. Residuals of mode q_{23} when $f_{L_{41}}$ is detected.

TABLE II
HYBRID DIAGNOSER REPORT FOR THE SIMULATION SCENARIO

Mode change	Reported event	Current diagnoser state	Occurrence time (s)	Detection time (s)
Initial mode q_1	-	$(q_1,\{\})$	-	-
$q_1 \rightarrow q_3$	δ_{56-67}	$q_3,\{\}q_{21},\{f_{1b}\}$ $q_{36},\{f_{2b}\}q_{50},\{f_{3b}\}$	3600	3600
$q_3 \rightarrow q_{57}$	δ_{67-70}	$q_{57},\{\}q_{138},\{f_{1a}\}$ $q_{51},\{f_{1b}\}q_{164},\{f_{2b}\}$ $q_{74},\{f_{3a}\}$	3900	3900
$q_{57} \rightarrow q_{38}$	σ_{70-69}	$q_{38},\{\}q_{66},\{f_{1a}\}$ $q_{79},\{f_{1b}\}$	4200	4500
$q_{38} \rightarrow q_{54}$	σ_{69-57}	$q_{54},\{\}q_{48},\{f_{1b}\}$ $q_{261},\{f_{2b}\}q_{271},\{f_{3a}\}$	5100	5400
$q_{54} \rightarrow q_5$	σ_{57-58}	$q_5,\{\}q_{23},\{f_{1b}\}$ $q_{38},\{f_{2b}\}q_{52},\{f_{3a}\}$	6600	6900
$q_5 \rightarrow q_{585}$ fault in $L_{41} \in \mathcal{F}_{ns}$	$\delta_{\mathcal{F}_{58}^2}$	$q_{585},\{f_7\}q_{599},\{f_{1b}f_8\}$ $q_{610},\{f_{2b}f_8\}q_{607},\{f_{3a}f_8\}$	7800	7800

TABLE III
SEWER NETWORK COMPLEXITY FOR THE SIMULATION SCENARIO

Mode change	$\lvert\Phi^k\rvert$,$\lvert\mathcal{Q}_\nu^k\rvert$	HA^k $\lvert\mathcal{Q}_\mathcal{N}^k \cup \mathcal{Q}_{\mathcal{F}_s}^k\rvert$,$\lvert\mathcal{Q}_{\mathcal{F}_{ns}}^k\rvert$	B^k $\lvert\overline{\mathcal{Q}^k}\rvert$, ($\lvert\overline{T}^k\rvert$)	D^k $\lvert\mathcal{Q}_D^k\rvert$
Initial mode q_1	130 , 13	51 ,56	182, (199)	13
$q_1 \rightarrow q_3$	240 , 24	95,112	437, (495)	45
$q_3 \rightarrow q_{57}$	340 , 34	139,168	686, (791)	74
$q_{57} \rightarrow q_{38}$	540 ,54	189,224	1180, (1383)	102
$q_{38} \rightarrow q_{54}$	610 , 61	228,280	1340 ,(1582)	130
$q_{54} \rightarrow q_5$	690 , 69	261,336	1506, (1781)	156
$q_5 \rightarrow q_{585}$ fault in $L_{41} \in \mathcal{F}_{ns}$	690 , 69	261,336	1506, (1781)	156

the occurrence time and the detection time of the identified events.

Table III shows in detail how the incremental automata HA^k, B^k, and D^k are built when an incoming event is observed and identified. The first column shows the transitions that occur during the simulation scenario, which is described in Table II. The second column shows how the number of generated residuals increases with every mode change. Similarly, the third, fourth and fifth columns show how the number of modes of HA^k, modes and transitions of B^k and states of D^k increase with every mode change.

Table IV provides a comparison of the results obtained with the proposed method and those obtained according to the no-incremental method of [6] and [31], standing out the benefits

TABLE IV
COMPARISON BETWEEN INCREMENTAL AND NONINCREMENTAL METHOD FOR THE SIMULATION SCENARIO

	Non-incremental method	Incremental method
number of modes to be explored	32768	261
number of non-structural faulty modes to be generated	458752	336
number of diagnoser states to be computed	2^{32768}	156
number of residuals to be computed	$10 * 32768 = 327680$	690
Spatial computational complexity	Exponential $(O(2^{n_q}))$ n_q: number of global behavior automaton states	Exponential $(O(2^{n_q^{nom}}))$ / Linear $(O(n_q^{nom}))$ n_q^{nom}: number of nominal modes

of the proposed method. As can be seen, the process complexity increases with the number of operation modes. Hence, the nonincremental method could have a very high cost.

As can be seen in Table IV, the complexity of the incremental method is much lower than the complexity of the standard method and is in accordance with the discussions in Section V-F. The number of explored and generated modes remains quite tractable.

