

Hybrid automaton incremental construction for online diagnosis *

Jorge Vento¹, Louise Travé-Massuyès^{2,3}, Ramon Sarrate¹ and Vicenç Puig¹

¹Advanced Control Systems (SAC), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
e-mail: {jorge.isaac.vento,ramon.sarrate,vicenc.puig}@upc.edu

² CNRS, LAAS, 7, avenue du Colonel Roche, F-31400 Toulouse, France

³ University of Toulouse, LAAS, F-31400 Toulouse, France
e-mail: louise@laas.fr

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 model that represents the hybrid system continuous and discrete behavioral dynamics. Diagnosis is performed by interpreting the events and measurements issued by the physical system directly on the hybrid automaton model. This interpretation leads to building the useful parts of the diagnoser incrementally, developing only the branches that are required to explain the occurrence of the incoming events. The resulting diagnoser adapts to the system operational life and is much less demanding in terms of memory storage. The proposed framework subsumes previous works in that it copes with both structural and non-structural faults. The method is validated on an application case study based on the sewer network of the Barcelona city.

1 Introduction

The approaches to detect and isolate faults in hybrid systems have been addressed by both the FDI and DX communities. In the FDI approach, the diagnosis is based on the hybrid automaton to track the system mode [Bayoukh *et al.*, 2008; Vento *et al.*, 2011] combining the continuous and discrete techniques to detect and isolate faults. On the other hand, in the DX approach some authors have proposed alternative ways to diagnose hybrid systems like using the hybrid bond graph formalism [Narasimhan and Biswas, 2007; Daigle, 2008].

This paper follows the work presented in [Vento *et al.*, 2011; Bayoukh *et al.*, 2008] where parity-space residuals are used to track the mode and diagnose the hybrid system. In [Vento *et al.*, 2011], the operation modes represent nominal behavior and diagnosis focuses on fault detection and isolation of *non-structural faults*, i.e. faults that do not change the structure of the model (e.g. additive faults in sensors and actuators). In [Bayoukh *et al.*, 2008], the operation modes may be nominal or faulty, leading to the capability

of detecting and isolating *structural faults* (e.g. an actuator stuck at a given position, opened or closed). In both cases, a set of analytical redundancy relations (ARR) are inferred from the set of equations in each mode and they are used to generate a set of residuals. In the case of non-structural faults, the fault effect on the residuals of every mode is assumed to be known and is captured by theoretical fault signatures. Tracking the system mode involves detecting that the residuals of the current mode are different from zero and checking the theoretical fault signatures against the residuals evaluated with measurements. In the case of structural faults, fault models are assumed to be known and the residuals of a faulty mode are expected to become zero when the fault is present.

The methods presented in the above mentioned works rely on a finite state machine called a *diagnoser* [Sampath *et al.*, 1995] which is built off-line from the hybrid model and the residuals are generated for each mode as explained in [Vento *et al.*, 2011; Bayoukh *et al.*, 2008]. The main issue with these off-line approaches is that since the number of states of the diagnoser grows exponentially with the number of states of the hybrid automaton, the generation of the set of residuals for every mode may be a limiting factor.

This paper proposes a method to track the system mode and diagnose the hybrid system without building the entire diagnoser off-line. Diagnosis is performed by interpreting the events and measurements issued by the physical system directly on the hybrid automaton model. This interpretation leads to building the useful parts of the diagnoser incrementally, developing only the branches that are required to explain the occurrence of the incoming events. Generally, a hybrid system operates in a small region compared to the entire behavioral space defined by the hybrid automaton states. A significant gain hence comes from the proposed approach. Moreover, the proposed framework subsumes previous works in the sense that structural and non-structural faults are considered at the same time.

The structure of the paper is the following. In Section 2, the hybrid model is presented. Section 3 provides the principles of the proposed method to diagnose faults in hybrid systems. In Section 4, the method to incrementally build the diagnoser of the hybrid system is presented as well as its implementation. In Section 5, an application case study based on the sewer network of the Barcelona city is used to assess the validity of the proposed approach. Finally, conclusions are given in Section 6.

*This work has been funded by the Spanish Ministry of Science and Technology through the CICYT project WATMAN (ref. DPI2009-13744) and by the Spanish Ministry of Economy and Competitiveness through the CICYT project SHERECS (ref. DPI2011-26243)

2 Hybrid System Modeling

The hybrid automaton model results from an adaptation of [Lygeros *et al.*, 2003; Bayouh *et al.*, 2008; Vento *et al.*, 2011]. This work assumes linear continuous dynamics in each mode represented by discrete-time state space models. Let us consider that the model of the hybrid system to be diagnosed can be described by the following hybrid automaton $HA = \langle \mathcal{Q}, \mathcal{X}, \mathcal{U}, \mathcal{Y}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \Sigma, \mathcal{T} \rangle$, where:

