Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Hybrid Systems Diagnosis by coupling Continuous and Discrete event Techniques ⋆

Mehdi Bayoudh * **Louise Travé-Massuyès** * **Xavier Olive** **

* *LAAS-CNRS, Université de Toulouse, FRANCE*
e-mail: {bayoudh, louise}@laas.fr
** *Thales Alenia Space, FRANCE*
e-mail: xavier.olive@thalesaleniaspace.com

Abstract: This paper deals with the problem of diagnosing systems that exhibit both continuous and discrete event dynamics. The proposed approach combines techniques from both continuous and discrete event diagnosis fields. On the on hand, an extension of the parity space approach is used to associate signatures to every operational mode of the system. On the other hand, signature switches arising from the transition from one mode to another are abstracted in the form of a set of events that capture the continuous dynamics. These events are merged into the original discrete dynamic model of the system, allowing us to apply the well-known discrete-event-systems diagnoser approach. This is illustrated on an example that shows the diagnosability improvement of the hybrid approach.

1. INTRODUCTION

In complex dynamical systems, the overall physical plant is inherently continuous, but control is often performed by a supervisory controller that imposes discrete switching behavior between several operating modes (like in McIlraith et al. [2000]). In such cases, diagnosis is often performed separately on the continuous dynamics and on the discrete dynamics. However, this approach misses the interaction between both dynamics, resulting in poor diagnosability. We show that model based diagnosis techniques can be combined if the system is represented by a hybrid model as mentioned in Hofbaur and Williams [2004].
The hybrid behavior is seen as the result of the underlying discrete event and continuous systems behaviors. The underlying continuous system (also called *the multimode system*) can be diagnosed by extending the parity space approach. The resulting residuals are linked with every operating mode, and allow us to check the consistency between observable behavior and the system model in each mode like in Cocquempot et al. [2004]. Every operating mode is indeed characterized by a mode signature. We then propose to abstract the signature switches in the form of discrete events typifying continuous dynamics. These events are then merged with pure discrete events in the discrete event model of the system. The resulting model allows us to build a hybrid diagnoser that takes as input all observable events. The hybrid diagnoser is used to support an online state tracking algorithm that achieves better diagnosability (mode discriminability) defined in Bayoudh et al. [2007] than a diagnosis approach based on the multimode system model or a diagnosis approach based on the original discrete event model. Our paper shows the advantage of hybrid diagnosis and uses an illustrative example that achieves diagnosability although neither the underlying continuous system nor the underlying discrete event system are diagnosable.
The paper is organized as follows: in section 2 we propose the theoretical framework to hybrid system modeling. In section 3 we propose an approach to diagnose the underlying continuous system. Section 4 provide the classical framework for discrete event system diagnosis. Then, in section 5 we present our approach proposed to hybrid system diagnosis by coupling both continuous and discrete event techniques.

⋆ This work was supported in part by Thales Alenia Space, France.

2. HYBRID MODELING FRAMEWORK

As mentioned in Henzinger [1996] and Hofbaur and Williams [2004], a hybrid system may be described by a hybrid automaton defined as a tuple $S = (\zeta, Q, \Sigma, T, C, (q_0, \zeta_0))$.
Where:

- ζ is the set of continuous variables, which include observable and non observable variables. The set of observable variables is denoted by ζ_{OBS}.
- Q is the set of discrete system states. Each state $q_i \in Q$ represents a functional mode of the system. It includes nominal and anticipated fault modes.
- Σ is the set of events. Events correspond to command value switches, spontaneous mode changes and fault events.
- $\Sigma o \subseteq \Sigma$ is the set of observable events. Without loss of generality, we assume that fault events are unobservable.
- T is the transition function, T: Q$\times\ \Sigma \rightarrow Q$.
- C is the set of system constraints linking continuous variables. It represents the set of differential and algebraic equations modeling the continuous behavior of the system.
- $(\zeta_0, q_0) \in \zeta \times Q$, is the initial condition.

2.1 The Underlying Discrete Event System (DES)

The discrete part of the hybrid automaton, given by $M = (Q, \Sigma, T, q_0)$, is a discrete automaton that describes the discrete dynamics of the system, i.e. the possible evolutions between operating modes of Q. Modes include nominal and fault modes as well as an unknown mode, which stands for all the non anticipated faulty situations. The unknown mode has no specified behavior and hence no associated constraints.

