A Methodology for Building a Fault Diagnoser for Hybrid Systems ⋆

Jorge Vento, Vicenç Puig and Ramon Sarrate

Advanced Control Systems (SAC)
Universitat Politècnica de Catalunya (UPC)
Rambla Sant Nebridi, 11, 08222 Terrassa, Spain
(e-mail: {jorge.isaac.vento,vicenc.puig,ramon.sarrate}@upc.edu).

Abstract: In this paper, a design methodology for building diagnosers for hybrid systems is proposed. The design methodology uses as a starting point a hybrid automaton model to represent the hybrid system behaviour by means of the interaction of continuous dynamics and discrete events. Then, a hybrid fault diagnoser is designed using the methodology described in this paper and implemented by means of a discrete event system which carries out the mode recognition and diagnostic tasks, both based on residuals generated using models. Both tasks interact each other since the diagnosis module adapts according to the current mode of the hybrid system. The mode recognition task involves detecting and identifying the mode change by determining the set of residuals that are consistent with the current mode of the hybrid system. On the other hand, the diagnostic task involves detecting and isolating faults by identifying the fault that can explain the set of residuals that are inconsistent. A section of the Barcelona sewer network is used as application case study to illustrate the proposed fault diagnosis for hybrid systems.

Keywords: Fault detection and isolation, hybrid systems, diagnosers.

1. INTRODUCTION

Most real systems are on-line controlled and supervised by means of automatic computer-based control systems. But, they are subject to faults that can appear in the plant components, sensors and actuators. Many of these systems present a behavior that changes with the operating mode that can be modeled as a hybrid system. Thus, fault diagnosis using models, mostly developed for non-hybrid systems, should be extended to handle the hybrid system behavior.

Recently, in the literature, model based techniques have been proposed to diagnose hybrid systems (Travé-Massuyès et al., 2008; Bayoudh, 2009; Cocquempot et al., 2004; Daigle, 2008). The continuous behavior in each mode is described using differential equations. These techniques extend, in some way, existing model-based approaches for non-hybrid systems being able to handle the continuous and discrete-event system behaviors. In hybrid systems, the diagnoser should be parameterized as a function of the current mode. Thus, the proposed diagnoser should be able to evaluate on-line the behaviour of the hybrid system and to detect and isolate the mode and the faults. In (Travé-Massuyès et al., 2008; Bayoudh, 2009), the discrete-event behavior is modeled as a set of discrete modes, that can include nominal or faulty modes, and transitions between them are governed by events. Following the methodology proposed by (Sampath et al., 1995), a diagnoser combining the discrete and the continuous dynamics is built by means of a behaviour automaton. In (Cocquempot et al., 2004), an algorithm to diagnose multi-mode systems is presented. The hybrid system mode is recognized by checking the consistency of the whole set of ARRs generated, considering all system modes, but a discrete-event diagnoser is not built. In a previous work (Vento et al., 2010), the authors propose a fault detection and isolation approach for hybrid systems based on identifying inconsistency between the measured and the estimated system behavior by means of analytical redundancy relations (ARR), extending the approach suggested by (Cocquempot et al., 2004). The hybrid system is described by means of a hybrid automaton model whereas the diagnoser is implemented as a discrete-event system by means of a finite state machine. The events handled by the diagnoser are generated by the residuals that are activated when a fault or a mode change occurs. An algorithm able to distinguish between both types of events is proposed. The set of residuals of the current mode are used to detect and isolate faults while the set of the residuals of the successor modes are used to identify a mode change.

The approach in (Vento et al., 2010) is different from (Travé-Massuyès et al., 2008; Bayoudh, 2009) since it considers faults that do not cause a change of mode, and the mode recognition uses the information provided by the residuals in a different approach.

In this paper, a general methodology for designing and building a diagnoser for hybrid systems is proposed following the diagnoser approach proposed in (Vento et al., 2010). Thus, this paper extends the results presented in

⋆ This work was supported in part by the grant CICYT HYFA DPI2008-01996 and WATMAN DPI2009-13744 of Spanish Ministry of Education.

(Vento et al., 2010; Travé-Massuyès et al., 2008; Bayoudh, 2009; Cocquempot et al., 2004) by proposing a methodology to build in an automatic way the hybrid diagnoser by identifying the set of states and events from the hybrid automaton and the set of residuals. Finally, the proposed algorithm is applied to a section of the Barcelona sewer network, which allows to assess its validity and performance. The hybrid diagnoser generated by the proposed methodology is implemented using SIMULINK/STATEFLOW.

The structure of this paper is the following. In Section 2, an hybrid model description is presented. In Section 3, an overview of the technique proposed to diagnose faults in hybrid systems is presented. In Section 4, the methodology to build a hybrid diagnoser 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 presented in Section 6.