- \mathcal{Q} is a set of modes. Each $q_i \in \mathcal{Q}$ with $|\mathcal{Q}| = n_q$ represents a nominal operation or faulty mode¹ of the system such that $\mathcal{Q} = \mathcal{Q}_N \cup \mathcal{Q}_{\mathcal{F}_s}$.
- $q_0 \in \mathcal{Q}$ is the initial mode.
- $\mathcal{X} \subseteq \mathcal{R}^{n_x}$ 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.
- $\mathcal{U} \subseteq \mathcal{R}^{n_u}$ defines the continuous input space. $\mathbf{u}(k) \in \mathcal{U}$ is the discrete-time input vector.
- $\mathcal{Y} \subseteq \mathcal{R}^{n_y}$ defines the continuous output space. $\mathbf{y}(k) \in \mathcal{Y}$ is the discrete-time output vector.
- \mathcal{F} is the set of faults that can be partitioned as $\mathcal{F} = \mathcal{F}_s \cup \mathcal{F}_{n_s}$ taking into account structural² and non-structural faults.
- \mathcal{G} defines a set of discrete-time state affine functions for each mode:

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

where $\mathbf{A}_i \in \mathcal{R}^{n_x \times n_x}$, $\mathbf{B}_i \in \mathcal{R}^{n_x \times n_u}$ and $\mathbf{E}_{x_i} \in \mathcal{R}^{n_x \times 1}$ are the state matrices in mode q_i , $\mathbf{f}_{n_s}(k)$ is a vector representing non-structural faults with \mathbf{F}_{x_i} being the fault distribution matrix. The case $\mathbf{f}_{n_s}(k) = 0$ corresponds to a nominal behaviour.

- \mathcal{H} defines a set of discrete-time output affine functions for each mode:

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

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.

- $\Sigma = \Sigma_s \cup \Sigma_c \cup \Sigma_{\mathcal{F}_s}$ is the set of events. Spontaneous mode switching events (Σ_s), input events (Σ_c) and structural fault events ($\Sigma_{\mathcal{F}_s}$) are considered. Σ can be partitioned as $\Sigma_o \cup \Sigma_{uo}$ where Σ_o represents the set of observable events and Σ_{uo} represents the set of unobservable events. $\Sigma_{\mathcal{F}_s} \subseteq \Sigma_{uo}$, $\Sigma_c \subseteq \Sigma_o$ and Σ_s may have elements in both sets.
- $\mathcal{T} : \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$ is the transition function. The transition from mode q_i to mode q_j labeled with an event $\sigma \in \Sigma$ is denoted by $\mathcal{T}(q_i, \sigma) = q_j$ or by t_{ij} when the event is of no interest³.

Alternatively, the model given by (1)-(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:

¹Faulty modes represent structural fault models.

²Every faulty mode $q_i \in \mathcal{Q}_{\mathcal{F}_s}$ has a corresponding structural fault $f_{q_i} \in \mathcal{F}_s$ and is associated with a fault event defined in the set Σ_f .

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

$$\mathbf{y}(k) = \mathbf{M}_i(p^{-1})\mathbf{u}(k) + \mathbf{T}_i(q^{-1})\mathbf{f}_{n_s}(k) + \mathbf{E}_{m_i}(p^{-1}) \quad (3)$$

where p^{-1} is the delay operator, $\mathbf{M}_i(p^{-1})$ represents the transfer function between inputs and outputs of the system, $\mathbf{T}_i(p^{-1})$ is the non-structural fault transfer function and $\mathbf{E}_{m_i}(p^{-1})$ is a constant term.

3 Proposed Hybrid Diagnosis Method

3.1 Principles of the Method

Model-based diagnosis is based on the use of a model of the monitored dynamic system to detect and isolate faults. The estimated system behaviour obtained from the system model is compared with the real behaviour available through sensor measurements. In particular, FDI algorithms for hybrid systems take into account which is the current operation mode to generate the set of residuals, used to build consistency indicators, and to achieve the diagnosis task [Vento *et al.*, 2011]. The scheme of the proposed method to diagnose hybrid systems is shown in Fig. 1. The figure shows the different tasks involved in online diagnosis. The original idea is to build an hybrid diagnoser in an incremental manner when an event occurs.

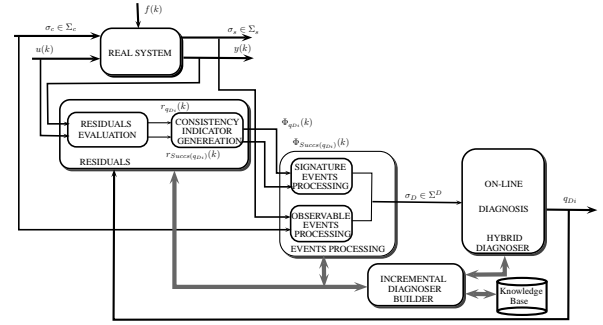


Figure 1: Conceptual block diagram for the proposed methodology

The method consists in tracking the mode sequence synchronously thanks to a diagnoser which is built incrementally from the so-called behaviour automaton, considering the possible current modes of the system and their successors. The original idea of the paper is to build the diagnoser in an incremental manner when an event occurs. The behaviour automaton includes so called signature events that abstract the residual behaviors. The transitions labelled by unobservable events in HA may hence turn observable by means of the signature events thanks to the discernability property (cf. section III.B).