2.2 The Underlying Continuous System (CS)

In this paper, we deal with linear systems, modeled in the state space by the evolution and the observation equations. Under this assumption, the continuous part of the hybrid automaton is given by the continuous models associated to each mode q_i in the following form:

$$\begin{cases} X_i(n+1) = A_i X_i(n) + B_i U(n) \\ Y(n) = C_i X_i(n) + D_i U(n) \end{cases}$$

with:

- $X_i(n)$: the state vector at time step nT_s.
- $U(n)$: the input vector at time step nT_s.
- $Y(n)$: the output vector at time step nT_s.

T_s is the sampling period; A_i, B_i, C_i and D_i are constant matrices of appropriate dimensions. The underlying continuous system $\Xi = (\zeta, Q, C, \zeta_0)$ (also called *the multimode system*) describes the whole continuous behavior of the system. Notice that transitions between modes are implicit and consequently, transitions are not constrained in any way.

2.3 Illustrative example – Underlying CS and DES

We consider a dynamic hybrid system whose underlying discrete system is described by the automaton of figure 1. o_1 and o_2 are observable events (commands, discrete sensor outputs, ...), uo_1, uo_2 and uo_3 are unobservable events, that possibly represent fault events. Modes q_1, q_2, q_3 and q_4 can be nominal

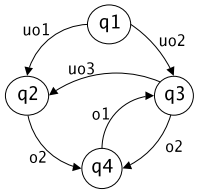

Fig. 1. The Underlying Discrete System

or anticipated fault modes [1].
The underlying continuous system is given by a state space form model for every mode $q_i, i \in [|1,4|]$. Without loss of generality, we consider no noise and no disturbances.

$$\begin{cases} X_i(n+1) = A_i X_i(n) + B_i U(n) \\ Y(n) = C_i X_i(n) + D_i U(n) \end{cases}$$

where

$$A_1 = \begin{pmatrix} 0.7 & 0 \\ 0 & 0.7 \end{pmatrix}, A_2 = \begin{pmatrix} -0.5 & 4 & 0 \\ 0 & 0.6 & 0 \\ 6 & 0 & 0.8 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 0.3 & -0.3 & 0 \\ 0 & 0.6 & 0 \\ -0.3 & 0 & 0.9 \end{pmatrix}, A_4 = \begin{pmatrix} 0.6 & -0.3 & 0 \\ 0.3 & 0.6 & 0 \\ -0.6 & 0 & 0.9 \end{pmatrix}$$

$$B_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix}, B_2 = B_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, B_4 = \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix}$$

$$C_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, C_4 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$$D_1 = D_2 = D_3 = D_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

[1] For sake of simplicity, the unknown mode is not represented in the automaton figure.

3. DIAGNOSIS OF THE UNDERLYING CS

The underlying continuous system is diagnosed by extending the parity space approach to multimode systems.

3.1 Parity Space Approach extended to multimode systems

Following the parity space approach, consistency tests may take the form of a set of Analytical Redundancy Relations (ARRs) by eliminating non observable variables Cordier et al. [2004]. The ARRs set associated to a mode q_i is denoted by ARR_i.
An Analytic Redundancy Relation ARR_{ij} can be expressed as $r_{ij} = 0$, where r_{ij} is called the residual of the ARR. Since ARRs are constraints that only contain observable variables, they can be evaluated on-line with the incoming observations given by the sensors, allowing one to check the consistency of the observed against the predicted system's behavior. ARRs are satisfied if the observations satisfy the model constraints, in which case the associated residuals are zero. In the opposite case, all or some of the residuals are non zero. The set of residuals in mode q_i hence results in a local Boolean fault indicator tuple.

$$r_{ij} = \begin{cases} 0 \text{ when } ARR_{ij} \text{ is satisfied} \\ 1 \text{ otherwise} \end{cases}$$