2. HYBRID SYSTEM MODELING

Let us consider that the model of the hybrid system to be diagnosed can be described by the following hybrid automaton $HA =< \mathcal{Q}, X, U, Y, F, G, H, \Sigma, \mathcal{T} >$, where:

- $\mathcal{Q} = \{q^i : i \in M\}$ is a set of discrete states and q^0 is the initial discrete state. The finite set $M = \{1, 2, \cdots, m\}$ represents the nominal and anticipated fault modes of the system.
- $\mathcal{X} \subseteq \Re^{nx}$ defines a discrete-time continuous state space. $\mathbf{x}(k) \in \mathcal{X}$ is the discrete-time state vector at sample k and $\mathbf{x}_0$ the initial state vector.
- $\mathcal{U} \in \Re^{nu}$ defines a discrete-time continuous input space. $\mathbf{u}(k) \in \mathcal{U}$ is the discrete-time continuous input vector.
- $\mathcal{Y} \in \Re^{ny}$ defines a discrete-time continuous output space. $\mathbf{y}(k) \in \mathcal{Y}$ is the discrete-time continuous output vector.
- $\mathcal{F}$ is a set of faults.
- $\mathcal{G} = \{g^i : i \in M\}$ defines a set of discrete-time state affine functions for each mode $i \in M$:

$$\mathbf{x}(k+1) = \mathbf{A}^i\mathbf{x}(k) + \mathbf{B}^i\mathbf{u}(k) + \mathbf{F}_x^i\mathbf{f}(k) + \mathbf{G}_x^i \quad (1)$$

  where $\mathbf{A}^i \in \Re^{nx \times nx}$, $\mathbf{B}^i \in \Re^{nx \times nu}$ and $\mathbf{G}_x^i \in \Re^{nx \times 1}$ are the state matrices in mode i, and $\mathbf{f}(k) \in \Re^{nf}$ represents the faults in the system, with $\mathbf{F}_x^i \in \Re^{nx \times nf}$ being the fault distribution matrix in mode i.
- $\mathcal{H} = \{h^i : i \in M\}$ defines a set of discrete-time output affine functions for each mode $i \in M$:

$$\mathbf{y}(k) = \mathbf{C}^i\mathbf{x}(k) + \mathbf{D}^i\mathbf{u}(k) + \mathbf{F}_y^i\mathbf{f}(k) + \mathbf{G}_y^i \quad (2)$$

  where $\mathbf{C}^i \in \Re^{ny \times nx}$, $\mathbf{D}^i \in \Re^{ny \times nu}$ and $\mathbf{G}_y^i \in \Re^{ny \times 1}$ are the output matrices in mode i, $\mathbf{F}_y^i \in \Re^{ny \times nf}$ being the fault distribution matrix in mode i.
- $\Sigma = \Sigma_s \cup \Sigma_c \cup \Sigma_f$ is a set of events. Spontaneous mode switching events (Σ_s), input events (Σ_c) and fault events Σ_f are considered. Each spontaneous event $\sigma_s \subseteq \Sigma_s$ defines when the state vector intersects a jump surface $S_{\sigma_s} = \{\mathbf{x}(k) \in \mathcal{X} : s_{\sigma_s}(\mathbf{x}(k)) = \mathbf{0}\}$, with s_{σ_s} being a linear switching condition.
- 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. $\Sigma_f \subseteq \Sigma_{uo}$, $\Sigma_c \subseteq \Sigma_o$ and Σ_s can be contained in both partitions.
- $\mathcal{T} : \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$ defines a discrete state transition function.

This hybrid automaton model results from an adaptation of (Lygeros et al., 2003), by introducing faults and events. Other alternative descriptions can be found in the literature (Cocquempot et al., 2004; Travé-Massuyès et al., 2008; Bayoudh, 2009)

Alternatively, the model given by (1) and (2) can be expressed in input-output form using the q-operator (or delay operator) and considering zero initial conditions as follows

$$\mathbf{y}(k) = \mathbf{M}^i(q^{-1})\mathbf{u}(k) + \mathbf{G}_f^i(q^{-1})\mathbf{f}(k) + \mathbf{Q}^i(q^{-1}) \quad (3)$$

where:

$$\begin{aligned}
\mathbf{M}^i(q^{-1}) &= \mathbf{C}^i(q\mathbf{I} - \mathbf{A}^i)^{-1}\mathbf{B}^i + \mathbf{D}^i \\
\mathbf{G}_f^i(q^{-1}) &= \mathbf{C}^i(q\mathbf{I} - \mathbf{A}^i)^{-1}\mathbf{F}_x^i + \mathbf{F}_y^i \\
\mathbf{Q}_y^i(q^{-1}) &= \mathbf{G}_y^i\frac{q}{q-1} \\
\mathbf{Q}^i(q^{-1}) &= \mathbf{Q}_x^i(q^{-1}) + \mathbf{Q}_y^i(q^{-1}) \\
\mathbf{Q}_x^i(q^{-1}) &= \mathbf{C}^i(q\mathbf{I} - \mathbf{A}^i)^{-1}\mathbf{G}_x^i\frac{q}{q-1}
\end{aligned}$$

3. OVERVIEW OF THE HYBRID SYSTEM DIAGNOSIS APPROACH

The hybrid system diagnosis approach used in this paper is an extension of the classic FDI approach where the estimated behaviour of the system obtained from a non-faulty model is compared with the real behaviour available through sensor measurements (see (Vento et al., 2010; Mezyani, 2007)). In particular, the FDI algorithm for hybrid systems takes into account which is the current operation mode i to adapt the model to generate the residuals. The residual expression is given by:

$$\mathbf{r}^i(k) = \mathbf{y}(k) - \mathbf{G}^i(q^{-1})\mathbf{u}(k) - \mathbf{H}^i(q^{-1})\mathbf{y}(k) - \mathbf{Q}_e^i \quad (4)$$

where $\mathbf{G}^i(q^{-1})$, $\mathbf{H}^i(q^{-1})$ and $\mathbf{Q}_e^i$ correspond with the estimated output for the mode i. These matrices can be calculated for instant, using observers, or state predictors (Meseguer et al., 2008).