To detect and isolate faults in the system two possibilities exist. On one hand, structural faults which produce changes in the dynamics are included in HA as faulty modes with their own dynamic model. Therefore, the corresponding mode is recognized when its consistency indicators are in agreement with measurements. On the other hand, non-structural faults are represented as disturbances on the models of the different modes of HA . Using the fault sensitivity of these models, a fault signature matrix can be generated. Then, a consistency test using this matrix is carried out comparing the set of observed consistency indicators with the columns of the fault signature matrix.

Diagnosis is based on the single fault assumption during the detection phase. However, two faults can occur sequentially, as long as the first one corresponds to a structural fault and the second one to a non-structural fault. Moreover, it is assumed that there is a minimal time between state transitions according to the dwell time of HA .

3.2 Diagnosis Based on Residual Consistency Indicators

Diagnosis based on continuous dynamics relies on residual properties. The set of residuals for the mode q_i is given by:

$$\mathbf{r}_i(k) = \mathbf{y}(k) - \mathbf{G}_i(p^{-1})\mathbf{u}(k) - \mathbf{H}_i(p^{-1})\mathbf{y}(k) - \mathbf{E}_i(p^{-1}) \quad (4)$$

where $\mathbf{G}_i(p^{-1})$, $\mathbf{H}_i(p^{-1})$ and $\mathbf{E}_i(p^{-1})$ can be calculated for instance, using the parity space or observer approach (for more details see f.e. [Ding *et al.*, 2008])

Once the residuals have been generated, they are evaluated with the measurements against a threshold, providing one consistency indicator of the following form for each residual:

$$\varphi_i^l(k) = \begin{cases} 0 & \text{if } |r_i^l(k)| \leq \tau_i^l \\ 1 & \text{if } |r_i^l(k)| > \tau_i^l \end{cases} \quad (5)$$

where $l \in \{1, \dots, n_{r_i}\}$, n_{r_i} is the number of residuals for mode q_i and τ_i^l is the threshold associated with the residual $r_i^l(k)$. The consistency indicators are then gathered in the vector $\Phi_{q_i}(k) = [\varphi_i^1(k), \dots, \varphi_i^{n_{r_i}}(k)]$. Summarizing, consistency indicator vector $\Phi_{q_i}(k)$ is built from the binarised residuals (5) of mode q_i evaluated with the measurements corresponding to the current mode of the system at time k .

An important property to track the system mode is discernability. Discernability between two modes is the property that assesses whether two modes can be distinguished based on continuous measurements. If two modes q_i and q_j are discernible and the system changes from mode q_i to mode q_j or viceversa, the sequence of signals $(\mathbf{u}(k), \mathbf{y}(k))$ change from being consistent with mode q_i to being consistent with mode q_j or viceversa. This property can be verified using the consistency indicators defined above [Mezyani, 2007]: two modes are discernible iff the set of consistency indicators satisfy $\Phi_{q_i}(k) \neq \mathbf{0}$ and $\Phi_{q_j}(k) = \mathbf{0}$ with measurements corresponding to mode q_j and viceversa.

If neither the consistency indicators of mode q_i nor those of mode q_j are in agreement with measurements it is assumed that a non-structural fault is affecting the system. This kind of faults are identified using the concept of fault sensitivity [Vento *et al.*, 2011], which is determined by the expression:

$$\Lambda_i(p^{-1}) = (\mathbf{I} - \mathbf{H}_i(p^{-1}))\Upsilon_i(p^{-1})$$

where $\Upsilon_i(p^{-1})$ represents the non-structural fault transfer function between the input and the non-structural fault vector in (1)-(2).

In particular, given the fault sensitivity of the j^{th} residual with respect to the l^{th} fault denoted as $\Lambda_i(j, l)$ (i.e., the element (j, l) of the sensitivity matrix Λ_i), the element (j, l) of the fault signature matrix is determined as follows:

$$\mathbf{FS}_i(j, l) = \begin{cases} 1 & \text{if } \Lambda_i(j, l)(p^{-1}) \neq 0 \\ 0 & \text{if } \Lambda_i(j, l)(p^{-1}) = 0 \end{cases} \quad (6)$$

i.e., if the j^{th} residual in mode q_i depends on the l^{th} fault, it is coded as a 1 and it is coded as a 0 otherwise. A non-structural faulty situation is detectable, i.e. discernible from the nominal mode, if its signature is different from 0. Two faulty situations corresponding to the l^{th} and the l'^{th} non-structural faults are discernible iff their signatures are different, i.e. $\mathbf{FS}_i(\bullet, l) \neq \mathbf{FS}_i(\bullet, l')$. Since in a hybrid system, residuals change with the mode, the fault sensitivity as well as the theoretical fault signature matrix depend on the mode.