$j = 1, ..., N_{ARR(q_i)}$, where $N_{ARR}(q_i)$ is the number of associated ARRs/residuals.
The Parity space approach has been recently extended to multimode systems in Cocquempot et al. [2004].
Given a vector V, let us denote by V^p the vector obtained by the concatenation of the vector values at every sampling instant $(n-p+k)$, $0 \leq k \leq p$, for a given order p.
Hence $V^p(n) = [V^T(n-p), ..., V^T(n-p+k), ..., V^T(n)]^T$.
Consider a multimode system like Ξ and a mode q_i, then the computational form of the residual vector,
$R_i = [r_{i1}, r_{i2}, ..., r_{iN_{ARR}(q_i)}]^T$ at order p_i is:

$$\rho_{c_i}^{p_i}(n) = \Omega_i^{p_i} Y^{p_i}(n) - \Omega_i^{p_i} L_i^{p_i}(A_i, B_i, C_i, D_i) U^{p_i}(n)$$

and its evaluation form is :

$$\rho_{e_i}(n) = 0$$

with:

$$L_i^{p_i}(M_i, N_i, P_i, Q_i) = \begin{pmatrix} Q_i & 0 & ... & 0 \\ P_i N_i & Q_i & ... & ... \\ ... & ... & ... & 0 \\ P_i M_i^{(p_i-1)} N_i & ... & P_i N_i & Q_i \end{pmatrix}$$

$$O_i^{p_i} = \begin{pmatrix} C_i \\ C_i A_i \\ ... \\ C_i A_i^{p_i} \end{pmatrix}$$

and $\Omega_i^{p_i}$ is a matrix orthogonal to $O_i^{p_i}$(there always exists an

order p_i such that $O_i^{p_i}$ exists).
In the multimode framework, the set of ARRs linked with each functional system mode generally is different, although some ARRs may be shared.

3.2 Illustrative example – ARRs Computation

Let us take again the example of figure 1. For every mode, the order of the parity space is 1, i.e. the computational form is calculated from the continuous observable variables U and Y, at time $n-1$ and n, and given as follows:

- $\rho^1_{c_1}(n) = \begin{pmatrix} -0.70 & 0.10 & 0.70 & -0.10 \\ -0.10 & -0.70 & 0.10 & 0.70 \end{pmatrix} \begin{pmatrix} y_1(n-1) \\ y_2(n-1) \\ y_1(n) \\ y_2(n) \end{pmatrix}$
  $+ \begin{pmatrix} 0.10 & -0.70 \\ -0.70 & -0.10 \end{pmatrix} \begin{pmatrix} u_1(n-1) \\ u_1(n) \end{pmatrix}$
- $\rho^1_{c_2}(n) = \begin{pmatrix} 0.32 & -0.84 & 0.32 & 0.30 \\ 0.20 & 0.44 & 0.20 & 0.84 \end{pmatrix} \begin{pmatrix} y_1(n-1) \\ y_2(n-1) \\ y_1(n) \\ y_2(n) \end{pmatrix}$
  $+ \begin{pmatrix} -0.93 & -0.32 \\ -1.25 & -0.20 \end{pmatrix} \begin{pmatrix} u_1(n-1) \\ u_1(n) \end{pmatrix}$
- $\rho^1_{c_3}(n) = (-0.34 \ 0.85 \ -0.17 \ 0.34) \begin{pmatrix} y_1(n-1) \\ y_2(n-1) \\ y_1(n) \\ y_2(n) \end{pmatrix}$
  $+ (0.00 \ 0.17) \begin{pmatrix} u_1(n-1) \\ u_1(n) \end{pmatrix}$
- $\rho^1_{c_4}(n) = (-0.34 \ 0.85 \ -0.17 \ 0.34) \begin{pmatrix} y_1(n-1) \\ y_2(n-1) \\ y_1(n) \\ y_2(n) \end{pmatrix}$
  $+ (0.00 \ 0.17) \begin{pmatrix} u_1(n-1) \\ u_1(n) \end{pmatrix}$

Figures 2 and 3 show the real-time evolution during 10 seconds of system residuals when the multimode system is under different modes indicated at the bottom of the figures. Residuals are computed according to the sampling period $T_s = 0.01s$, by the residual bench that takes as input observable variables: the input U and the output Y.
We can verify that residuals of mode q_i are null when the system mode is q_i, $\forall i \in [|1..4|]$. We notice that residuals of mode q_3 and q_4: $\rho_3 = [r_5]$ and $\rho_4 = [r_6]$, are null in mode q_3 as well as in mode q_4. Hence, a diagnosability problem can appear and will be discussed in subsection 3.6.