The fault diagnoser in hybrid systems can be organized in two modules: a fault diagnosis module and a mode recognition module.

3.1 Fault diagnosis module

The diagnosis task is implemented through the fault detection and isolation submodules (see Fig. 1).

Fault detection. This submodule performs the fault detection task by monitoring whether the set of residuals (4) belonging to the current mode become inconsistent. The fault effect form of these residuals is obtained by replacing (3) in (4) (Meseguer et al., 2008) leading to:

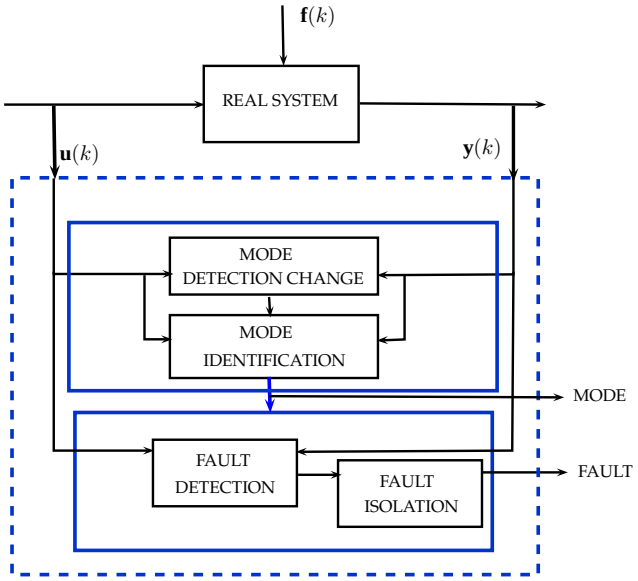

Fig. 1. Conceptual block diagram of hybrid system FDI module

$$\mathbf{r}^i(k) = \mathbf{S}_f^i(q^{-1})\mathbf{f}(k) \tag{5}$$

where:

$$\mathbf{S}_f^i(q^{-1}) = (\mathbf{I} - \mathbf{H}^i(q^{-1}))\mathbf{G}_f^i(q^{-1})$$

The residual expression (5) allows the study of how a fault (without disturbances or noise) affects a given residual through the fault sensitivity matrix defined as follows

$$\mathbf{S}_f^i(q^{-1}) = \frac{\partial \mathbf{r}^i(k)}{\partial \mathbf{f}} \tag{6}$$

Thereby, using this residual expression it is possible to determine the fault signature matrix $\mathbf{FSM}^i$ corresponding to mode i. The theoretical binary fault signature can be determined in automatic way by means of the residual fault sensitivity (6). In particular, given the fault sensitivity of the j^{th} residual with respect to the l^{th} fault denoted as $s_f^i(j,l)$ (i.e, the element (j,l) of the sensitivity matrix $\mathbf{S}_f^i$), the element (j,l) of the fault signature matrix is determined as follows:

$$fsm^i(j,l) = \begin{cases} 1 \text{ if } s_f^i(j,l)(q^{-1}) \neq 0 \\ 0 \text{ if } s_f^i(j,l)(q^{-1}) = 0 \end{cases} \tag{7}$$

i.e., if the j^{th} residual in mode i depends on the l^{th} fault it is coded as a 1 or otherwise as a 0. Since in a hybrid system, residuals change with the mode, the fault sensitivity depends on the mode as well as the theoretical fault signature matrix. Consequently, fault detectability [1] and isolability [2] properties also depend on the system mode.

[1] The l^{th} fault is detectable if there exist some residual that is sensitive to it, i.e., the l^{th} column in the fault signature matrix should contain at least an element equal to 1

[2] The fault l^{th} is isolable if the l^{th} column in the fault signature matrix is different from the rest of the columns

Fault isolation. The isolation submodule is responsible of identifying which is the fault that is present in the system by checking the observed fault signature against the theoretical fault signature matrix. For example, let the theoretical fault signature matrix corresponding to mode i be given by Table 1. Then, the logic of the fault isolation module corresponding to mode i will be based on the following binary test:

$$f_1 = \overline{s}_1^i \wedge \overline{s}_2^i \wedge s_3^i$$
$$f_2 = s_1^i \wedge \overline{s}_2^i \wedge \overline{s}_3^i$$

It compares the observed inconsistent residual with the inconsistency values that should have been observed in case some of the faults considered in the fault signature matrix had been occurred.

Binary codification		Faults	
	Signature	f_1	f_2
$\mathbf{r}^i(k)$	r_1^i	0	1
	r_2^i	0	0
	r_3^i	1	0

Table 1. Fault signature matrix corresponding to mode i

3.2 Mode recognition module

The mode recognition task is implemented through the mode change detection and recognition submodules (see Fig. 1).