VII. CONCLUSION

A method to incrementally build online a hybrid diagnoser has been presented. The diagnoser is built whenever the system requires it after an event occurs (signature-event or input event). The method comprises the detection and isolation of structural and nonstructural faults which are modeled in the system model. The diagnoser executes the tasks of mode recognition and identification using the consistency indicators generated from a set of residuals for every mode, and then builds the part of the diagnoser required by the system's operation. Thus, the obtained diagnoser requires less memory space and can be efficiently computed online. The application of the proposed method to the Barcelona sewer network case study clearly shows its advantages compared to the off-line diagnoser generation approach.

The proposed approach could be accommodated by computing off-line the part of the diagnoser corresponding to the modes with highest probability, in particular the nominal modes, and building the rest of the diagnoser as proposed whenever it is necessary. This would achieve even better space/time complexity.

The implementation used in the application presented in this paper is totally software since the sampling time was large enough to allow real-time operation. In case of a shorter sampling time, the use of a hardware or mixed hardware/software implementation would be necessary. These alternative implementation architectures will be part of future research work.

ACKNOWLEDGMENT

The authors would like to thank Y. Pencolé for his input about DES diagnosis and for helping with and allowing use of the software DIADES that he developed.

REFERENCES

[1] A. Arogeti, D. Wang, and C. B. Low, "Mode identification of hybrid systems in the presence of fault," *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1452–1467, Apr. 2010.

[2] F. Basile, P. Chiacchio, and G. De Tommasi, "Diagnosis of a class of distributed discrete-event systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 6, pp. 731–752, Nov. 2000.

[3] F. Basile, P. Chiacchio, and G. De Tommasi, "An efficient approach for online diagnosis of discrete event systems," *IEEE Trans. Autom. Control*, vol. 54, no. 4, pp. 749–759, Apr. 2009.

[4] M. Bayoudh, "Active diagnosis of hybrid systems guided by diagnosability properties—Application to autonomous satellites," Ph.D. dissertation, Inst. Nat. Polytech., LAAS, Toulouse, France, 2009.

[5] M. Bayoudh and L. Travé-Massuyès, "Diagnosability analysis of hybrid systems cast in a discrete-event framework," *Discrete Event Dyn. Syst.*, vol. 24, no. 3, pp. 309–338, 2014.

[6] M. Bayoudh, L. Travé-Massuyès, and X. Olive, "Hybrid systems diagnosis by coupling continuous and discrete event techniques," in *Proc. 17th World Congr. Int. Fed. Autom. Control (IFAC-WC)*, Seoul, Korea, 2008, pp. 7265–7270.

[7] M. Bayoudh, L. Travé-Massuyès, and X. Olivem, "On-line analytic redundancy relations instantiation guided by component discrete-dynamics for a class of non-linear hybrid systems," in *Proc. Decis. Control Conf. (CDC/CCC)*, Shanghai, China, 2009, pp. 6970–6975.

[8] E. Benazera and L. Travé-Massuyès, "Set-theoretic estimation of hybrid system configurations," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 5, pp. 1277–1291, Jan. 2009.

[9] S. Biswas, D. Sarkar, P. Bhowal, and S. Mukhopadhyay, "Diagnosis of delay-deadline failures in real time discrete event models," *ISA Trans.*, vol. 46, no. 4, pp. 569–582, Oct. 2007.

[10] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault Tolerant Control*, 2nd ed. Berlin, Germany: Springer, 2006.

[11] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988.

[12] A. Bregon, C. Alonso, G. Biswas, B. Pulido, and N. Moya, "Fault diagnosis in hybrid systems using possible conflicts," in *Proc. 8th IFAC Symp. Fault Detection Supervision Safety Tech. Process.*, Mexico City, Mexico, 2012, pp. 132–137.

[13] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. New York, NY, USA: Springer, 2008.

[14] E. Chow and A. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Trans. Autom. Control*, vol. 29, no. 7, pp. 603–614, Jul. 1984.

[15] V. Cocquempot, M. Staroswiecki, and T. El Mezyani, "Switching time estimation and fault detection for hybrid systems using structured parity residuals," in *Proc. 15th IFAC Symp. Fault Detection Supervision Safety Tech. Process.*, Washington, DC, USA, 2003, pp. 681–686.

[16] M. Daigle, "A qualitative event-based approach to fault diagnosis of hybrid systems," Ph.D. dissertation, Faculty Graduate School Vanderbilt Institute for Software Integrated Systems (ISIS), Nashville, TN, USA, 2008.

[17] N. de Freitas, "Rao-Blackwellised particle filtering for fault diagnosis," in *Proc. IEEE Aerosp. Conf.*, vol 4. Big Sky, MT, USA, 2002, pp. 1767–1772.

[18] X. Ding, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Model Based Fault Diagnosis Techniques*. Berlin, Germany: Springer, 2008.

[19] M. Dotoli, M. P. Fanti, A. M. Mangini, and W. Ukovich, "On-line fault detection in discrete event systems by Petri nets and integer linear programming," *Automatica*, vol. 45, no. 11, pp. 2665–2672, 2009.

[20] E. Gascard and Z. Simeu-Abazi, "Modular modeling for the diagnostic of complex discrete-event systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 4, pp. 1101–1123, Oct. 2007.