3.3 Residual Filter

Mode switches are characterized by a spurious jump of the residual values (c.f. figures 2 and 3) that is due to the fact that the temporal window –over which observations are recorded to evaluate the residuals– overlaps over two modes. Since residuals have been designed for every mode separately, this may cause false alarm problems. This is solved by implementing a residual filter that takes as input residual values computed at every time step, and generating as output clean boolean indicators that reflect the consistency between model and observed behavior. Filtered residuals are denoted by the same symbols r_i

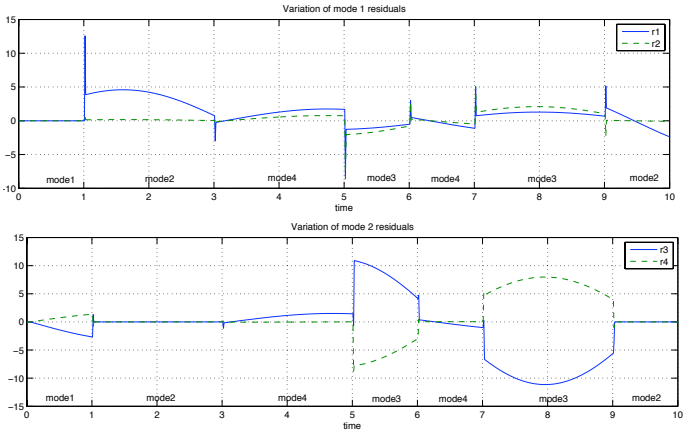

Fig. 2. Residuals of Mode 1 and 2: $\rho_{c_1} = [r_1, r_2]^T$ and $\rho_{c_2} = [r_3, r_4]^T$

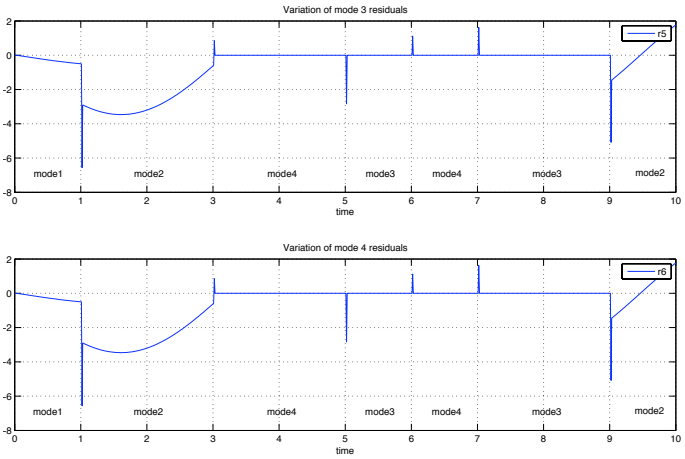

Fig. 3. Residuals of Mode 3 and 4: $\rho_{c_3} = [r_5]$ and $\rho_{c_4} = [r_6]$

as rough residuals.

The principle of the filtering is to hold-on to the current value as long as the residual is not computed to a different value during a number of steps specified by a prefixed temporal window T_{Filter}. The value of T_{Filter} determines the filter sensitivity with respect to the rough residual changes. It is set according to the physical properties of the dynamic system (time response, etc ...).

3.4 Mode Signature

The concept of fault signature used for continuous systems is now extended to multimode systems. Boolean residuals associated to every mode are computed by the residual filter and they are used to evaluate the observed signature at every time step. Observed signature is thereafter compared to pre-computed mode signatures. New concepts of mirror and reflexive signatures are defined and lead to the definition of mode signature as defined in Bayoudh et al. [2007].

Mirror and Reflexive signatures

Definition 1. Mirror Signature.
Given the vector $R_k = [r_{k1}, r_{k2}, ..., r_{kN_{ARR(q_k)}}]^T$ of system residuals in mode q_k, the q_k-mirror signature of mode

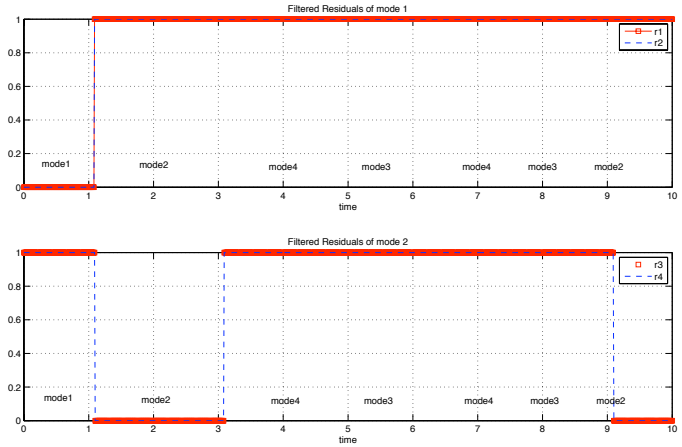

Fig. 4. Filtered residuals of Mode 1 and 2 (graphs for r_1 (r_3) and r_2 (r_4) are superposed)