Mode change detection. The aim of this submodule is to detect when a mode transition occurs in the hybrid system. A model change can be inferred when an inconsistency in the set of residuals of the current mode i is detected while at the same time the set of residuals corresponding to another mode j is proved to be consistent. When the system is in mode j, $\mathbf{y}(k)$ can be obtained through equation (3). In a fault-less situation, results in $\mathbf{y}(k) = \mathbf{M}^j(q^{-1})\mathbf{u}(k)$. Replacing this expression in (4) leads to:

$$\begin{aligned}\mathbf{r}^{i/j}(k) = &\left(\mathbf{I} - \mathbf{H}^i(q^{-1})\right)\left(\mathbf{M}^j(q^{-1})\mathbf{u}(k) + \mathbf{Q}^j(q^{-1})\right) \\ &-\mathbf{G}^i(q^{-1})\mathbf{u}(k) - \mathbf{Q}_e^i(q^{-1})\end{aligned} \tag{8}$$

As long as the mode assumed by the diagnoser coincides with the current mode of the system (i.e., $i = j$) $\mathbf{r}^i(k) = 0$. On the other hand, when $i \neq j$ means that the mode assumed by the diagnoser is different from the current mode of the system. Then, $\mathbf{r}^i(k) \neq 0$ as long as both modes are discernable. Two modes i and j are non-discernable if $(I - \mathbf{H}^i(q^{-1}))\mathbf{M}^j(q^{-1}) = \mathbf{G}^i(q^{-1})$ and $\left(\mathbf{I} - \mathbf{H}^i(q^{-1})\right)\mathbf{Q}^j(q^{-1}) = \mathbf{Q}_e^i(q^{-1})$. The notion of non-discernability was first introduced in (Cocquempot et al., 2004), where necessary and sufficient conditions were provided for the parity space method.

Considering the whole set of residuals that can be generated for all system operating points, a mode signature matrix $\mathbf{MSM}^i$ for each mode i can be automatically generated. This table contains as many rows the residuals corresponding to the current mode and its successor modes,

and as many columns as the current mode and its successor modes). Let $\mathbf{K}_1^{i,j}(q^{-1}) = \left(\mathbf{I} - \mathbf{H}^i(q^{-1})\right)\mathbf{M}^j(q^{-1}) - \mathbf{G}^i(q^{-1})$ and $\mathbf{K}_2^{i,j}(q^{-1}) = \left(\mathbf{I} - \mathbf{H}^i(q^{-1})\right)\mathbf{Q}^j(q^{-1}) - \mathbf{Q}_e^i(q^{-1})$. Then, the residual expression can be rewritten as:

$$\mathbf{r}^{i/j}(k) = \mathbf{K}_1^{i,j}(q^{-1})\mathbf{u}(k) + \mathbf{K}_2^{i,j}(q^{-1}) \tag{9}$$

Therefore, each element of the $\mathbf{MSM}^i$ is given by:

$$msm^i(r_l^j) = \begin{cases} 0 \text{ if } K_1^{i,j}(q^{-1}) = 0 \wedge K_2^{i,j}(q^{-1}) = 0 \\ 1 \quad \text{otherwise} \end{cases} \tag{10}$$

where i is the current mode and j is the successor mode and l is the residual component.

Mode identification. Once a mode transition has been detected, the new mode should be identified using the mode signature matrix. To identify the reached mode, the current observed signature is compared against the state signatures of the successor states. When a set of residuals of a given mode j is proved to be consistent with the observed signature, it is assumed that a change from the current mode to a new state j has been occurred. To illustrate this procedure, let us consider, as an example, the state signature matrix presented in Table 2. Once a mode change is detected (because e.g. $\mathbf{r}^1 \neq 0$), the following set of consistency tests are evaluated using the residuals of the successor modes.

$$q_2 = \overline{s}_1^2 \wedge \overline{s}_2^2 \wedge \overline{s}_3^2$$
$$q_3 = \overline{s}_1^3 \wedge \overline{s}_2^3 \wedge \overline{s}_3^3$$

The new current mode i will be the mode which satisfies $\mathbf{r}^i(k) = 0$. In this example, the new current mode could be q_2 or q_3, depending on wether the set of residuals that are consistent is $\mathbf{r}^2$ or $\mathbf{r}^3$. The values denoted by $*$ in the table can be 0 or 1 depending on wether the condition given by (10) is satisfied. But, these values are not taken into account to track the state sequence.

		Current mode	Successor modes	
		q_1	q_2	q_3
r^1	r_1^1	0	$*$	$*$
	r_2^1	0	$*$	$*$
	r_3^1	0	$*$	$*$
r^2	r_1^2	$*$	0	$*$
	r_2^2	$*$	0	$*$
	r_3^2	$*$	0	$*$
r^3	r_1^3	$*$	$*$	0
	r_2^3	$*$	$*$	0
	r_3^3	$*$	$*$	0

Table 2. Mode signature matrix corresponding to mode 1

3.3 Hybrid diagnoser

A diagnoser for hybrid systems based on the conceptual scheme of Fig. 1 integrates all previous modules to track the mode change sequence and detect and isolate faults. *Algorithm* 1 briefly describes the logic used by the diagnoser to reason based on the observed signature and the fault and mode signature tables.