If the situation is such that neither a structural fault nor a non-structural fault can be isolated, the system mode is assumed unknown. The diagnosis based on consistency indicators assumes that the residual dynamics have time to establish between two consecutive transitions.

4 Hybrid Diagnosis

The diagnoser of the hybrid system is a finite state machine built from the behavior automaton and used, on one hand, to perform on-line diagnosis and on the other hand, to check the diagnosability of the hybrid system as presented in [Bayouh *et al.*, 2008]. The method proposes to incrementally build the hybrid diagnoser from the behaviour automaton obtained while the system is monitored.

4.1 Behaviour Automaton

The behaviour automaton is the finite state generator of the abstract language $L(HA)$ resulting from abstracting the continuous dynamics captured by the residual consistency indicators in terms of discrete signature-events [Bayouh *et al.*, 2008]. The behaviour automaton is defined by $B = \langle \overline{Q}, \overline{\Sigma}, \overline{\mathcal{T}}, \overline{q}_0 \rangle$.

- $\overline{Q} = Q \cup Q_t \cup Q_{\mathcal{F}_{n.s}}^t \cup Q_{\mathcal{F}_{n.s}}$ is the set of discrete states where:
 - Q is the set of system modes,
 - Q^t is the set of transient modes between two discernible modes in HA ,
 - $Q_{\mathcal{F}_{n.s}}^t$ is the set of transient modes to represent a non-structural fault occurrence,
 - $Q_{\mathcal{F}_{n.s}}$ is the set of modes representing non-structural fault behaviours.
- \overline{q}_0 is the initial state,
- $\overline{\Sigma} = \Sigma \cup \Sigma^{Sig} \cup \Sigma_{\mathcal{F}_{n.s}} \cup \Sigma_{\mathcal{F}_{n.s}}^{Sig}$ is the set of events where:
 - Σ is the set of system events,
 - Σ^{Sig} is the set of signature-events generated by function $f_{Sig_{ev}}$ defined by (7),
 - $\Sigma_{\mathcal{F}_{n.s}}$ is the set of fault events related to a non-structural fault occurrence,
 - $\Sigma_{\mathcal{F}_{n.s}}^{Sig}$ is the set of signature-events for non-structural faults generated by function $f_{Sig_{ev}}$ defined by (7),
- $\overline{\mathcal{T}} : \overline{Q} \times \overline{\Sigma} \mapsto \overline{Q}$ is the partial transition function of the behaviour automaton.

In this paper, it is proposed to build B incrementally following Algorithm 1, which is an adaptation of the previous approach proposed in [Vento *et al.*, 2011]. The algorithm explores HA taking into account only the modes in which the real system is possibly operating at time instant k .

Assuming that the system is possibly operating in a given mode or set of modes denoted by q_D , to build incrementally