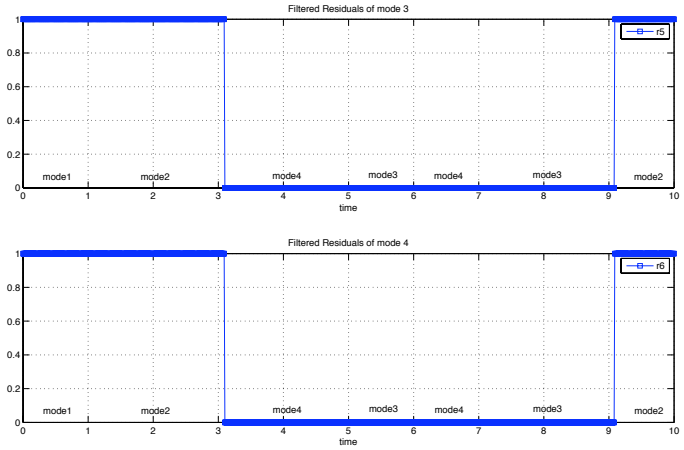

Fig. 5. Filtered residuals of Mode 3 and 4

q_j is given by vector $S_{j/k} = [s_{1_{j/k}}, ..., s_{N_{ARR(q_k)_{j/k}}}]^T = [R_k(\zeta^j_{OBS})]^T$.

The q_k-mirror signature of mode q_j is the vector of residuals of mode q_k computed with the observations ζ^j_{OBS} obtained when the system is in mode q_j. We say that it is the signature of mode q_j seen in mode q_k.

Definition 2. Reflexive Signature.
Given the vector $R_j = [r_{j1}, r_{j2}, ..., r_{jN_{ARR(q_j)}}]^T$ of system residuals in mode q_j, the reflexive signature of mode q_j is given by vector $S_{j/j} = [0, 0, ..., 0]^T = [R_j(\zeta^j_{OBS})]^T$.

The reflexive signature of mode q_j is the vector of residuals of mode q_j computed with the observations ζ^j_{OBS} obtained when the system mode is q_j. Note that the reflexive signature of a mode is actually its own mirror signature.

Mode signatures

Definition 3. Mode Signature.
The mode signature of a mode q_j is the vector obtained by the concatenation of all its mirror signatures, $Sig(q_j) = [S^T_{j/1}, S^T_{j/2}, ..., S^T_{j/j}, ..., S^T_{j/m}]^T$.

3.5 Illustrative example – Mode signatures

The pre-computed mode signatures of the example of figure 1 are given in table 1. The observed mode signature

$Sig(q_1) = \begin{pmatrix} S_{1/1} \\ S_{1/2} \\ S_{1/3} \\ S_{1/4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ - \\ 1 \\ 1 \\ - \\ 1 \\ - \\ 1 \end{pmatrix}$	$Sig(q_2) = \begin{pmatrix} S_{2/1} \\ S_{2/2} \\ S_{2/3} \\ S_{2/4} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ - \\ 0 \\ 0 \\ - \\ 1 \\ - \\ 1 \end{pmatrix}$
$Sig(q_3) = \begin{pmatrix} S_{3/1} \\ S_{3/2} \\ S_{3/3} \\ S_{3/4} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ - \\ 1 \\ 1 \\ - \\ 0 \\ - \\ 0 \end{pmatrix}$	$Sig(q_4) = \begin{pmatrix} S_{4/1} \\ S_{4/2} \\ S_{4/3} \\ S_{4/4} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ - \\ 1 \\ 1 \\ - \\ 0 \\ - \\ 0 \end{pmatrix}$

Table 1. Mode Signatures of the underlying continuous system Ξ

$[r_1, r_2, r_3, r_4, r_5, r_6]^T$ is evaluated on-line from observable variables (U, Y) and the corresponding boolean residuals when the system goes along a sequence of modes are given in figures 4 and 5.