Algorithm 1 Hybrid Diagnoser

```
1: i ← 0
2: repeat
3:    Evaluate r^i(k) according to (8)
4: until r^i(k) ≠ 0
5: for all j such that q^j ∈ {q : ∀σ ∈ Σ, q = T(q^i, σ)} do
6:    Evaluate r^j(k) according to (8)
7:    if r^j(k) = 0 then
8:       print Transition from mode i to j
9:       i ← j
10:      goto line 2
11:   end if
12: end for
13: for all Fault in the system, f ∈ F do
14:   if r^i(k) = fsm^i(•, f) then
15:      print Fault f occurred
16:      STOP
17:   end if
18: end for
19: print Unknown event
```

The key idea is to check the consistency of the residuals of the current mode until an inconsistency is detected. Then, once the residuals of the current mode are proved to be inconsistent, two hypothesis should be verified: a mode change or a fault occurrence. First, it is assumed that a mode change has occurred so that the diagnoser waits until any of the residual sets corresponding to a successor mode is proved to be consistent and identify the new mode. If, after some time window, this consistency is not proved, then a fault is assumed and identified by comparing the observed fault signature against the ones stored in the fault signature matrix corresponding to the current mode.

4. FAULT DIAGNOSER BUILDING METHODOLOGY

The goal of the diagnoser is to detect the occurrence of faults or mode changes in a system with hybrid behavior. According to the preceding section, one way to accomplish this is by monitoring residuals. However, the events defined on the hybrid automaton model can also be useful in this task. Thus, the diagnoser is described by a finite state machine that accepts events generated from the hybrid automaton as well as from residuals.

An algorithm to build a diagnoser for hybrid systems is presented in this section. The procedure to construct the diagnoser consists in two steps. The first step involves obtaining a discrete-event abstraction of the hybrid automaton model of the system. In the second step, the theory of (Sampath et al., 1995) is applied to this discrete-event abstraction to obtain the fault diagnoser. The first step involves the abstraction of the hybrid automaton model into a finite state machine with observable and unobservable events using the information provided by the theoretical fault and mode signature matrices. A finite state machine that reflects the normal and failed behavior of the system is built. The model describes admissible sequences of observable or unobservable events. Typically, observable events include commands issued by the operator while unobservable events include failure events.

In (Sampath et al., 1995), a methodology to design fault diagnosers for discrete-event systems was developed. The

diagnoser design methodology produces a state machine that only accepts observable events, and binds a diagnosis statement to every state.

The discrete-event abstraction model will be called Behavior Automaton (this is an extension of the behavior automaton introduced in (Travé-Massuyès et al., 2008), adding faulty modes without dynamic in the abstraction of the hybrid model). The Behavior Automaton is defined as follows:

$$BA =< \mathcal{Q}_{BA}, q^0, \Sigma_{BA}, \mathcal{T}_{BA} >$$

- $\mathcal{Q}_{BA}$ is a set of discrete states, which can be divided into the following types:
  - States related to nominal operation modes in the hybrid automaton model.
  - Faulty states related to the occurrence of a fault in a mode of the hybrid automaton.
  - States related to the detection of a fault in a mode of the hybrid automaton.
  - Intermediate states between two successor nominal modes of the hybrid automaton that are discernable.
- q^0 is the initial state that coincides with the initial nominal mode of the hybrid automaton.
- Σ_{BA} is the set of discrete events, which can be classified into the following categories:
  - Observable and unobservable events defined in the hybrid automaton. Observable events include all input events and, possibly, some spontaneous events. Unobservable events include, possibly, the rest of spontaneous events.
  - Observable events related to detectable faults. These are generated based on the fault signature matrix.
  - Unobservable events signaling the occurrence of a fault.
  - Observable events related to a discernable mode switching. These are generated based on the mode signature matrix.
- $\mathcal{T}_{BA} : \mathcal{Q}_{BA} \times \Sigma_{BA} \rightarrow \mathcal{Q}_{BA}$ is a partial transition function, defined by Algorithm 2.

The fault and mode signature matrices play an important role in the diagnoser and behavior automaton due to the fact that they provide information about fault detectability and isolability properties, as well as discernability between nominal successor modes.

The algorithm is developed based on the following three assumptions:

- It is assumed that a mode change and a fault do not occur at the same time instant.
- The residual dynamics have time to stabilize between two consecutive mode switchings.
- A mode change is not possible after a fault has occurred.

Once a fault has been detected, the diagnoser stops and it is not possible to detect further mode changes. It must be stated that the algorithm has been designed for the single fault case.

5. APPLICATION CASE STUDY

To illustrate the methodology, a small part of the Barcelona sewer network will be used. Fig. 2 shows the model of this part of the Barcelona network using the virtual tank modeling approach (Ocampo and Puigs, 2009). This approach is based on decomposing the network into catchments that are modeled as a tank [3]. Then, the mass balance conservation law is applied to this catchment (tank).

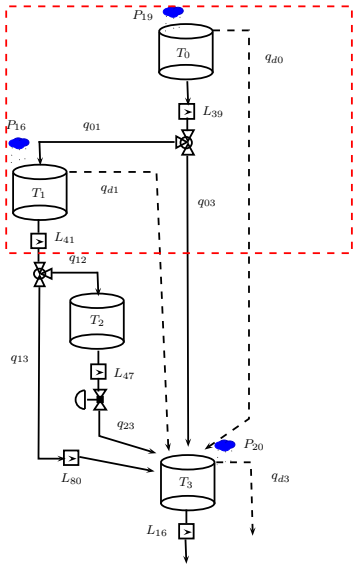

Fig. 2. Sewer network scheme

The network shown in Fig. 2 is composed of 3 virtual tanks (T_0, T_1 and T_3), 1 real tank (T_2) and 3 control gates. 4 output sensors measure the water level (L_{39}, L_{41}, L_{47} and L_{16}) and 3 input sensors measure the rain intensity (P_{19}, P_{16} and P_{20}). The flows denoted by q_{d0}, q_{d1} and q_{d3} represent the overflow in virtual tanks T_0, T_1 and T_3, respectively. A hybrid automaton will be used to model such behaviour.