Algorithm 1 $B_Builder(q_D)$

```

1: Create a queue  $\mathcal{L}$ .
2: for all  $q_i \in q_D$  do
3:   Enqueue  $q_i$  onto  $\mathcal{L}$ 
4: end for
5: while  $\mathcal{L}$  is not empty do
6:    $q_i := \text{dequeue } \mathcal{L}$ 
7:   for all  $q_j \in Succs(q_i)$  do
8:     if  $q_j \notin \mathcal{Q} \cap \overline{\mathcal{Q}}$  then
9:        $\overline{\mathcal{Q}} = \{q_j\} \cup \overline{\mathcal{Q}}$ 
10:      Compute residual expression  $r_j(\bullet)$ 
11:      Classify  $q_j$  into  $\mathcal{Q}_{disc}$ .
12:      if  $q_j$  creates a new group  $\nu$  in  $\mathcal{Q}_{disc}$  then
13:        Calculate  $FS_{\nu_j}$  associated to  $\mathcal{F}_{ns}$ 
14:        Determine the subsets of detectable faults  $\mathcal{F}_{\nu_j}^*$ .
15:        Determine the set of non-detectable faults  $\mathcal{F}'_{\nu_j}$ 
16:      end if
17:      end if
18:      Define  $\sigma$  such that  $\mathcal{T}(q_i, \sigma) = q_j$  :
19:      if  $\sigma \notin \overline{\Sigma}$  then
20:         $\overline{\Sigma} = \{\sigma\} \cup \overline{\Sigma}$ 
21:      end if
22:      if  $\sigma \in \Sigma_o$  then
23:         $\overline{\mathcal{T}}(q_i, \sigma) := q_j$ .
24:      else
25:        if  $q_i$  and  $q_j$  are discernible then
26:           $\mathcal{Q}^t = \{q_{i-j}^t\} \cup \mathcal{Q}^t$ 
27:          if  $\delta_{\nu_i-\nu_j} \notin \Sigma^{Sig}$  then
28:             $\Sigma^{Sig} = \{\delta_{\nu_i-\nu_j}\} \cup \Sigma^{Sig}$ 
29:          end if
30:           $\overline{\mathcal{T}}(q_i, \sigma) := q_{i-j}^t$ .
31:           $\overline{\mathcal{T}}(q_{i-j}^t, \delta_{\nu_i-\nu_j}) := q_j$ .
32:        else
33:          Enqueue  $q_j$  onto  $\mathcal{L}$ 
34:           $\overline{\mathcal{T}}(q_i, \sigma) := q_j$ .
35:        end if
36:      end if
37:    end for
38:    for all  $f_l \in \mathcal{F}_{ns}$  do
39:       $\overline{\mathcal{Q}} = \{q_{i-f_l}^t\} \cup \overline{\mathcal{Q}}$ 
40:      if  $\sigma_{f_l} \notin \overline{\Sigma_{\mathcal{F}_{ns}}}$  then
41:         $\overline{\Sigma_{\mathcal{F}_{ns}}} = \{\sigma_{f_l}\} \cup \overline{\Sigma_{\mathcal{F}_{ns}}}$ 
42:      end if
43:      if  $f_l \notin \mathcal{F}_{\nu_i}$  then
44:         $\overline{\mathcal{Q}} = \{q_{i-f_l}^t\} \cup \overline{\mathcal{Q}}$ 
45:        if  $\delta_{\mathcal{F}_{\nu_i}}^* \notin \Sigma_{\mathcal{F}_{ns}}^{Sig}$  then
46:           $\Sigma_{\mathcal{F}_{ns}}^{Sig} = \{\delta_{\mathcal{F}_{\nu_i}}^*\} \cup \Sigma_{\mathcal{F}_{ns}}^{Sig}$ 
47:        end if
48:         $\overline{\mathcal{T}}(q, \sigma_{f_l}) := q_{i-f_l}^t$ .
49:         $\overline{\mathcal{T}}(q_{i-f_l}^t, \delta_{\mathcal{F}_{\nu_i}}^*) := q_{i-f_l}^t$ .
50:      else
51:         $\overline{\mathcal{T}}(q_i, \sigma_{f_l}) := q_{i-f_l}^t$ .
52:      end if
53:    end for
54: end while

```

B , all the successor modes for each mode of q_D ($Succs(q_i)$) are explored.

The set of explored modes is partitioned into subsets of non discernible-modes, forming the partition denoted by $\mathcal{Q}_{disc} = \mathcal{Q}_{\nu_1} \cup \dots \cup \mathcal{Q}_{\nu_N}$. This information is stored in a knowledge base used by Algorithm 1 such that a new set of residuals is generated only when a mode has not been previously visited.

Transitions of HA are integrated in B , evaluating the discernability property if necessary and analyzing the fault signature matrix to include non-structural faults as faulty modes. If a transition of HA associated with an observable event is found in the exploration, it is kept in B (see line 23). Otherwise, the discernability property is evaluated between these pair of modes. If the two modes are discernible then

an intermediate⁴ mode is added between these modes. The outgoing transition is associated a signature-event, indicating that this mode change can be observed by means of the consistency indicators (see lines from 25-31).

To include information about non-structural faults in B , lines 38 to 49 of Algorithm 1 show how a fault signature matrix is generated associated with the current (set of) modes q_D . It is then analyzed to include the non-structural faulty modes $\mathcal{Q}_{\mathcal{F}_{ns}}$ and the transient modes $\mathcal{Q}_{\mathcal{F}_{ns}}^t$ based on discernability. Analyzing this matrix, the set of non-structural faults can be partitioned into detectable ($\mathcal{F}_{\nu_j}^*$) and non-detectable (\mathcal{F}'_{ν_j}) fault subsets. The set of detectable faults is further partitioned into: $\mathcal{F}_{\nu_j}^* = \mathcal{F}_{\nu_j}^1 \cup \dots \cup \mathcal{F}_{\nu_j}^N$ (lines 13 to 15), where N is the number of discernible, hence isolable, fault subsets.