3.6 CS Diagnosability Discussion

We notice that mode 3 and 4 have the same mode signature, therefore the underlying continuous system is not diagnosable w.r.t the diagnosability definition of Bayoudh et al. [2007] defined as follows:

Definition 4. Diagnosability.
Two modes q_i and q_j ($i \neq j$) are diagnosable iff $Sig(q_i) \neq Sig(q_j)$. The underlying continuous system Ξ is diagnosable iff all pairs of modes q_i and q_j, $i \neq j$, are diagnosable.

The hybrid system can be diagnosable although the underlying continuous system is not, as shown in Bayoudh et al. [2007]. Hybrid diagnosis must call upon both continuous and discrete knowledge. In our approach, we associate discrete events to signature switches, in order to combine continuous and discrete informations in the same framework. Then, DES diagnosis techniques can be performed.

4. DES DIAGNOSIS FRAMEWORK

4.1 Diagnoser Approach

The discrete part of the system is modeled by the finite state machine $M = (Q, \Sigma, T, q_0)$ (see section 2.1). We consider $\Sigma_F \subseteq \Sigma_{uo}$ as the set of fault events to be diagnosed. We assume that the underlying discrete event system, M, has no unobservable cycles (i.e cycles containing unobservable events only).
The set of fault events Σ_F is partitioned into disjoint sets corresponding to different fault types F_i, $\Sigma_F = \Sigma_{F_1} \cup \Sigma_{F_2} \cup ... \cup \Sigma_{F_n}$ and $\Sigma_{F_i} \cap \Sigma_{F_j} = \emptyset$, $for\, i \neq j$.
The aim of the diagnosis is to make inferences about past occurrences of fault types on the basis of the observed events. In order to solve this problem the system model is directly converted into a diagnoser.

The diagnoser $Diag(M) = (Q_{Diag}, \Sigma_{Diag}, T_{Diag}, q_{0\,Diag})$ is a deterministic finite state machine built from the system model $M = (Q, \Sigma, T, q_0)$, where:

- $\Sigma_{Diag} = \Sigma_o$ is the set of observable events of the system.
- Q_{Diag} is the set of states of the diagnoser: $Q_{Diag} \subseteq 2^{Q \times 2^{\Sigma_F}}$ i.e. $Q_{Diag} \subseteq \mathcal{P}(Q \times \mathcal{P}(\Sigma_F))$, where $\mathcal{P}(E)$ denotes the power set of E. The states of the diagnoser provide the set of diagnosis candidates as a set of couples whose first element refers to the state of the original system and the second is a label providing the set of faults on the path leading to this state.
- T_{Diag} is the diagnoser transition function built by a recursive process that consists in computing all the reachable states from the diagnoser initial state and by propagating the diagnosis information. For more details see Sampath et al. [1995].
- $q_{0Diag} = \{(q_0, \{\emptyset\})\} \in Q_{Diag}$, is the initial state of the diagnoser.

Definition 5. Uncertain Diagnoser State.
Given a diagnoser state $q_{Diag} \in Q_{Diag}$, this state is F_i-uncertain iff F_i does not belong to all the labels of the state whereas F_i belongs to at least one label of the state.

4.2 DES Diagnosability Discussion

We recall the definition of DES diagnosability.

Definition 6. DES Diagnosability.
A fault F is diagnosable iff its occurrence is always followed by a finite observable sequence of events that allows one to diagnose F with certainty (as defined in Pencolé [2004]). The system is said to be diagnosable iff all the anticipated faults are diagnosable.

Formally, let $s_F t$ be a sequence of events (or trajectory) such that s_F ends with the occurrence of F, and t is a continuation of s_F. F is diagnosable iff:
$\forall$ trajectory $s_F t$, $\exists$ an integer n: length(t) $\geq n \Rightarrow$ ($\forall$ trajectory s such that $P_{\Sigma_o}(s)$=$P_{\Sigma_o}(s_F t)$, F occurs in s), where P_{Σ_o} is the projection operator on the set of observable events.

The criterion to check DES diagnosability using the diagnoser, defined in Sampath et al. [1995], is the following:

Theorem 1. The system M is not diagnosable iff the associated diagnoser $Diag(M)$ contains an uncertain cycle, i.e. a cycle in which there is at least one F_i-uncertain state for some F_i and whose states also define a cycle in the original system M.

In the DES community the diagnoser is used to analyze the diagnosability property of the system or as a discrete observer. The definitions and theorem defined above for a fault event from $\Sigma_F \subset \Sigma_{uo}$ generalize to any unobservable event from Σ_{uo}. In our approach, the diagnoser is extended to hybrid systems and used for on-line supervision [2].