[21] J.-P. Georges, D. Theilliol, V. Cocquempot, J.-C. Ponsart, and C. Aubrun, "Fault tolerance in networked control systems under intermittent observations," *Int. J. Appl. Math. Comput. Sci.*, vol. 21, no. 4, pp. 639–648, 2011.

[22] M. Hofbaur and B. Williams, "Hybrid estimation of complex systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2178–2191, Oct. 2004.

[23] M. Krysander, J. Åslund, and M. Nyberg, "An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 197–206, Jan. 2008.

[24] J. Lygeros, K. Henrik, and J. Zhang, "Dynamical properties of hybrid automata," *IEEE Trans. Autom. Control*, vol. 48, no. 1, pp. 1–17, Jan. 2003.

[25] J. Meseguer, V. Puig, and T. Escobet, "Fault diagnosis using a timed discrete event approach based on interval observers," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 5, pp. 900–916, Sep. 2010.

[26] J. Meseguer, V. Puig, and T. Escobet, "Observer gain effect in linear interval observer-based fault detection," *J. Process Control*, vol. 20, no. 8, pp. 944–956, 2010.

[27] S. Narasimhan and G. Biswas, "Model-based diagnosis of hybrid systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 3, pp. 348–361, May 2007.

[28] M. Sampath, R. Sengupta, and S. Lafortune, "Diagnosability of discrete-event system," *IEEE Trans. Autom. Control*, vol. 40, no. 9, pp. 1555–1575, Sep. 1995.

[29] L. Travé-Massuyès, M. Bayoudh, and X. Olive, "Hybrid systems diagnosis by coupling continuous and discrete event techniques," in *Proc. 17th World Congr.*, Seoul, Korea, Jul. 2008, pp. 7265–7270.

[30] J. Vento, V. Puig, and R. Sarrate, "Fault detection and isolation of hybrid system using diagnosers that combine discrete and continuous dynamics," in *Proc. Conf. Control Fault-Tolerant Syst.*, Nice, France, Oct. 2010, pp. 149–154.

[31] J. Vento, V. Puig, and R. Sarrate, "A methodology for building a fault diagnoser for hybrid systems," in *Proc. 9th Eur. Workshop Adv. Control Diagn.*, Budapest, Hungary, Nov. 2011, pp. 1–8.

[32] J. Vento, V. Puig, and R. Sarrate, "Fault detection and isolation of hybrid systems using diagnosers that reason on components," in *Proc. 8th IFAC Symp. Fault Detection Supervision Safety Tech. Processes*, Mexico City, Mexico, Aug. 2012, pp. 1250–1255.

[33] J. Vento, V. Puig, and R. Sarrate, "Parity space hybrid system diagnosis under model uncertainty," in *Proc. 20th Mediterr. Conf. Control Autom. (MED)*, Barcelona, Spain, Jul. 2012, pp. 685–690.

Jorge Vento received the system engineer, M.Sc., and Ph.D. degrees from Los Andes University, Mirida, Venezuela, and the Universitat Politécnica de Catalunya, Barcelona, Spain, in 2005, 2009, and 2014, respectively.

His current research interests include automation, control systems, and hybrid systems. He has published several papers in international conferences and journals and has been involved in academic activities as a Professor in Venezuela.

Louise Travé-Massuyès is a Research Director with the LAAS Research Laboratory, Centre National de la Recherche Scientifique, Toulouse, France, where she led the "Diagnosis and Supervisory Control" Team for several years. She has been particularly active in bridging the AI and Control Engineering Model-Based Diagnosis Communities, as a Leader of the BRIDGE Task Group of the MONET European Network. She is a Coordinator at the Maintenance and Diagnosis Strategic Field within the Aerospace Valley World Competitiveness Cluster, and serves as the Contact Evaluator for the projects submitted to the French Research Funding Agency. Her current research interests include qualitative and model-based reasoning and applications to dynamic systems, supervision, and diagnosis. She has been responsible for several industrial and European projects and has published over 250 papers in international conference proceedings and scientific journals.

Dr. Travé-Massuyès is an Editorial Board Member of the *Artificial Intelligence* Journal. She is a member of the IFAC Safeprocess Technical Committee.

Vicenç Puig has received the telecommunications engineering degree and the Ph.D. degree in control engineering, both from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1993 and 1999, respectively.

He is currently an Associate Professor of Automatic Control and a Leader of the Advanced Control Systems Research Group of the Research Center for Supervision, Safety and Automatic Control at UPC. His current research interests include fault detection and isolation of fault-tolerant control of dynamic systems. He has been involved in several European projects and networks and has published several papers in international conference proceedings and scientific journals.

Ramon Sarrate received the M.Sc. and Ph.D. degrees in industrial engineering from the Universitat Politècnica de Catalunya, Terrassa, Spain, in 1994 and 2003, respectively.

He is currently an Assistant Professor with the Department of Automatic Control, Universitat Politécnica de Catalunya. His current research interests include model-based fault diagnosis and hybrid systems. He has been involved in several national and European research projects and has published several papers in scientific journals and international conference proceedings.