5.1 Hybrid model

For this particular example, the hybrid automaton describing the sewer network is shown in Fig. 3. There are 8 discrete states. The state variables are the volumes of the virtual tanks. The input is the rain intensity and the output is the level of the tanks. The volume is not measured so events are unobservable

The set of possible faults $\mathcal{F}$ is divided into output sensor faults $\{f_{L_{39}}, f_{L_{41}}, f_{L_{47}}, f_{L_{16}}\}$ and input sensor faults $\{f_{P_{19}}, f_{P_{16}}, f_{P_{20}}\}$.

5.2 Diagnoser building

The diagnoser is built applying the methodology explained in Section 4. A set of 4 residuals are designed for each mode based on structural analysis (Blanke et al., 2006). Then, the theoretical binary matrices, $\mathbf{FSM}^i$ and $\mathbf{MSM}^i$,

[3] At any given time, the stored volume in a virtual tank represents the amount of water inside the mains of the corresponding catchment.

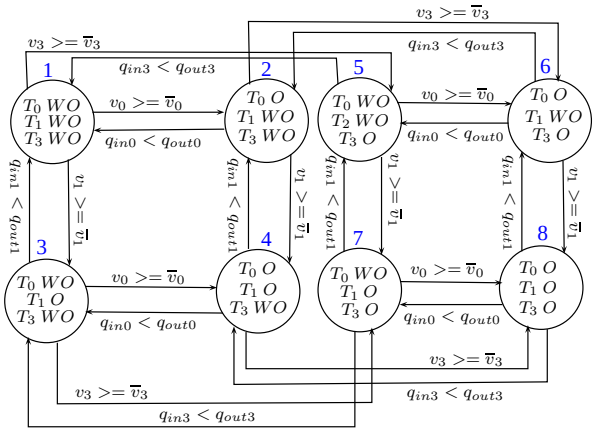

Fig. 3. Hybrid automata for the sewer network

are obtained using the approach explained in Section 3. Table 3 and 4 present $\mathbf{MSM}^1$ and $\mathbf{FSM}^1$ respectively, corresponding to mode q^1. In total, there would be 8 tables, one for each mode.

	Nominal	Successors		
	q^1	q^2	q^3	q^5
r^1_1	0	*	*	*
r^1_2	0	*	*	*
r^1_3	0	*	*	*
r^1_4	0	*	*	*
r^2_1	*	0	*	*
r^2_2	*	0	*	*
r^2_3	*	0	*	*
r^2_4	*	0	*	*
r^3_1	*	*	0	*
r^3_2	*	*	0	*
r^3_3	*	*	0	*
r^3_4	*	*	0	*
r^5_1	*	*	*	0
r^5_2	*	*	*	0
r^5_3	*	*	*	0
r^5_4	*	*	*	0

Table 3. Mode signature matrix $\mathbf{MSM}^1$ for mode q^1

	Nominal	Faults						
	q^1	f_{L39}	f_{L41}	f_{L47}	f_{L16}	f_{P19}	f_{P16}	f_{P20}
r^1_1	0	1	0	0	0	1	0	0
r^1_2	0	1	1	0	0	0	1	0
r^1_3	0	0	1	1	0	0	0	0
r^1_4	0	1	1	1	1	0	0	1

Table 4. Fault signature matrix $\mathbf{FSM}^1$ for mode q^1

$\mathbf{MSM}^i$ is used to abstract the mode switchings in a set of events, whereas $\mathbf{FSM}^i$ is used to abstract detectable fault occurrences in a set of another events. Both set of events are incorporated to the behaviour automaton along with the events of the hybrid automaton applying the steps of the Algorithm 2. $\mathbf{FSM}^1$ shows that all faults are detectable. Furthermore, all faults can be isolated except, f_{L16} and f_{P20}, since they have the same fault signature. The information provided by $\mathbf{MSM}^1$ shows mode q^1 and their successors are discernable since all have a different mode signature.