4.3 Illustrative example – DES Diagnosability

The diagnoser of the underlying discrete event system of figure 2.1 is given in figure 6. The underlying discrete system is non diagnosable because of the presence of one uncertain cycle $(o1, o2)$ and whose states also define a cycle in the original automaton (theorem 1).

[2] This paper deals with the hybrid diagnosis approach. Diagnosability analysis of hybrid systems has been studied in Bayoudh et al. [2007].

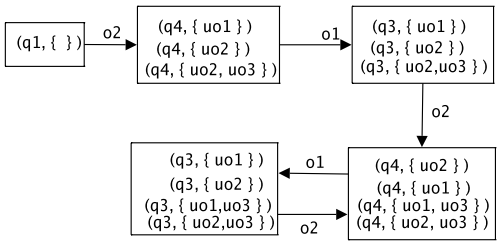

Fig. 6. The diagnoser of the underlying discrete system M

5. HYBRID DIAGNOSIS BY COMBINING CS AND DES APPROACHES

In our approach, a fault event is necessarily followed by the system being in the corresponding fault mode. Hence, in this framework, state tracking, i.e. estimating the sequence of modes of the system, is equivalent to fault event detection . Our method includes the following steps: the parity space approach is applied to diagnose the underlying continuous system. Then, mode signatures are generated after residual filtering. The mode signature switches, i.e. change of value of a subset of boolean residuals, emerge on the discrete domain in the form of discrete events as explained in 5.1. Finally, the DES approach diagnoser technique can be applied to perform hybrid diagnosis.

5.1 The abstraction of CS dynamics in terms of discrete events

The mode signature switches are abstracted in terms of discrete events. By adding these events to the underling DES, we obtain the behavior automaton, that couples both discrete and continuous knowledge (like in Lunze [2000] and Bayoudh et al. [2006]).

Assumption 1. We assume that the dynamics of the discrete command events are slower than the dynamics of residual generators (mode signatures have time to establish between two consecutive discrete events).

We define a function f_{CS_DES}, which associates an event issued from the continuous domain (representing a change of the mode signature) for each discrete transition of the underlying DES.
This function aims at defining Σ^{Sig} as the set of discrete events issued from the abstraction of continuous dynamics of the multimode system.

$$f_{CS\_DES} : Q \times T(Q) \longrightarrow \Sigma^{Sig}$$
$$(q_i, q_j) \longmapsto \begin{cases} Ro_{ij} \in \Sigma_o^{Sig} \text{ if } Sig(q_i) \neq Sig(q_j) \\ Ruo_{ij} \in \Sigma_{uo}^{Sig} \text{ if } Sig(q_i) = Sig(q_j) \end{cases}$$

- Σ_o^{Sig} is a set of observable events, generated when the mode signature of the source mode is different from the mode signature of the destination mode.
- Σ_{uo}^{Sig} is a set of unobservable events generated when the mode signature of the source mode is equal to the mode signature of the destination mode.
- We define $\Sigma^{Sig} = \Sigma_o^{Sig} \cup \Sigma_{uo}^{Sig}$.

Hence, the behavior of the hybrid system can be described by a word of the hybrid language $L_{hyb} \subseteq (\Sigma^{Sig} \cup \Sigma)^*$ according to the language theory as in Ramadge and Wonham [1989]. The hybrid language can be generated by the behavior automaton as in Bayoudh et al. [2006].

5.2 Illustrative example – behavior automaton

The behavior automaton of the hybrid system of figure 2.1 is given in figure 7.

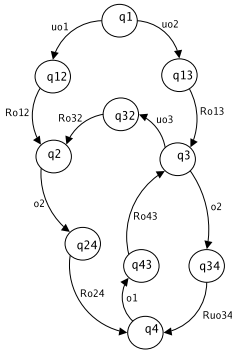

Fig. 7. The behavior automaton of the hybrid system S

5.3 The Diagnoser Approach applied to Hybrid Systems

The diagnoser approach is applied to hybrid systems by computing a hybrid diagnoser from the behavior automaton. We use the DIADES software from Pencolé [2006] to compute offline the finite state machine modeling the diagnoser. Then, this diagnoser is implemented in a MATLAB/SIMULINK block and used to track the system mode on-line. It takes as input observable (pure) discrete events, and observable events issued from the abstraction of continuous dynamics.