The signature-events can represent both a mode change or a non-structural fault occurrence, according to this the signature event can be labeled according to:

$$f_{Sig_ev} : \mathcal{Q} \times \overline{\mathcal{Q}} \rightarrow \Sigma^{Sig} \quad (7)$$

$$f_{Sig_ev}(q_i, q_j) \mapsto \begin{cases} \delta_{\nu_i-\nu_j} \in \Sigma^{Sig} & \text{if } (q_i, q_j) \text{ are discernible} \\ \delta_{\mathcal{F}_{\nu_i}}^* \in \Sigma_{\mathcal{F}_{ns}}^{Sig} & \text{if } FS_{\nu_i}(\bullet, f_l) \neq \mathbf{0} \end{cases}$$

where $q_i \in \mathcal{Q}_{\nu_i}$ and $q_j \in \mathcal{Q}_{\nu_j}$ with $\mathcal{Q}_{\nu_i}, \mathcal{Q}_{\nu_j} \subseteq \mathcal{Q}_{disc}$, hence $\delta_{\nu_i-\nu_j}$ denotes an event associated with a mode change between modes in HA and $\delta_{\mathcal{F}_{\nu_i}}^*$ denotes an event associated with a fault f_l belonging to the set $\mathcal{F}_{\nu_i}^* \subseteq \mathcal{F}_{\nu_i}^*$, and $q_j \in \mathcal{Q}_{\mathcal{F}_{ns}}$ associated with the non-structural fault.

4.2 Hybrid Diagnoser

The diagnoser is a finite state machine

$$D = \langle \mathcal{Q}_D, \Sigma_D, \mathcal{T}_D, q_{D_0} \rangle, \text{ where:}$$

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

The transition function \mathcal{T}_D can be calculated according to the propagation algorithms explained in [Sampath *et al.*, 1995], from the incremental B obtained during system mode tracking. The algorithm to build the transition function is executed after the occurrence of an observable event whenever the state has not been previously visited. The part of the diagnoser obtained takes into account only the possible successor modes and hence the transitions that can occur next.

⁴The transient mode is a way to account for the hybrid automaton HA dwell time requirement. This requirement guarantees that residuals, and hence consistency indicators, can be properly computed and that signature-events can be properly issued [Bayouth *et al.*, 2008]

5 Application Case Study

To illustrate the method, a part of the Barcelona sewer network presented in [Vento *et al.*, 2011] is used (see Fig. 2). The elements that appear in the example are: two virtual tanks T_0 and T_1 , a control gate ($gate_1$), two pluviometers P_{19} and P_{16} to measure the rain intensity and two limnimeters L_{39} and L_{41} to measure the sewer level. The control gate is commanded by a controller applying open/close gate actions depending on the flow in the sewer.

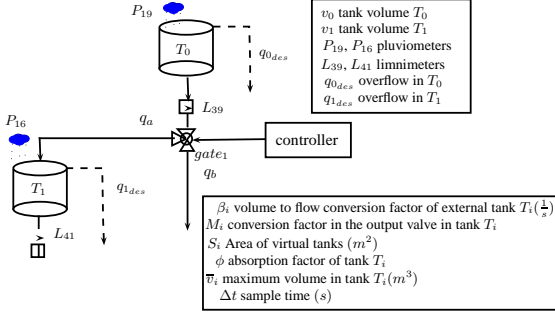


Figure 2: A small part of the sewer network

5.1 Hybrid Modeling

A hybrid automaton model can be obtained to represent the hybrid phenomena present in the network associated with the virtual tanks and the control gate. A way to obtain the hybrid model is to provide the automata for each component (T_0 , T_1 and $gate_1$) and then synchronizing all automata to get the global model [Henzinger, 1996]. The automaton for a virtual tank is given by two discrete states: overflow (O) and non overflow (WO) as is shown in Fig. 3. Regarding the control gate, there are four discrete states, the nominal behaviours (open or closed) and the faulty behaviours (stuck open or stuck closed).

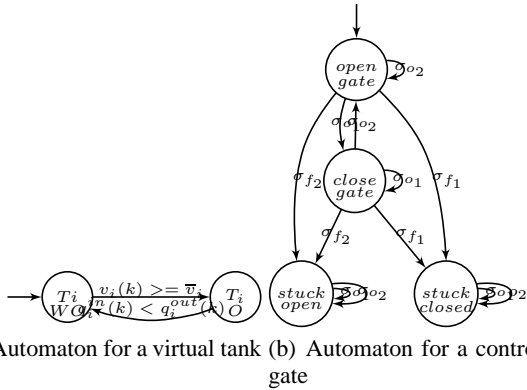


Figure 3: Automaton for the components

The global hybrid automaton has 16 operation modes where 8 of them correspond to nominal modes and the rest correspond to different configurations involving a control gate fault (stuck closed or stuck open). The set of structural faults is given by $\mathcal{F}_s = \{f_1, f_2\}$ and the non-structural faults correspond to additive faults in sensor ($L_{39}, L_{41}, P_{19}, P_{16}$) given by $\mathcal{F}_{ns} = \{f_3, f_4, f_5, f_6\}$, respectively.