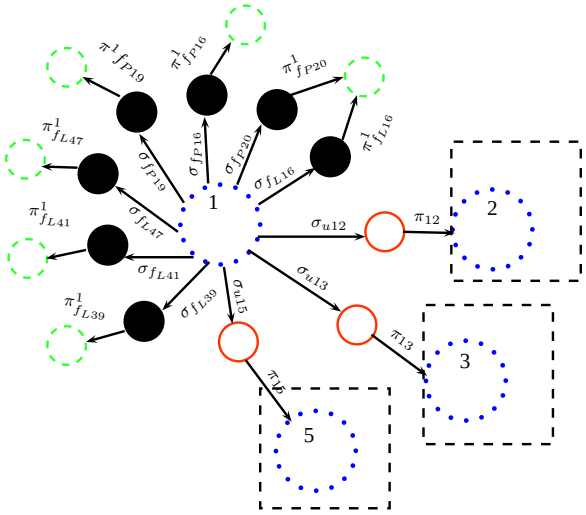

Fig. 4. A part of the behaviour automaton corresponding to nominal mode q^1

A part of the behaviour automaton that corresponds to mode q^1 is shown in Fig. 4. Leaving from the initial state q^1, 7 successor faulty states are defined (states in black color), that correspond to every input and output sensor fault. For every faulty state, there is one unobservable event denoted by $\sigma_{f_{P19}}$, $\sigma_{f_{P16}}$, $\sigma_{f_{P20}}$, $\sigma_{f_{L39}}$, $\sigma_{f_{L41}}$, $\sigma_{f_{L47}}$, and $\sigma_{f_{L16}}$ (see step 14 in Algorithm 2). For every fault isolability set, there is 1 detectable faulty state (states in dashed line). There are 6 detectable faulty states since $f_{L_{16}}$ and $f_{P_{20}}$ belong to the same fault isolability set. These states are related to events $\pi_{f_{L39}}$, $\pi_{f_{L41}}$, $\pi_{f_{L47}}$, $\pi_{f_{P19}}$, $\pi_{f_{P16}}$, $\pi_{f_{P20}}$ and $\pi_{f_{L16}}$. Events $\pi_{f_{L16}}$ and $\pi_{f_{P20}}$ are equivalent (see step 15 in Algorithm 2).

The nominal successor modes of q^1 are q^2, q^3 and q^5. The corresponding events in the hybrid automaton (σ_{u12}, σ_{u13} and σ_{u15}) are unobservable and mode q^1 and their successors are discernable, according to $\mathbf{MSM}^1$. Therefore, an intermediate state (states in solid line) between q^1 and their successor mode are defined. The first transition is related to an unobservable event defined in the hybrid automaton (step 26 in Algorithm 2). Then, according to $\mathbf{MSM}^1$, the second transition to the nominal successor mode is bind to an observable event denoted by π_{12}, π_{13} and π_{15}. From the behaviour automaton, it can be seen that the detectability and isolability properties depend on the current mode q^i

Then, applying (Sampath et al., 1995) the resulting diagnoser is obtained (see Fig. 5).

The resulting diagnoser was implemented in STATEFLOW to validate the methodology. A SIMULINK model was implemented that contains the different modules to track the state sequence of the system and diagnose the possible faults.

5.3 Application to fault scenarios

A simulation scenario including a fault in sensor L_{41} is presented to validate the diagnoser operation. Fig. 6 shows how the diagnoser tracks the state and the detectable faults. The state sequence estimated by the diagnoser is $\{q^1, q^3, q^7, q^5\}$. Initially, neither tank is in overflow. Then,

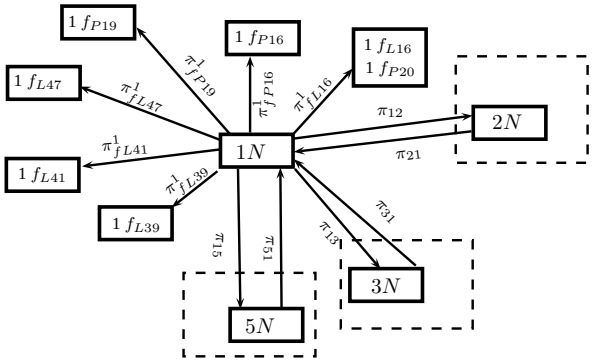

Fig. 5. A part of the diagnoser for the sewer network including the 4 tanks

in the mode q^3, T_1 is in overflow while later T_3 and T_1 are in overflow in mode q^7. Finally, T_1 leaves the overflow situation but T_3 continues in overflow and then the fault appears.

Fig. 6*a* and 6*b* show the set of residuals $\mathbf{r}^5$ and $\mathbf{r}^7$ corresponding to modes q^5 and q^7. While the system remains in mode q^7 the set of residuals $\mathbf{r}^7 = \mathbf{0}$ until a transition from this mode to mode q^5 takes place. Then, the set of residuals $\mathbf{r}^5$ become $\mathbf{0}$.

When the fault occurs the set of residuals $\mathbf{r}^5$ (see Fig. 6*b*) become different from zero. The fault is then isolated checking the observed fault signature against the $\mathbf{FSM}^5$. The fault is detectable and isolable. The fault occurrence and detection are shown in Fig. 6*d*).

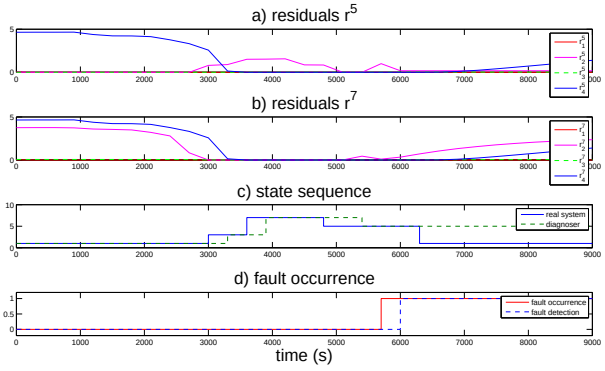

Fig. 6. Tracking mode and fault detection

The diagnoser reports the events (transitions and faults) shown in Table 5. Fig. 6*c* illustrates the state sequence estimated by the diagnoser and the real state sequence. Finally, the fault in L_{41} is detected at $6000s$, but has occurred at $5700s$ (see Fig. 6*d*)).