5.4 Illustrative example – Results

Let us take the hybrid system of figure 1. We have showed in subsections 3.6 and 4.3 that the underlying discrete event and continuous systems are non diagnosable.
The system model is implemented in a MATALAB/SIMULINK block, and diagnosed on-line. The sampling period is $0.01s$. The system state is tracked at every time step during the simulation time ($10s$).
We command the system from the initial state q_1 to follow the discrete trajectory: $[(uo_1, t = 1s), (o_2, t = 3s), (o_1, t = 5s), (o_2, t = 6s), (o1, t = 7s), (uo3, t = 9s)]$. We display both real and estimated system modes. We notice that the hybrid diagnoser is able to follow the system mode, even after a non observable event $uo1$. The hybrid diagnoser can also diagnose modes q_3 and q_4 that have the same mode signature, by coupling continuous and discrete informations. We can notice that the diagnoser tracks the system mode with a little delay, due to the sensitivity of the residual filter T_{Filter}, and the computation time.
The results are quite good and open numerous perspectives.

6. CONCLUSION

This paper deals with the problem of diagnosing systems that exhibit both continuous and discrete event dynamics. The method that has been proposed combines techniques from both continuous and discrete event diagnosis fields. The resulting on-line hybrid model based state tracking algorithm achieves improved diagnosability with respect to using separately continuous dynamic diagnosis and discrete dynamic diagnosis. This is illustrated by an example. It is interesting to notice that, although techniques from both continuous and discrete event diagnosis fields are combined, the method remains quite

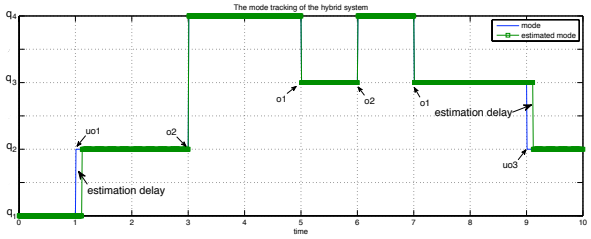

Fig. 8. Mode tracking of the hybrid system

modular. In particular, we have used the parity space approach to derive the signatures associated to each operational mode but another FDI approach could be used. The key issue is actually the abstraction of signature changes into a relevant set of events. Numerous perspectives arise from this work. In particular we are interested in extending the approach in a distributed framework. This would allow us to tackle the combinatorial problem of the diagnoser approach. On the other hand, we aim at closing the loop, i.e. at using the results of our diagnoser to perform reconfiguration actions guided by diagnosability properties of the system.

REFERENCES

M. Bayoudh, L. Travé-Massuyès, and Xavier Olive. Hybrid systems diagnosability by abstracting faulty continuous dynamics. In *Proceedings of the 17th International Workshop on Principles of Diagnosis DX'06*, pages 9–15, Burgos, Spain, 2006.

M. Bayoudh, L. Travé-Massuyès, and Xavier Olive. State tracking in the hybrid space. In *Proceedings of the 18th International Workshop on Principles of Diagnosis DX'07*, pages 221–228, Nashville, TN, USA, 2007.

V. Cocquempot, T. El Mezyani, and M. Staroswiecki. Fault detection and isolation for hybrid systems using structured parity residuals. *IEEE/IFAC-ASCC: Asian Control Conference*, 2004.

M.O. Cordier, P. Dague, F. Lévy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès. Conflicts versus analytical redundancy relations : A comparative analysis of the model-based diagnostic approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man and Cybernetics, Part B.*, 34(52163-2177), 2004.

T. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 278–292, New Brunswick, New Jersey, 1996.

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

Jan Lunze. Diagnosis of quantized systems by means of timed discrete-event representations. In *HSCC*, pages 258–271, 2000.

Sheila A. McIlraith, Gautam Biswas, Dan Clancy, and Vineet Gupta. Hybrid systems diagnosis. In *HSCC*, pages 282–295, 2000.

Y. Pencolé. Diagnosability analysis of distributed discrete event systems. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004*, pages 43–47, 2004.

Y. Pencolé. Diades, diagnosis of discrete event systems, 2006. URL `http://www.laas.fr/~ypencole/DiaDes`.

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

M. Sampath, R. Sengputa, S. Lafortune, K. Sinnamohideen, and D. Teneketsis. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40:1555–1575, 1995.