The set $\Sigma_s = \{\sigma_{uo1}, \sigma_{uo2}, \sigma_{uo3}, \sigma_{uo4}\}$ represents the unobservable spontaneous events. Event σ_{uo1} corresponds to the volume in tank T_0 reaching its maximum $v_0 \geq \bar{v}_0$. Event σ_{uo2} corresponds to the input flow being less than the output flow from T_0 (i.e., $q_0^{in} < q_0^{out}$). The other events are related to the virtual tank T_1 . The set $\Sigma_{\mathcal{F}_s} = \{\sigma_{f1}, \sigma_{f2}\}$ represents the fault events related to the structural faulty modes and $\Sigma_{\mathcal{F}_{ns}} = \{\sigma_{f3}, \sigma_{f4}, \sigma_{f5}, \sigma_{f6}\}$ the non-structural fault events. The set $\Sigma_c = \{\sigma_{o1}, \sigma_{o2}\}$ gathers input events corresponding to closing or opening the valve issued by the controller.

5.2 Simulation Results

Assume that the system tracks the mode sequence $\{q_1, q_3, q_1, q_5\}$ and the sampling time is $\Delta t = 300s$. Mode q_1 refers to the situation in which no tank is in overflow. Then, T_1 is in overflow during a period of time (mode q_3) until it leaves the overflow situation (mode q_1). Later, the control gate is closed. The diagnoser must track the right mode sequence and detect and isolate the possible faults from an incrementally built behavior automaton B .

Assuming that the initial mode is known and it is q_1 , then applying Algorithms 1 the initial B is shown in Fig. 4. The initial diagnoser is obtained applying the propagation algorithm described in [Sampath *et al.*, 1995] to the initial B . Then, the diagnoser waits for the occurrence of an event. Notice that the initial B includes the possible events that may occur. These events are $\delta_{13}, \delta_{14}, \delta_{12}, \delta_{F_{v1}}^*, \delta_{F_{v1}}^{*2}, \delta_{F_{v1}}^{*3}, \sigma_{o1}$ and σ_{o2} .

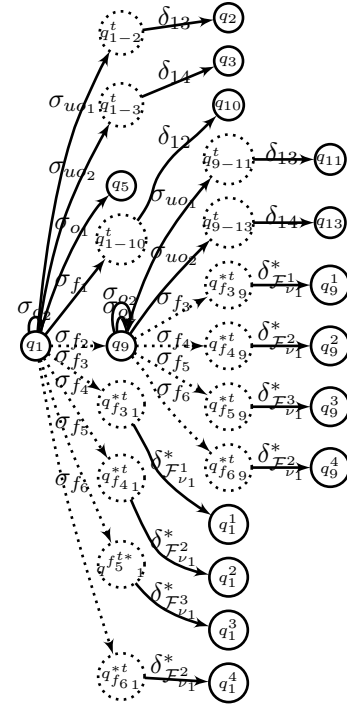


Figure 4: Initial incremental B

Notice that $Q_{disc} = Q_{v1} \cup Q_{v2} \cup Q_{v3} \cup Q_{v4} \cup Q_{v5}$. Fig. 5 shows the value of the set of residuals for all groups in Q_{disc} . According to the set of residuals for the set of modes of HA , two signature-events, δ_{14} and δ_{41} , were identified using the consistency indicators appropriately. These signature-events correspond to transitions $q_1 \rightarrow q_3$ and

$q_3 \rightarrow q_1$ with $q_1 \in \mathcal{Q}_{\nu_1}$ and $q_3 \in \mathcal{Q}_{\nu_4}$. Notice for instance that when the system is in mode q_3 , $\Phi_{\nu_1}(k) \neq \mathbf{0}$ and $\Phi_{\nu_4}(k) = \mathbf{0}$. Both modes $q_1, q_3 \in \mathcal{Q}$ represent a nominal behaviour.

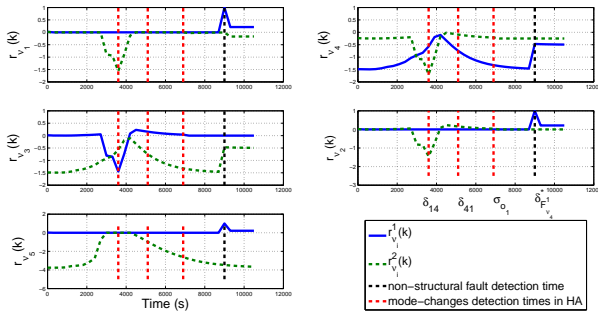


Figure 5: Residuals generation for the set of modes in HA

Later, the observable event σ_{o1} occurs, corresponding to the control gate closing. This event is identified instantaneously and indicates that a mode change from q_1 to q_5 takes place. Fig. 5 shows the set of residuals of mode q_1 and $q_5 \in \mathcal{Q}_{\nu_2}$. It should be noticed that the residuals in mode q_5 are consistent with measurements after δ_{o1} is detected, i.e. $\Phi_{\nu_2}(k) = \mathbf{0}$.