Diagnoser output	Simulation time (s)	Detection time (s)	Events sequence
$q^1 \rightarrow q^3$	3300	3600	$\sigma_{u13} \rightarrow \pi_{13}$
$q^3 \rightarrow q^7$	3600	3900	$\sigma_{u37} \rightarrow \pi_{37}$
$q^7 \rightarrow q^5$	4800	5400	$\sigma_{u75} \rightarrow \pi_{75}$
Fault in L_{41}	5700	6000	$\sigma_{f_{L41}} \rightarrow \pi^5_{f_{L41}}$

Table 5. Diagnoser report

6. CONCLUSION

This paper has presented a methodology for designing and building a diagnoser for hybrid systems. The design methodology uses a hybrid automaton model and abstract the continuous dynamics based on residuals using models, obtaining the **MSM** and **FSM** matrices. The resulting diagnoser is able to track the system mode and detect and isolate the faults. Both tasks interact each other since the diagnosis module adapts according to the current mode of the hybrid system. The mode recognition task involves detecting and identifying the mode change by determining the set of residuals that are consistent with the current mode of the hybrid system. On the other hand, the diagnostic task involves detecting and isolating the fault by identifying faults that can explain the set of residuals that are inconsistent. Finally, a section of the Barcelona sewer network has been used to assess the validity and performance of the proposed methodology. As future work, a tool that will allow to build the diagnoser in an automatic way will be developed, following the methodology explained in this work. Furthermore, uncertainty in the model will be taken into account to improve the detection robustness in the residuals activation.

REFERENCES

Bayoudh, M. (2009). *Active diagnosis of Hybrid systems Guided by Diagnosability Properties- Application to Autonomous Satellites*. Ph.D. thesis, l'Université de Toulouse, Institut National Polytechnique, France.

Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. (2006). *Diagnosis and Fault Tolerant Control*. Springer, 2nd edition.

Cocquempot, V., Mezyani, T., and Staroswiecki, M. (2004). Fault detection and isolation for hybrid systems using structured parity residuals. In *5th Asian Control Conference*.

Daigle, M. (2008). *A Qualitative Event-Based Approach to Fault Diagnosis of Hybrid Systems*. Ph.D. thesis, Faculty of the Graduate School of Vanderbilt University, Nashville, Tennessee.

Lygeros, J., Henrik, K., and Zhang, J. (2003). Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, 48.

Meseguer, J., Puig, V., and Escobet, T. (2008). Fault diagnosis using a timed discrete event approach based on interval observers. In *Proceedings of the 17th World Congress*, 6914–6919. Seoul, Korea.

Mezyani, T. (2007). *Diagnostic des Systèmes Dynamiques Hybrides*. Ph.D. thesis, Université Lille1, France.

Ocampo, C. and Puigs, V. (2009). Fault-tolerant model predictive control within the hybrid systems framework: Application to sewer networks. 23(8).

Sampath, M., Sengupta, R., and Lafortune, S. (1995). Diagnosability of discrete-event system. *IEEE Transactions on Automatic Control*, 40(9), 1555–1575.

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

Vento, J., Puig, V., and Sarrate, R. (2010). Fault detection and isolation of hybrid system using diagnosers that combine discrete and continuous dynamics. In *Con-*

ference on Control and Fault Tolerant System. Nice, French.

Algorithm 2 Steps to build the behavior automaton

for all nominal mode q^i in the hybrid automaton **do**
  Determine the set of observable and unobservable events that lead to a nominal successor mode in the hybrid automaton model.
  Generate the set of residuals $\mathbf{r}^i(k)$
  Build the fault signature matrix $\mathbf{FSM}^i$ according to the set of faults $\mathcal{F}$
  Build the mode signature matrix $\mathbf{MSM}^i$ according to the successor modes of q^i
  Determine the set of faults that are detectable in q^i, according to the $\mathbf{FSM}^i$. Among them, determine the fault isolability sets.
  Define the set of observable events related to those fault isolability sets, based on the $\mathbf{FSM}^i$.
  For the nominal successor modes that are reachable through an unobservable event in the hybrid automaton, determine those that are discernable, according to the $\mathbf{MSM}^i$.
  Define the set of observable events related to a discernable mode switching, based on the $\mathbf{MSM}^i$.
end for
for all nominal mode q^i in the hybrid automaton **do**
  Define a state in the behavior automaton
  for all fault $f \in \mathcal{F}$ **do**
    Define a successor faulty state in the behavior automaton. This successor state is reachable through the unobservable event that signals the occurrence of f
    if f is detectable **then**
      Define a successor state to the faulty state in the behavior automaton. This successor state is reachable through the observable event that is related to the isolability set to which f belongs
    end if
  end for
  for all nominal successor modes q^j of q^i in the hybrid automaton model **do**
    if q^j is reachable through an observable event **then**
      The nominal successor state in the behavior automaton is reachable through this observable event
    else if q^j is reachable through an unobservable event **then**
      if q^i and q^j are not discernable **then**
        The nominal successor state in the behavior automaton is reachable through this unobservable event
      else
        Define an intermediate state in the behavior automaton. This successor state is reachable through that unobservable event.
        The nominal successor state in the behavior automaton is reachable from this intermediate state through the observable event related to the corresponding mode switching in the hybrid automaton model.
      end if
    end if
  end for
end for