The set of residuals are only generated for modes that are visited in HA . In this way, the efficient use of memory is guaranteed. There is a set of two residuals per group using the expression given by (4).

A non-structural fault then occurs at $9000s$ (indicated in Fig. 5 with a black vertical dashed line). Then, the diagnoser detects the fault at $9300s$. The set of consistency indicators of mode q_5 are used to isolate the fault. The observed signature is $[1 \ 0]^t$ which, according to \mathbf{FS}_{ν_4} , corresponds to a fault in sensor L_{39} . Finally, the hybrid diagnoser stops and reports the diagnosis. Indeed, a non-structural faults needs to be repaired before the diagnoser can resume.

The report given by the hybrid diagnoser is shown in Table 1. The first column represents mode changes in HA , 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 the occurrence time and the detection time of the identified events.

Mode change	Reported event	State diagnoser (total number of states)	Generated residuals	Occurrence time (s)	Detection time (s)
$q_1 \rightarrow q_3$	δ_{14}	$(q_3, \{\}), (q_{13}, \{f_2\})$ (7)	8	3000	3300
$q_3 \rightarrow q_1$	δ_{41}	$(q_1, \{\}), (q_9, \{f_2\})$ (13)	10	4200	4500
$q_1 \rightarrow q_5$	σ_{o1}	$(q_5, \{\}), (q_9, \{f_2\})$ (23)	10	6900	6900
$q_5 \rightarrow q_5^*$	$\delta_{F_{34}}^*$	$(q_5^*, \{f_3\}), (q_5^*, \{f_5\})$ (23)	10	9000	9300
fault $f_3 \in \mathcal{F}_{\nu_4}$ in Mode q_5					
Total		23	10		

Table 1: Hybrid diagnoser report

Table 2 provides a comparison of the results obtained with the present method and those obtained with the off-line diagnoser generation [Vento *et al.*, 2011; Bayouduh *et al.*, 2008], standing out the benefits of the proposed method.

6 Conclusions

A method to incrementally build a hybrid diagnoser has been presented. The diagnoser is built whenever the system

	Previous methods	Proposed method
Number of diagnoser states	75	23
Number of residuals generated	10	10
Computational complexity	Exponential ($2^{N_{statesD}}$)	Lineal ($N_{Success(q_D)}$)
	$N_{statesD}$ Total number of diagnoser state	$N_{Success(q_D)}$ Total number of successors

Table 2: Comparison with previous methods according to the simulation scenario

requires it after an event occurs (signature-event or input event). The method comprises the detection and isolation of structural and non-structural faults which are included 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 operation. Thus, the diagnoser obtained requires less memory space and can be efficiently obtained online. An illustrative example of the proposed approach based on a piece of the Barcelona sewer network is used. Future work will consider to add the incremental design of the HA from the component hybrid automata. This will nicely complete the proposed incremental approach, avoiding not only to store the whole diagnoser but also the whole hybrid model.

References

- [Bayouduh *et al.*, 2008] M. Bayouduh, L. Travé-Massuyès, and X. Olive. Hybrid systems diagnosis by coupling continuous and discrete event techniques. In *Proceedings of the 17th International Federation of Automatic Control, World Congress, IFAC-WC*, pages 7265–7270, Seoul (Korea), 2008.
- [Daigle, 2008] M. Daigle. *A Qualitative Event-Based Approach to Fault Diagnosis of Hybrid Systems*. PhD thesis, Faculty of the Graduate School of Vanderbilt University, Nashville, Tennessee, 2008.
- [Ding *et al.*, 2008] X. Ding, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Model Based Fault Diagnosis Techniques*. Springer, 2008.
- [Henzinger, 1996] T. Henzinger. The theory of hybrid automata. *Logic in Computer Science, 1996. LICS '96. Proceedings., Eleventh Annual IEEE Symposium on*, pages 278–292, July 1996.
- [Lygeros *et al.*, 2003] J. Lygeros, K. Henrik, and J. Zhang. Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, 48, January 2003.
- [Mezyani, 2007] T. Mezyani. *Diagnostic des Systèmes Dynamiques Hybrides*. PhD thesis, Université Lille1, France, 2007.
- [Narasimhan and Biswas, 2007] S. Narasimhan and G. Biswas. Model-based diagnosis of hybrid systems. *IEEE Transactions on Systems, Man and Cybernetics*, 37(3), May 2007.
- [Sampath *et al.*, 1995] M. Sampath, R. Sengupta, and S. Lafortune. Diagnosability of discrete-event system. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.
- [Vento *et al.*, 2011] J. Vento, V. Puig, and R. Sarrate. A methodology for building a fault diagnoser for hybrid systems. In *9th European Workshop on Advance Control and Diagnosis*, Budapest, Hungry, November 2011.